



**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

Кафедра

«Обчислювальна математика і програмування»

**КОНСПЕКТ ЛЕКЦІЙ
з дисципліни «Інформатика» для студентів заочної форми
навчання напрямку «Гірництво»**

Донецьк - 2011



**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**

**КОНСПЕКТ ЛЕКЦІЙ
з дисципліни «Інформатика» для студентів заочної форми
навчання напрямку «Гірництво»**

Розглянуто на засіданні
кафедри ОМіП
протокол № 4 від 23.11.2011

Затверджено на засіданні
навчально- видавничої Ради ДонНТУ
протокол № ____ від _____.

Донецьк-2011



УДК

Конспект лекцій з дисципліни «Інформатика» для студентів заочної форми навчання напрямку «Гірництво» /Укл.: О.В.Прокопенко, С.В.Масло - Донецьк: ДонНТУ, 2011. - 225с.

Приведено тексти лекцій, у яких викладені основні принципи структури сучасних комп'ютерів, принципи апаратної реалізації й основні представлення про програмне забезпечення, приведені основи з алгоритмізації, алгоритми основних обчислювальних процесів, викладені основи алгоритмічної мови Visual Basic For Application (VBA), основи роботи з електронними таблицями, системою керування базами даних, основи в середовищі MATHCAD принципи комп'ютерної презентації, структура і принципи функціонування комп'ютерних мереж, основи функціонування і роботи в мережі Інтернет.

Укладачі: О.В. Прокопенко, ст.викладач

С.В. Масло, асистент

Рецензент: І.В.Філатова, доцент, к.т.н.

Відповідальний

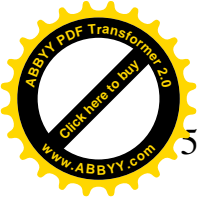
за випуск

В.М. Павлиш, проф., д.т.н.

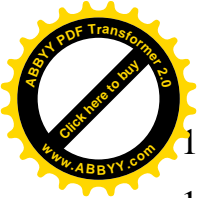


СОДЕРЖАНИЕ

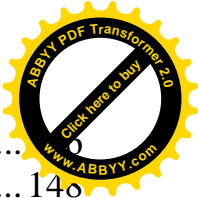
| | |
|---|----|
| ВВЕДЕНИЕ | 12 |
| 1. НАУЧНЫЕ ОСНОВЫ ИНФОРМАТИКИ | 14 |
| 1.1. Цели и задачи дисциплины..... | 14 |
| 1.2. Теоретические основы информатики..... | 14 |
| 1.3. Практические основы информатики | 15 |
| 1.3.1. Структура компьютера и принцип программного управления .. | 15 |
| 2. СИСТЕМЫ СЧИСЛЕНИЯ И КОДИРОВАНИЯ ИНФОРМАЦИИ | 17 |
| 2.1. Виды систем счисления | 17 |
| 2.2. Перевод чисел из восьмеричной и шестнадцатеричной системы счисления в десятичную | 19 |
| 2.3. Перевод чисел из восьмеричной и шестнадцатеричной системы счисления в двоичную | 19 |
| 2.4. Перевод чисел из двоичной системы счисления в восьмеричную и шестнадцатеричную..... | 20 |
| 2.5. Перевод чисел из десятичной системы счисления в другие..... | 20 |
| 2.5.1. Перевод целых десятичных чисел в другие системы счисления | 21 |
| 2.5.2. Перевод дробных чисел..... | 22 |
| Индивидуальные задания по теме 1..... | 22 |
| 3. АЛГОРИТМ И ПРОГРАММА. ВИДЫ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ | 25 |
| 3.1. Алгоритм | 25 |
| 3.1.1. Требования к алгоритму..... | 27 |
| 3.2. Программа | 28 |
| 3.3. Виды вычислительных процессов..... | 28 |
| 4. МАССИВЫ. ТИПОВЫЕ АЛГОРИТМЫ | 38 |
| 4.1. Определение суммы чисел произвольного ряда..... | 38 |
| 4.2. Определение произведения чисел произвольного ряда..... | 40 |
| 4.3. Алгоритм определения наибольшего (наименьшего) числа из заданного ряда чисел | 40 |
| 4.4. Алгоритм определения наибольшего (наименьшего) значения вычисляемой функции | 42 |
| 4.5. Вычисление функции с одновременно изменяющимися несколькими аргументами | 43 |
| 4.6. Итерационные циклы | 44 |
| 5. ТИПОВЫЕ АЛГОРИТМЫ | 47 |



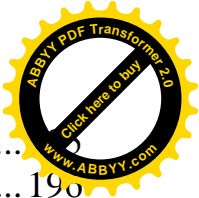
| | |
|--|-----------|
| 5.1. Сортировка ряда чисел..... | 47 |
| 5.1.1. Метод смежных пар..... | 47 |
| 5.1.2. Метод поиска наименьшего (наибольшего)..... | 50 |
| 5.2. Типовые алгоритмы решения задач с использованием матриц..... | 51 |
| 6. ОСНОВЫ ПРОГРАММИРОВАНИЯ В VBA..... | 55 |
| 6.1. Элементы языка VBA | 57 |
| 6.2. Типы данных языка VBA..... | 58 |
| 6.3. Переменные | 59 |
| 6.4. Константы..... | 60 |
| 6.5. Операции | 60 |
| 6.6. Математические функции VBA | 62 |
| 6.7. Строковые функции | 62 |
| 6.8. Функции преобразования типов Str() и Val() | 63 |
| 6.9. Организация ввода/вывода информации | 63 |
| 7. ОПЕРАТОРЫ ЯЗЫКА VBA И РЕАЛИЗАЦИЯ ОСНОВНЫХ АЛГОРИТМИЧЕСКИХ КОНСТРУКЦИЙ..... | 66 |
| 7.1. Программирование алгоритмов линейной структуры | 67 |
| 7.2. Операторы управления порядком выполнения команд | 68 |
| 7.3. Программирование алгоритмов разветвляющейся структуры | 70 |
| 7.4. Программирование алгоритмов циклической структуры..... | 72 |
| 8. МАССИВЫ | 76 |
| 8.1. Одномерные массивы | 77 |
| 8.2. Некоторые алгоритмы обработки матриц..... | 79 |
| 9. ПОДПРОГРАММЫ И ФУНКЦИИ | 80 |
| 10. ОТЛАДКА ПРОГРАММ..... | 84 |
| 10.1. Ошибки при создании программы | 84 |
| 10.2. Ошибки при выполнении программы..... | 85 |
| 10.3. Тестирование программ и поиск ошибок | 86 |
| 10.4. Отладка программ в редакторе VBA..... | 87 |
| 11. ПРИКЛАДНЫЕ ПРОГРАММЫ..... | 91 |
| 11.1. Текстовый редактор Microsoft Word | 91 |
| Основное меню | 93 |
| 11.2. Электронные таблицы. MS Excel. Основные представления | 94 |
| 12. ЭЛЕКТРОННАЯ ТАБЛИЦА MS EXCEL. ДАННЫЕ..... | 97 |
| 12.1. Запуск и завершение программы MS Excel | 97 |



| | |
|--|------------|
| 12.2. Данные. Ввод данных | 27 |
| 13.1. Редактирование данных | 101 |
| 13.2.Операции с данными..... | 102 |
| 13.3. Работа с формулами | 104 |
| 13.4. Использование в формулах адресов ячеек..... | 104 |
| 14. ЭЛЕКТРОННАЯ ТАБЛИЦА MS EXCEL. ФУНКЦИИ | 105 |
| 14.1. Операции и функции..... | 105 |
| 14.2. Относительная и абсолютная адресация ячеек..... | 108 |
| 15. ПОСТРОЕНИЕ ДИАГРАММ В СРЕДЕ EXCEL | 110 |
| 16. РЕШЕНИЕ МАТЕМАТИЧЕСКИХ ЗАДАЧ СРЕДСТВАМИ MS EXCEL | 113 |
| 16.1. Операции с матрицами..... | 113 |
| 16.1.1. Транспонирование матриц | 113 |
| 16.1.2. Вычисление определителя матрицы..... | 114 |
| 16.1.3. Нахождение обратной матрицы | 115 |
| 16.1.4. Сложение и вычитание матриц..... | 115 |
| 16.1.5. Умножение матрицы на число | 116 |
| 16.1.6.Умножение матриц | 116 |
| 17. РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ..... | 117 |
| 17.1. Метод обратной матрицы | 118 |
| 17.2 Решения системы линейных уравнений методом Гаусса | 124 |
| 17.3 Решение системы линейных уравнений с помощью надстройки "Поиск решения" | 125 |
| 18. НАХОЖДЕНИЯ КОРНЕЙ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ..... | 126 |
| 18.1. Решение уравнения с одним неизвестным..... | 129 |
| 19. ЭЛЕКТРОННЫЕ ТАБЛИЦЫ MS EXCEL. ФОРМАТИРОВАНИЕ ДАННЫХ. ОПЕРАЦИИ С БОЛЬШИМИ ТАБЛИЦАМИ И ЛИСТАМИ..... | 131 |
| 19.1. Форматирование данных | 131 |
| 19.1.1.Форматирование текста | 131 |
| 19.1.2.Форматирование чисел..... | 132 |
| 19.1.3.Форматирование символов..... | 133 |
| 19.1.4. Автоформатирование..... | 133 |
| 19.1.5.Копирование форматирования..... | 133 |
| 19.2.Операции с большими таблицами | 134 |
| 19.3.Операции с рабочими листами..... | 136 |
| 19.4.Печать..... | 138 |
| 20. ПОСТРОЕНИЕ ГРАФИКОВ В MICROSOFT EXCEL..... | 140 |
| 21. БАЗЫ ДАННЫХ. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ | 144 |
| 21.1. Форма хранения информации..... | 146 |
| 12.2. Реляционная модель базы данных..... | 146 |



| | | |
|------------------|--|------------|
| 21.2.1. | Структура таблиц базы данных | 148 |
| 21.2.2. | Индексы базы данных | 148 |
| 21.2.3. | Связи базы данных | 149 |
| 21.3. | Основные виды работы с базами данных | 152 |
| 22. | СИСТЕМА УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ MS ACCESS | 153 |
| 22.1. | Назначение и возможности СУБД MS Access | 153 |
| 22.2. | Запуск и завершение работы программы MS Access | 154 |
| 23.3. | Типы данных | 154 |
| 22.4. | Разработка и создание базы данных..... | 156 |
| 22.4.1. | Разработка структуры базы данных | 156 |
| 22.4.2. | Имена полей и объектов..... | 157 |
| 22.5. | Создание новой базы данных средствами СУБД MS Access..... | 157 |
| 22.5.1. | Создание таблиц базы данных с помощью мастера таблиц | 158 |
| 22.5.2. | Создание таблиц в режиме конструктора..... | 160 |
| 22.5.3. | Создание связей | 162 |
| 22.5.4. | Редактирование, удаление связей | 163 |
| 23. | ПРОГРАММА MS ACCESS. РАБОТА С ОБЪЕКТАМИ..... | 164 |
| 23.1. | Формы. Работа с формами..... | 164 |
| 23.1.1. | Создание формы в режиме автоформ..... | 164 |
| 23.1.2. | Создание формы в режиме мастера форм | 165 |
| 23.1.3. | Ввод и редактирование записей с помощью формы | 166 |
| 23.1.4. | Простейшие модификации формы с помощью конструктора | 167 |
| 23.2. | Поиск и запросы | 167 |
| 23.2.1. | Поиск и фильтрация данных..... | 167 |
| 23.2.2. | Применение фильтра | 169 |
| 23.3. | Типы запросов | 170 |
| 23.3.1. | Создание запроса на выборку | 172 |
| 23.3.2. | Выполнение запроса..... | 174 |
| 23.3.3. | Модификация запроса | 174 |
| 24. | СОЗДАНИЕ ОТЧЕТОВ В MS ACCESS. ПРЕЗЕНТАЦИЯ..... | 175 |
| 24.1. | Создание отчетов..... | 175 |
| 24.1.1. | Режим автоотчета..... | 176 |
| 24.1.2. | Создание отчета с помощью мастера | 176 |
| 24.1.3. | Модификация отчета в режиме конструктора | 179 |
| 24.2. | Печать отчета..... | 180 |
| 25. | РАБОТА В СРЕДЕ MATHCAD..... | 181 |
| 26. | ИНТЕРНЕТ, ОБЩИЕ ПРЕДСТАВЛЕНИЯ. ОСНОВЫ РАБОТЫ В | |
| ИНТЕРНЕТЕ | | 194 |
| 26.1. | Основные понятия и определения..... | 194 |
| 26.2. | Протоколы Интернет..... | 194 |



| | |
|--|------------|
| 26.2.1. IP – адреса и классы сетей..... | 198 |
| 26.2.2. Доменные адреса | 198 |
| 26.3. Узел Интернета..... | 197 |
| 26.3.1. Получение доступа к Интернету..... | 197 |
| 26.3.2. Вход в Интернет | 199 |
| 26.4. Информационные ресурсы Интернет..... | 200 |
| 26.5. Web – страница..... | 201 |
| 26.5.1. Адрес ресурса | 201 |
| 26.6. Электронная почта | 202 |
| 26.7. Сообщения..... | 203 |
| 27. ЭЛЕКТРОННАЯ ПОЧТА. ОБОЗРЕВАТЕЛИ СЕТИ ИНТЕРНЕТ | 205 |
| 27.1. Почтовый сервер | 205 |
| 27.2. Операции с исходящими сообщениями..... | 206 |
| 27.3. Операции с входящими сообщениями..... | 207 |
| 27.4. Электронные доски объявлений..... | 208 |
| 27.5. Телеконференции | 208 |
| 27.6. Пересылка файлов..... | 209 |
| 27.7. Интернет – пейджеры..... | 209 |
| 27.8. Базы данных в Интернет | 210 |
| 27.9. Обзорщики сети Интернет | 210 |
| 27.9.1. Обзорщик Internet Explorer | 211 |
| 28. СОЗДАНИЕ WEB – СТРАНИЦ, ПОИСКОВЫЕ СИСТЕМЫ | 215 |
| 28.1. создание Web – страниц..... | 215 |
| 28.2. Поиск Web – страниц в Паутине | 218 |
| 28.3. Загрузка файлов из сети Интернет | 220 |
| 28.4. Работа с почтовым сайтом | 221 |
| 28.5. Программа Outlook Express | 221 |
| 28.5.1. Запуск и интерфейс программы Outlook Express..... | 221 |
| 28.5.2. Создание учетной записи | 222 |
| 28.5.3. Чтение сообщений | 223 |
| 28.5.4. Создание и отправка сообщения..... | 223 |
| 28.5.5. Разметка и форматирование сообщений | 224 |
| 28.6. Работа с FTP – серверами с помощью встроенного FTP – клиента Total Commander..... | 224 |
| ЛИТЕРАТУРА..... | 225 |

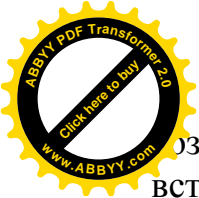


ВВЕДЕНИЕ

Современный образованный человек и тем более специалист высшей квалификации, независимо от профиля его специализации, должен иметь представление об устройстве и принципе действия компьютера, аппаратном и программном его обеспечении, владеть основами его эксплуатации, то есть иметь элементарные навыки работы с компьютером. К ним относятся: умение включить компьютер, запустить программу, скопировать документ, отформатировать гибкий диск и др. Любой специалист высшей квалификации должен уметь пользоваться популярным в настоящее время пакетом прикладных программ MS Office. Весьма полезно уметь пользоваться текстовым редактором MS Word, позволяющим создавать и редактировать тексты любой сложности, таблицы любой формы, математические формулы, графики. В практической деятельности просто необходимо уметь работать с электронной таблицей MS Excel, системой управления базами данных MS Access, использовать возможности программы Power Point для создания компьютерной презентации различных объектов (изделий, научных докладов, проектов и др.).

При возникновении какой – либо нестандартной задачи специалист должен суметь правильно ее сформулировать, определить исходные данные, разработать алгоритм ее решения. Последнее требует хорошего логического мышления. В предлагаемом курсе лекций заметное внимание уделяется вопросу алгоритмизации различных задач. В настоящее время разработаны пакеты прикладных программ, такие как Matcad, Matlab и др., которые позволяют решать широкий круг инженерных вопросов, не прибегая к программированию в классическом его понимании. Это обстоятельство породило в кругу специалистов дискуссию о целесообразности изучения алгоритмических языков студентами, профиль специализации которых не связан непосредственно с вопросами программирования. Тем не менее в лекционном курсе уделяется внимание изучению основ современного алгоритмического языка высокого уровня Visual Basic с целью получения элементарных навыков программирования. Это основано на убеждении, что специалист должен иметь представление о сути программирования, так как это позволяет более глубоко понимать особенность вычислений, выполняемых с помощью компьютера. Более того, умение программировать может быть полезным при решении нестандартных задач, реализация которых с помощью специальных пакетов прикладных программ представляется сложной процедурой или обеспечивается не в полной мере.

В предлагаемом курсе лекций не столь глубоко излагаются особенности работы с текстовым редактором MS Word. Это объясняется тем, что большинство студентов уже получили опыт работы с ним еще в период обучения в школе и часто пользуются им в повседневной жизни при подготовке отчетов, рефератов и др. документации. Основной акцент по изучению этого редактора переносится на практические занятия, целью которых является приобретение навыков создания и форматирования текстов, создание и редактирование таблиц любой конфигурации,



создание математических формул различной сложности с использованием встроенного редактора формул, получение навыков работы с векторной графикой при создании блок-схем, электрических схем и других графических объектов.

В предлагаемой версии дисциплины «Введение в информатику» рассматриваются вопросы изучения электронной таблицы MS Excel, ее интерфейс, возможности и особенности ее применения для решения инженерных задач.

Современный мир – это мир информации и информационных технологий. Поэтому весьма важным для современного специалиста является умение пользоваться большими объемами информации, организовывать ее хранение, обновлять и рационально ее использовать. В настоящее время разработаны различные пакеты программ по созданию и управлению базами данных. Они позволяют систематизировать и хранить большие объемы информации, создавать средства для рационального ее использования. В предлагаемом лекционном курсе уделяется внимание распространенной программе MS Access, представляющей собой систему управления базами данных. Излагаются основные принципы реляционного метода организации и хранения информации, используемые этой системой, рассматриваются основные объекты, позволяющие рационально использовать большие объемы информации.

Во многих случаях весьма важно правильно и привлекательно представить новые разработки (технические решения, результаты научных исследований, изделия, технические объекты, проекты и т.п.). От этого во многом зависит успех дальнейших действий. Современный специалист, работая в конкурентной среде, при стремлении достичь желаемого результата своего труда, должен владеть такими навыками. В настоящее время имеются программные продукты, позволяющие создавать компьютерные презентации (представления) различных объектов. Одной из таких является программа Power Point. Особенности работы с ней, способам создания презентаций, уделяется внимание в излагаемом курсе дисциплины.

Все те вопросы, которые рассматриваются в предлагаемом курсе лекций, позволяют получить основные навыки и начальные знания в области информатики и послужить основой для их совершенствования и углубления при дальнейшем ее изучении.



ЛЕКЦИЯ №1

1. НАУЧНЫЕ ОСНОВЫ ИНФОРМАТИКИ

1.1. Цели и задачи дисциплины

Основной задачей учебного курса «ВВЕДЕНИЕ В ИНФОРМАТИКУ» является изучение основ информатики, которые позволили бы студенту:

- *получить* представление об основных понятиях информатики,
- *уяснить* принципы аппаратной реализации вычислений на компьютере,
- *узнать* о методах сбора, хранения и обработки информации в компьютере и компьютерных сетях,
- *ознакомиться* с тенденциями развития информационных технологий;
- *знать* архитектуру, структуру и принцип действия компьютера, логические основы образования цифровых схем и основные возможности компьютера, его информационное окружение, техническую базу и программные способы реализации информационных процессов, методы алгоритмизации и программирования вычислений;
- *уметь* пользоваться компьютерами и компьютерными сетями, использовать операционные и файловые системы, пользоваться системным программным обеспечением, алгоритмическими языками высокого уровня в интегрированной среде, текстовыми и графическими редакторами, базами данных и электронными таблицами при решении практических задач.

1.2. Теоретические основы информатики

Информатика - это фундаментальная естественная наука о средствах, методах, способах получения, хранения, переработки и передачи информации.

Информация – это какие-либо сведения технического, экономического, социального или другого характера. В информатике эти сведения принято называть **данными**.

Составными частями информатики являются:

- теория информации и кодирования,
- теория алгоритмов и формальных языков.

Базовыми понятиями науки информатики являются: «информация», «сообщение», «информационная модель», «кодирование», «алгоритм», «компьютер». Сведения различного характера представляются с помощью *сообщений*. Например: «Занятия по информатике переносятся...». Сообщения могут передаваться различными способами (письмо, звук, электрический сигнал и др.). Соответствие между информацией и сообщением, с помощью которого они передаются, не всегда является однозначной. Одна и та же информация может передаваться различными сообщениями и наоборот одно и то же сообщение можно воспринимать по-разному, то есть извлекаться различная информация. В том



лучае, когда сообщение передается на неизвестном слушателю языке зашифрованным непонятным способом, его получатель вообще не сможет извлечь какую-либо информацию. Таким образом, возникает вопрос о способе, качестве и полноте извлечения информации из сообщения. Обычно способ выявления смысла сообщения является результатом договоренности отправителя и получателя сообщения. Иначе говоря, задается правило интерпретации сообщения или же *кодирования*.

Кодирование сообщений производится различными способами. Если сообщение представляется в письменном виде, то оно предполагает изложение информации в виде совокупности букв, цифр и других специальных знаков. Если сообщение передается в виде звуковых сигналов, то оно кодируется сигналами различной продолжительности, тональности и др.

Кибернетика - наука об общих принципах управления в различных системах:

- Технические
- Биологических
- Социальных и т. д.

Информатика занимается изучением процессов преобразования и создания новой информации, практически не решая задачи управления объектами как кибернетика.

1.3. Практические основы информатики

Техническим инструментом реализации различных задач информатики является компьютерная техника. Основу ее составляет *компьютер*.

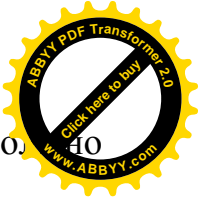
Компьютер – это сложное техническое устройство, предназначенное для автоматической обработки больших объемов информации.

1.3.1. Структура компьютера и принцип программного управления

Под *архитектурой* компьютера понимается совокупность сведений об основных устройствах компьютера и их назначении, о способах представления программ и данных в машине, об особенностях ее организации и функционировании.

Основные принципы устройства вычислительной машины были сформулированы в 1945 году американским математиком Джоном фон Нэйманом, которые сводились к тому, что каждая вычислительная машина должна состоять из:

- *арифметического и логического устройства* (АЛУ), которое должно выполнять арифметические и логические операции;
- *устройства управления* (УУ), которое должно читать команды (инструкции) программы и организовывать их выполнение;



– *запоминающего устройства (ЗУ)*, или *памяти*, которое до-
хранить данные и программу;

– *внешних устройств* для ввода, вывода и хранения информации.

Все современные компьютеры строятся на этих принципах.

Функционально – структурная организация компьютеров.

Составные части ПК.

- Системный блок
- Клавиатура
- Монитор
- Мышь

Могут быть подключены:

- Колонки
- Принтер
- Сканер

Процессор осуществляет вычисление и обеспечивает общее управление компьютером. В состав процессора входят: арифметико-логическое устройство (АЛУ), устройство управления.

Память компьютера – запоминающее устройство, напрямую связанное с процессором и предназначенное для хранения программ и данных в процессе работы компьютера.

Память состоит:

ОЗУ – оперативно – запоминающее устройство.

ПЗУ – постоянно запоминающее устройство.

В ОЗУ хранятся программы и данные в процессе работы компьютера.

В ПЗУ не может быть изменений. В нём хранятся основные функции ввода и вывода.

Внутри системного блока находится устройство, предназначенное для долговременного хранения информации – винчестер или жёсткий диск.

Внутри системного блока – видеокарта – устройство, соединяющее монитор с материнской платой. Объём видеокарты до 128 Мб.

Звуковая карта служит для преобразования цифрового сигнала в звуковой.

TV-тюнер – специальная плата, позволяющая просматривать телевизионные программы на мониторе.

Модем – устройство для подключения к телефонной сети с целью передачи/получения информации.

Сетевая карта – устройство, с помощью которого компьютеры объединяются в локальную сеть.

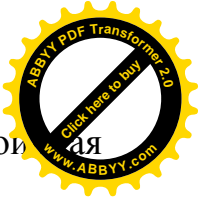
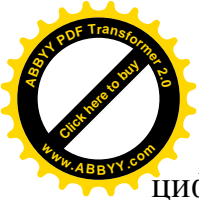
Клавиатура, мышь, принтер, сканер.

Матричный принтер – преимущество – дешевизна расходных материалов.

Струйный принтер – плюсы – высококачественная цветная печать. Минус – высокая стоимость расходных материалов.

Лазерный принтер - плюсы – высококачественная цветная печать. Минус – высокая стоимость расходных материалов.

Представление информации в ПК. Биты и байты. Системы счисления.



Бит – термин, являющийся сокращением от словосочетания «двоица цифра».

Байт – количество информации, необходимое для размещения одного символа. На одной странице текста – 4000 символов или байт.

Килобайт 1 Кб = 2^{10} байт = 1024 байт.

Мегабайт 1 Мб = 1024 Кб = 2^{20} байт = 1048576 байт.

Гигабайт 1 Гб = 1024 Мб = 2^{30} байт.

Терабайт 1 Тб = 1024 Гб = 2^{40} байт.

ЛЕКЦИЯ №2

2. СИСТЕМЫ СЧИСЛЕНИЯ И КОДИРОВАНИЯ ИНФОРМАЦИИ

2.1. Виды систем счисления

Под *системой счисления* понимают способ представления чисел. Существуют различные системы счисления. От их особенностей зависит наглядность представления чисел и сложность выполнения арифметических операций.

В настоящее время наибольшее распространение получила *позиционная система* счисления. Эта система называется позиционной потому, что в зависимости от места расположения цифры в числе ее значение различно. Позиционной является десятичная система, в которой используется десять различных символов, называемых цифрами: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Например, число 2222 состоит из двоек, но значение каждой из них различно. Так самая правая двойка означает две единицы, вторая справа – два десятка, третья – две сотни, четвертая – две тысячи. Каждая новая позиция изменяет значение в 10 раз. Количество символов, используемых в системе счисления, принято называть *основанием системы*. Если основание системы счисления обозначить через S , то количество символов в системе будет $0, 1, 2, \dots, (S - 1)$. В общем виде в системе счисления с основанием S любое число может быть представлено в виде

$$x = a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s^1 + a_0 s^0 + a_{-1} s^{-1} + a_{-2} s^{-2} + \dots,$$

где в качестве коэффициентов a могут быть любые из S цифр, а $n = R-1$. Здесь R – количество разрядов, занимаемых числом x . Например, запись

$$5 \cdot 10^3 + 4 \cdot 10^2 + 2 \cdot 10^1 + 7 \cdot 10^0 + 6 \cdot 10^{-1} + 1 \cdot 10^{-2}$$

определяет число 5427.61.

Принято представлять числа как последовательность цифр

$$x = a_n a_{n-1} \dots a_1 a_0 . a_{-1} a_{-2} \dots$$

В этой последовательности запятая отделяет целую часть от дробной. Если число целое, то запятая в конце не ставится. Позиции цифр называют *разрядами*. В позиционной системе значение каждого соседнего разряда слева больше разряда справа в S раз, то есть в число раз равно основанию системы счисления.

В вычислительной технике применяют



- двоичную,
- восьмеричную,
- шестнадцатеричную

системы счисления. Наибольшее распространение получила двоичная система счисления. В этой системе любое число может быть представлено последовательностью двух цифр 0 и 1.

$$x = b_m b_{m-1} \dots b_1 b_0, b_{-1} b_{-2} \dots,$$

где b либо 0 либо 1. Например, 0110100111,011.

Эта запись соответствует сумме степеней числа 2 взятых с указанными в ней коэффициентами:

$$x = a_m 2^m + a_{m-1} 2^{m-1} + \dots + a_1 2^1 + a_0 2^0 + a_{-1} 2^{-1} + a_{-2} 2^{-2} + \dots$$

Например, двоичное число $(11011101.011)_2$ может быть представлено как

$$1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3}$$

Это соответствует числу в десятичной системе счисления

$$128 + 64 + 0 + 16 + 8 + 4 + 0 + 1 + 0 + 0.25 + 0.125 = (221.375)_{10}.$$

В восьмеричной системе используется восемь символов : 0,1,2,3,4,5,6,7 , в шестнадцатеричной – 16: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F, где A=10, B=11, C=12, D=13, E=14, F=15.

Число в восьмеричной системе представляется как

$$x = a_n 8^n + a_{n-1} 8^{n-1} + \dots + a_1 8^1 + a_0 8^0 + a_{-1} 8^{-1} + a_{-2} 8^{-2} + \dots$$

В шестнадцатеричной системе:

$$x = a_n 16^n + a_{n-1} 16^{n-1} + \dots + a_1 16^1 + a_0 16^0 + a_{-1} 16^{-1} + a_{-2} 16^{-2} + \dots$$

Рассмотрим системы с основанием 2,8,10,16.

Рассмотрим на примерах как перевести двоичное число в более привычную десятичную систему счисления.

$$(101101)_2 = 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 32 + 8 + 4 + 1 = (45)_{10}$$

$$(10101,1101)_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} =$$

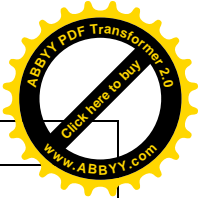
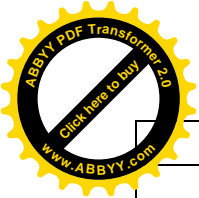
$$= 16 + 4 + 1 + 1/2 + 1/4 + 1/16 = 21 + 13/16 = (21,8125)_{10}$$

Существенным недостатком двоичной системы счисления является громоздкая запись чисел. Для упрощения записи двоичных чисел могут быть использованы восьмеричная и шестнадцатеричная системы счисления. Запись чисел в двоичной, восьмеричной и шестнадцатеричной системах счисления приведена в таблице 1.1.

Двоичная система связана с восьмеричной и шестнадцатеричной соотношениями: $2^3=8$ и $2^4=16$, т.е. цифры восьмеричной системы можно представить двоичными триадами, шестнадцатеричной – тетрадами, что облегчает взаимный перевод.

Таблица 1.1

| ЧИСЛО | | | | |
|------------|----------|--------------|----------|-------------------|
| десятичное | двоичное | Восьмеричное | двоичное | шестнадцатеричное |
| 0 | 0000 | 0 | 0000 | 0 |
| 1 | 0001 | 1 | 0001 | 1 |



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

2.2. Перевод чисел из восьмеричной и шестнадцатеричной системы счисления в десятичную

Правила перевода чисел из восьмеричной и шестнадцатеричной системы счисления в десятичную – просты. Следует числа 8 и 16 представить как 2^3 и 2^4 соответственно. Тогда восьмеричное число в двоичном исчислении выглядит как

$$x = a_n 2^{3n} + a_{n-1} 2^{3(n-1)} + \dots + a_1 2^3 + a_0 2^0 + a_{-1} 2^{3(-1)} + a_{-2} 2^{3(-2)} + \dots,$$

а шестнадцатеричное число как

$$x = a_n 2^{4n} + a_{n-1} 2^{4(n-1)} + \dots + a_1 2^4 + a_0 2^0 + a_{-1} 2^{4(-1)} + a_{-2} 2^{4(-2)} + \dots.$$

Например,

$$(B2E.4)_{16} = 11 * 16^2 + 2 * 16^1 + 14 * 16^0 + 4 * 16^{-1} = (2862.25)_{10}$$

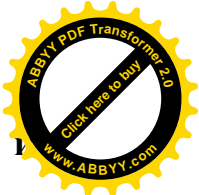
2.3. Перевод чисел из восьмеричной и шестнадцатеричной системы счисления в двоичную

Для перевода *восьмеричного* числа в двоичное необходимо заменить каждую цифру восьмеричного числа *трехразрядным* двоичным, при переводе *шестнадцатеричного* числа заменить каждую цифру шестнадцатеричного числа *четырёхразрядным* двоичным. При этом в последней слева группе отбрасываются нули слева, а в последней справа группе отбрасываются нули справа. Например, восьмеричное число $(305.4)_8$ в двоичной системе будет выглядеть как

$$\frac{3}{011} \frac{0}{000} \frac{5}{101} \cdot \frac{4}{100} = (11000101.1)_2$$

шестнадцатеричное число $(B2E.4)_{16}$

$$\frac{B}{1011} \frac{2}{0010} \frac{E}{1110} \cdot \frac{4}{1000} = (101100101110.1)_2.$$



2.4. Перевод чисел из двоичной системы счисления в восьмеричную и шестнадцатеричную

Для перехода с двоичной системы в восьмеричную (шестнадцатеричную) систему счисления следует:

- двигаясь от точки, разделяющую целую и дробную части, влево и вправо, разбивать двоичное число на группы по *три (четыре)* разряда. Недостающие разряды чисел в самой левой и самой правой группе следует представлять нулями;

- каждую группу из *трех (четырех)* двоичных разрядов заменять соответствующей цифрой восьмеричной (шестнадцатеричной) системы счисления.

Например, двоичное число 1101111001.1101 при переводе в восьмеричную систему представляется группами по три разряда с заполнением нулями недостающих разрядов в самой левой и в самой правой группе

$$\frac{001\ 101\ 111\ 001}{1\ 5\ 7\ 1} \cdot \frac{110\ 100}{6\ 4} = (1571.64)_8$$

При переводе этого числа в шестнадцатеричную систему оно разбивается на группы по четыре разряда с заполнением нулями недостающих разрядов в самой левой и самой правой группе.

$$\frac{0011\ 0111\ 1001}{3\ 7\ 9} \cdot \frac{1101}{D} = (379.D)_{16}$$

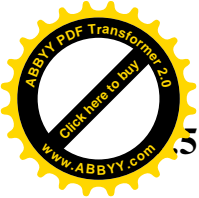
В современных компьютерах используется двоичная система счисления. Шестнадцатеричная (реже восьмеричная) система используется при составлении программ для получения более короткой и удобной записи двоичных кодов команд, так как эта система не требует специальных операций по переводу в двоичную систему.

Числовые данные, необходимые для решения задач, вводятся в компьютер в привычной, десятичной форме, перевод же в двоичную форму осуществляется программным обеспечением компьютера. Оно же осуществляет обратный перевод при выводе результатов.

Например, $(110100101)_2 \rightarrow (110\ 100\ 101)_2 \rightarrow (645)_8$
 $(11011101)_2 \rightarrow (011\ 011\ 101)_2 \rightarrow (335)_8$
 $(101111101111)_2 \rightarrow (1011\ 1110\ 1111)_2 \rightarrow (BEF)_{16}$
 $(10111011000)_2 \rightarrow (0101\ 1101\ 1000)_2 \rightarrow (5D8)_{16}$

2.5. Перевод чисел из десятичной системы счисления в другие

При переводе чисел с *десятичной* системы в двоичную (восьмеричную, шестнадцатеричную) систему счисления используют различные правила перевода для *целой* и *дробной* части.



5.1. Перевод целых десятичных чисел в другие системы счисления

Для того чтобы перевести целое число из десятичной системы счисления в другую необходимо выполнить следующие действия:

- 1) поделить данное число на основание новой системы счисления;
- 2) перевести остаток от деления в новую систему счисления; получается младший разряд нового числа;
- 3) если частное от деления больше основания новой системы, продолжать деление, как указано в п.1; новый остаток, переведенный в новую систему счисления, дает второй разряд числа и т.д.

Пример. Перевести число 256 из десятичной системы счисления в восьмеричную. (Далее, будем записывать кратко $(256)_{10} \rightarrow (?)_8$)

Решение

$$\begin{array}{r}
 256 \quad | \quad 8 \\
 \hline
 \underline{-256} \quad 32 \quad | \quad 8 \\
 0 \quad \underline{-32} \quad 4 \\
 0
 \end{array}$$

Ответ: $(256)_{10} \rightarrow (400)_8$. Сделаем проверку, чтобы убедиться в правильности перевода: $(400)_8 = 4 \cdot 8^2 + 0 \cdot 8^1 + 0 \cdot 8^0 = 256 + 0 + 0 = (256)_{10}$

Пример. Пусть необходимо выполнить перевод $(397)_{10} \rightarrow (?)_{16}$

Решение:

$$\begin{array}{r}
 397 \quad | \quad 16 \\
 \hline
 \underline{-384} \quad 24 \quad | \quad 16 \\
 13 \quad \underline{-16} \quad 1 \\
 8
 \end{array}$$

Ответ: $(397)_{10} \rightarrow (18D)_{16}$.

Проверка: $(18D)_{16} = 1 \cdot 16^2 + 8 \cdot 16^1 + 13 \cdot 16^0 = 256 + 128 + 13 = (397)_{10}$

Пример. Выполнить преобразование $(25)_{10} \rightarrow (?)_2$

Решение

$$\begin{array}{r}
 25 \quad | \quad 2 \\
 \hline
 \underline{-24} \quad 12 \quad | \quad 2 \\
 1 \quad \underline{-12} \quad 6 \quad | \quad 2 \\
 0 \quad \underline{-6} \quad 3 \quad | \quad 2 \\
 0 \quad \underline{-2} \quad 1 \\
 1
 \end{array}$$

Ответ: $(25)_{10} \rightarrow (11001)_2$. Сделаем проверку:

$(25)_{10} = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 16 + 8 + 1 = (25)_{10}$



2.5.2. Перевод дробных чисел

При переводе вещественных чисел, необходимо отдельно переводить целую и дробную часть. Правило перевода дробей из одной системы счисления в другую заключается в выполнении следующих действий:

- 1) умножить дробную часть числа на основание системы счисления;
- 2) в полученном произведении выделить целую часть числа (это будет старшим разрядом искомого числа);
- 3) дробную часть произведения снова умножить на основание новой системы счисления; целая часть произведения будет следующим разрядом дробной части искомого числа;
- 4) п.3 повторить до получения необходимого количества разрядов искомого числа.

Пример. Выполнить перевод $(0,784)_{10} \rightarrow (?)_2$, $0,6125_{10} \rightarrow (?)_8$ и $(0,378)_{10} \rightarrow (?)_{16}$. Оставить четыре знака после запятой.

| | | |
|--|---|---|
| $\begin{array}{r l} 0,784 & \\ \times 2 & \\ \hline 1 & 568 \\ \times 2 & \\ \hline 1 & 136 \\ \times 2 & \\ \hline 0 & 272 \\ \times 2 & \\ \hline 0 & 544 \\ \times 2 & \\ \hline 1 & 088 \end{array}$ | $\begin{array}{r l} 0,6125 & \\ \times 8 & \\ \hline 4 & 9000 \\ \times 8 & \\ \hline 7 & 2000 \\ \times 8 & \\ \hline 1 & 6000 \\ \times 8 & \\ \hline 4 & 8000 \end{array}$ | $\begin{array}{r l} 0,378 & \\ \times 16 & \\ \hline 6 & 046 \\ \times 16 & \\ \hline 0 & 768 \\ \times 16 & \\ \hline 12 & 288 \\ \times 16 & \\ \hline 4 & 608 \end{array}$ |
|--|---|---|

Результат получаем, читая цифры сверху вниз:

$$(0,784)_{10} \rightarrow (0,1100)_2$$

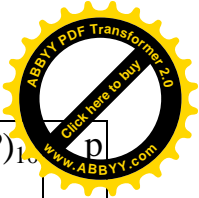
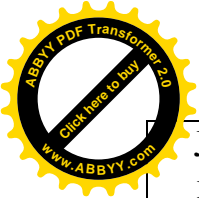
$$0,6125_{10} \rightarrow (0,4714)_8$$

$$(0,378)_{10} \rightarrow (0,60C4)_{16}$$

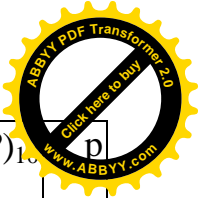
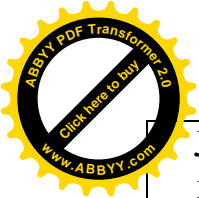
Индивидуальные задания по теме 1

Выполнить перевод из одной с/с в другую, сделать проверку.

| № п/ п | X ₂ →(?) ₁₀ | Y ₂ →(?) ₈ , Y ₂ →(?) ₁₆ | Z ₁₀ →(?) ₂ , z ₁₀ →(?) ₈ , Z ₁₀ →(?) ₁ 6 | S ₈ →(?) ₁₆ | U ₁₆ →(?) ₈ | W _p →(?) ₁₀ | p |
|--------------|-----------------------------------|---|--|-----------------------------------|-----------------------------------|-----------------------------------|----------|
| 1 | 11011,1101 | 111101 | 235,647 | 735 | 6D3,2B | 122 | 3 |
| 2 | 11101,0011 | 110011 | 417,813 | 156 | 95C,F7 | 322 | 4 |
| 3 | 10101,1111 | 101111 | 176,451 | 376 | 3D6,B5 | 341 | 5 |



| № Π/ Π | $X_2 \rightarrow (?)_{10}$ | $Y_2 \rightarrow (?)_8,$ $Y_2 \rightarrow (?)_{16}$ | $Z_{10} \rightarrow (?)_2,$ $z_{10} \rightarrow (?)_8,$ $Z_{10} \rightarrow (?)_1$ 6 | $S_8 \rightarrow (?)_{16}$ | $U_{16} \rightarrow (?)_8$ | $W_p \rightarrow (?)_1$ | p |
|-----------------------|--|--|---|--|--|---|----------|
| 4 | 10001,1001 | 100001 | 156,848 | 176 | 63D,C1 | 405 | 6 |
| 5 | 10011,0001 | 110010 | 601,003 | 641 | 34F,A8 | 501 | 7 |
| 6 | 11001,1000 | 101000 | 126,012 | 622 | 13A,DE | 608 | 9 |
| 7 | 11110,1000 | 111000 | 325,632 | 252 | D13,2B | 271 | 9 |
| 8 | 10111,0111 | 110111 | 484,191 | 441 | 9E7,2D | 143 | 5 |
| 9 | 11100,1110 | 101110 | 681,534 | 634 | 3CD,1E | 215 | 6 |
| 10 | 11000,1010 | 001010 | 183,654 | 134 | 5CA,1E | 306 | 7 |
| 11 | 10100,1001 | 010001 | 273,021 | 221 | C3F,E7 | 201 | 3 |
| 12 | 10010,0101 | 101010 | 289,713 | 271 | C1F,D5 | 331 | 4 |
| 13 | 11111,0101 | 110101 | 259,527 | 252 | 986,37 | 320 | 5 |
| 14 | 10111,1111 | 100111 | 201,113 | 203 | 689,37 | 103 | 6 |
| 15 | 11111,1101 | 111101 | 114,453 | 153 | 467,EA | 150 | 7 |
| 16 | 10101,0101 | 100101 | 176,724 | 164 | ADE,71 | 212 | 3 |
| 17 | 11010,1001 | 101001 | 106,398 | 632 | E73,DE | 102 | 4 |
| 18 | 100010,011 | 100011 | 982,754 | 275 | D19,AB | 204 | 5 |
| 19 | 101101,011 | 101001 | 417,983 | 415 | BE7,D9 | 410 | 6 |
| 20 | 100111,011 | 100101 | 742,429 | 242 | 1CD,8E | 261 | 7 |
| 21 | 111011,001 | 110101 | 659,832 | 652 | 5FA,1E | 102 | 3 |
| 22 | 101110,101 | 011101 | 286,327 | 262 | C38,97 | 232 | 4 |
| 23 | 111001,011 | 100011 | 687,321 | 672 | C9F,D5 | 241 | 5 |
| 24 | 10101,0101 | 101010 | 945,325 | 453 | 9A6,B7 | 451 | 6 |
| 25 | 110101,100 | 110110 | 467,894 | 674 | D89,F5 | 653 | 7 |
| 26 | 10100,1010 | 101101 | 395,532 | 355 | 469,FA | 210 | 3 |
| 27 | 10111,0111 | 101111 | 984,291 | 421 | 8DE,7A | 302 | 4 |
| 28 | 110001,110 | 110011 | 671,544 | 614 | E79,8E | 413 | 5 |
| 29 | 11010,1111 | 110111 | 883,154 | 354 | 6D8,BA | 305 | 6 |
| 30 | 111011,001 | 111000 | 473,621 | 476 | E5C,A4 | 620 | 7 |
| 31 | 101111,001 | 101100 | 989,413 | 414 | 9D6,8C | 121 | 3 |
| 32 | 10110,1001 | 101100 | 759,527 | 755 | 679,82 | 333 | 4 |



| N_0 Π Π | $X_2 \rightarrow (?)_{10}$ | $Y_2 \rightarrow (?)_8,$ $Y_2 \rightarrow (?)_{16}$ | $Z_{10} \rightarrow (?)_2,$ $z_{10} \rightarrow (?)_8,$ $Z_{10} \rightarrow (?)_1$ 6 | $S_8 \rightarrow (?)_{16}$ | $U_{16} \rightarrow (?)_8$ | $W_p \rightarrow (?)_1$ | p |
|-------------------------|----------------------------|--|---|----------------------------|----------------------------|-------------------------|----------|
| 33 | 10111,1101 | 111110 | 201,213 | 203 | A4F,CB | 423 | 5 |
| 34 | 11110,0001 | 111100 | 614,353 | 643 | FCA,D8 | 553 | 6 |
| 35 | 101110,011 | 101111 | 196,754 | 165 | AE7,89 | 105 | 7 |



ЛЕКЦИЯ №3

3. АЛГОРИТМ И ПРОГРАММА. ВИДЫ ВЫЧИСЛИТЕЛЬНЫХ ПРОЦЕССОВ

Любая задача, решаемая с помощью компьютера, должна пройти следующие этапы:

1. Постановка или формулировка задачи. На этом этапе излагается суть задачи, определяются исходные данные и формулируется цель.

1. *Разработка математической модели* исследуемого процесса или явления, сформулированного в задаче. На этом этапе рассматриваемый в задаче физический процесс, социальные явления или экономические проблемы представляются в виде математических выражений, формул, уравнений, отображающих поведение объекта исследования. При этом принимается решение о том, какие факторы, оказывающие влияние на поведение объекта исследования, учесть и в какой мере, а какими пренебречь.

2. *Выбор метода решения.* На этом этапе выбирается один из нескольких возможных методов решения задачи. При этом оцениваются достоинства и недостатки, рассматриваемых методов.

3. *Разработка алгоритма вычисления.* На этом этапе создается план вычислительного процесса, производится его оптимизация с позиций минимального расхода машинного времени и памяти компьютера.

4. *Написание программы* на одном из алгоритмических языков.

5. *Отладка и выполнение программы.* На этом этапе устраняются синтаксические и логические ошибки, допущенные при написании программы.

6. *Анализ полученных результатов.* Это важный этап работы. Он подводит итог исследования. На этом этапе сравниваются результаты вычислений с результатами экспериментальных исследований, уточняется математическая модель объекта. При этом устанавливается степень влияния тех или иных факторов с целью учета неучтенных, которые оказали в результате существенное влияние, и пренебрежения учтенных из-за их незначительного влияния. Дополнительно уточняются исходные данные, метод вычислений и др.

Первые три и последний этап относятся к прикладным дисциплинам, в которых осуществляется постановка задачи и анализ поведения объекта исследования. Четвертый, пятый и шестой этапы являются областью дисциплины «Информатика».

3.1. Алгоритм

Алгоритм – это точное и строгое предписание вычислительных операций, образующих вычислительный процесс, от ввода исходных данных и до получения конечного результата.

Проще говоря, алгоритм это план вычислений. В вычислительной технике принято изображать алгоритм графическим методом. В этом случае все операции вычислительного процесса представляются в виде графических, условных обозначений, называемых *блоками*. Связывая такие блоки линиями связи, можно создавать схемы вычислений. Схемы, состоящие из совокупности блоков, получили название *блок – схем*. На изображение блоков установлен государственный стандарт (ГОСТ). В таблице 7.1 приводится изображение основных блоков, применяемых для создания блок–схем. Блоки операций в алгоритмах принято размещать в направлении сверху вниз.

Таблица 4.1 – Перечень операционных блоков

| № п.п. | Условное обозначение | Наименование | Комментарий |
|--------|----------------------|---------------------------|---|
| 1 | | Пуск, останов | Начало и конец алгоритма |
| 2 | | Процесс | Вычислительное действие, совокупность вычислительных действий |
| 3 | | Решение | Проверка условия ветвления, выбор дальнейшего направления процесса решения в зависимости от условия |
| 4 | | Модификация | Заголовок цикла, проверка условий цикла |
| 5 | | Ввод, вывод Данных | Ввод исходных данных, вывод результатов |
| 6 | | Предопределенный процесс | Использование ранее созданных алгоритмов (подпрограмм) |
| 7 | | Документ | Вывод данных на печать |
| 8 | | Соединитель страничный | Разрыв линий потока в пределах одной страницы |
| 9 | | Соединитель межстраничный | Перенос линий потока на другую страницу |
| 10 | | Узел | Слияние линий потока |

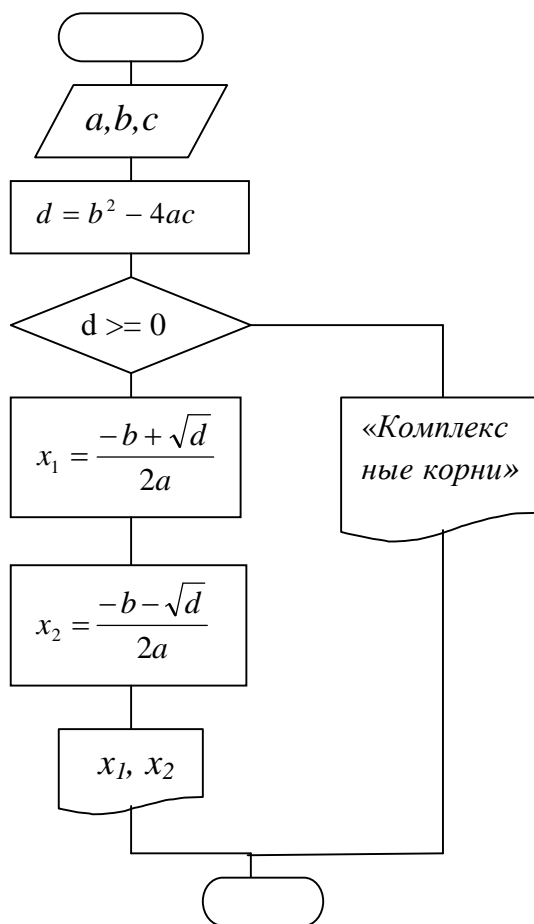


Рис. 3.1 Пример выполнения блок–схемы для вычисления корней квадратного уравнения

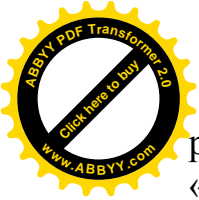
Рекомендуется принимать соотношение ширины блока к его высоте равной 1.5. Хорошим стилем считается *вертикальное* начертание линий связи в местах входа в блок и выхода из него. Линии связи, направленные *снизу вверх* и *справа налево* **обязательно** должны заканчиваться стрелками. В других случаях это необязательно.

3.1.1. Требования к алгоритму

К алгоритму предъявляются требования, в соответствии с которыми он должен обладать:

- *определенностью*, то есть быть *однозначно* понятым различными специалистами или, то есть *не иметь двойного смысла*;
- *массовостью*, что означает быть применимым для целой группы однотипных задач;
- *результативностью*, то есть приводить к конечному результату во всех случаях.

Следует заметить, что разработка алгоритма является самой трудоемкой и интеллектуальной частью работы, требующей высокого уровня логического мышления. Именно алгоритм определяет профессиональный уровень



разработчика. Поэтому очень важно с первых же шагов изучения дисциплины «Информатика» развивать логическое мышление, начиная с простейших задач.

3.2. Программа

Программа – это записанный на одном из алгоритмических языков алгоритм вычислений.

Следует четко представлять, что компьютер сам по себе ничего не «умеет» делать. Все его способности сводятся к выполнению элементарных операций. Но и эти операции он может выполнять лишь только тогда, когда получит соответствующие *инструкции* на понятном для себя языке. Поэтому для реализации каких-либо вычислений, необходимо составить для компьютера перечень инструкций, который называют *программой*.

Для написания программ в настоящее время разработаны алгоритмические языки высокого уровня. Они позволяют записывать различные операции, предусмотренные алгоритмом, по определенным правилам, с использованием определенных слов и фраз. Количество их ограничено, слова не имеют синонимов, а фразы имеют только один смысл, определенный языком. Вследствие этого алгоритмические языки лаконичны и имеют объем слов значительно меньший, чем у разговорных языков.

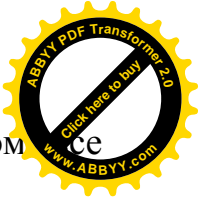
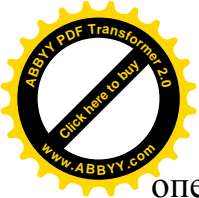
Язык программирования использует понятные человеку слова и фразы (в основном английские). Но он не «понятен» компьютеру. Ведь вся информация, обрабатываемая компьютером, кодируется *двоичным* или, как еще его называют, *машинным* кодом. Следовательно, программу, написанную на алгоритмическом языке, необходимо перевести на машинный язык. Эту функцию выполняют специальные программы – *компиляторы*. В упрощенном виде программа представляет собой описание, «фотографирование» алгоритма средствами алгоритмического языка. Написание ее, в основном, представляет собой больше формальную, чем интеллектуальную работу и не составляет большого труда.

Здесь очень важно понять, что *сначала необходимо разработать алгоритм, а затем по нему писать программу*. Практика написания программы без алгоритма весьма порочна и даже вредна. Так можно поступать в тех случаях, когда задача не громоздка и проста. Здесь последовательность операций можно держать в памяти человека. Но, если процесс вычислений объемный, с большим количеством ветвлений, то удержать его в голове просто невозможно. Здесь обойтись без алгоритма, как наглядной формы изображения вычислений, нельзя. Поэтому следует с самого начала изучения «Информатики» приучить себя на простых задачах разрабатывать алгоритм, а затем по нему писать программу, а не наоборот.

3.3. Виды вычислительных процессов

Различают три вида вычислительных процессов:

- *линейные* или *простые*,
- *разветвляющиеся*,
- *циклические*.



Линейным называют такой вычислительный процесс, при котором операции выполняются последовательно и в одном направлении.

Пример: вычислить значение функции $y = k(ax^2 + bx - c)$ при различных значениях параметров a, b, c, k, x .

На рис. 3.2. приведен алгоритм линейного вычислительного процесса. Как видно из рисунка все операции алгоритма выполняются последовательно и в одном направлении. Алгоритм применим для вычисления функции y при любых значениях a, b, c, k, x , следовательно, он обладает *массовостью*, не может быть двояко истолкован (обладает *определенностью*), всегда приводит к конечному результату (*результативный*).

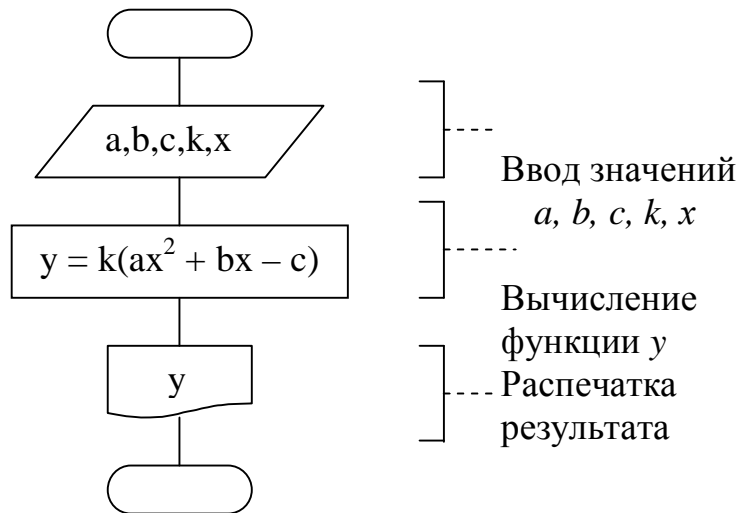
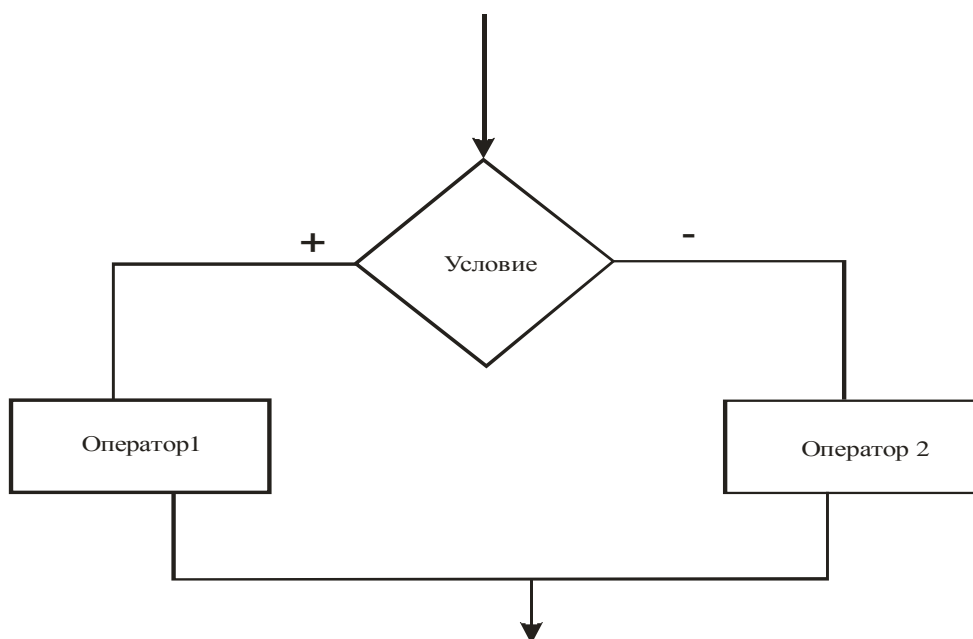


Рис. 3.2. Линейный вычислительный процесс

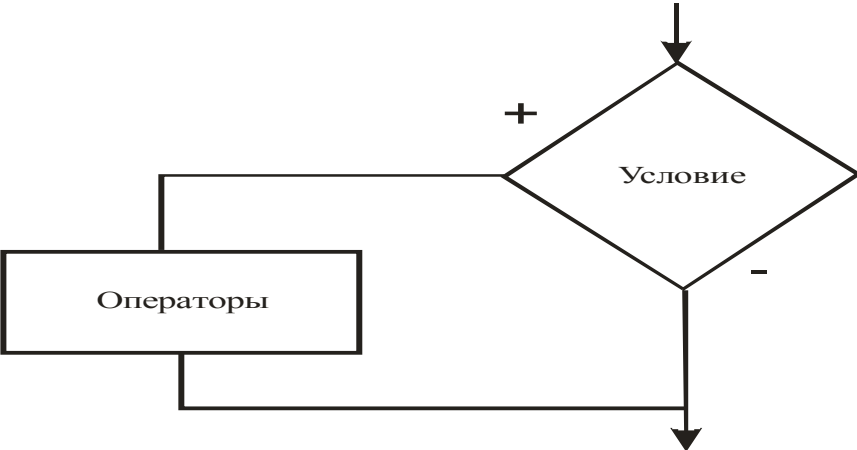
Разветвляющимся называют такой вычислительный процесс, у которого на отдельных участках часть операций могут, в зависимости от какого-либо условия, выполняться различными путями, то есть в одном или другом направлении.

Разновидности разветвлённого процесса

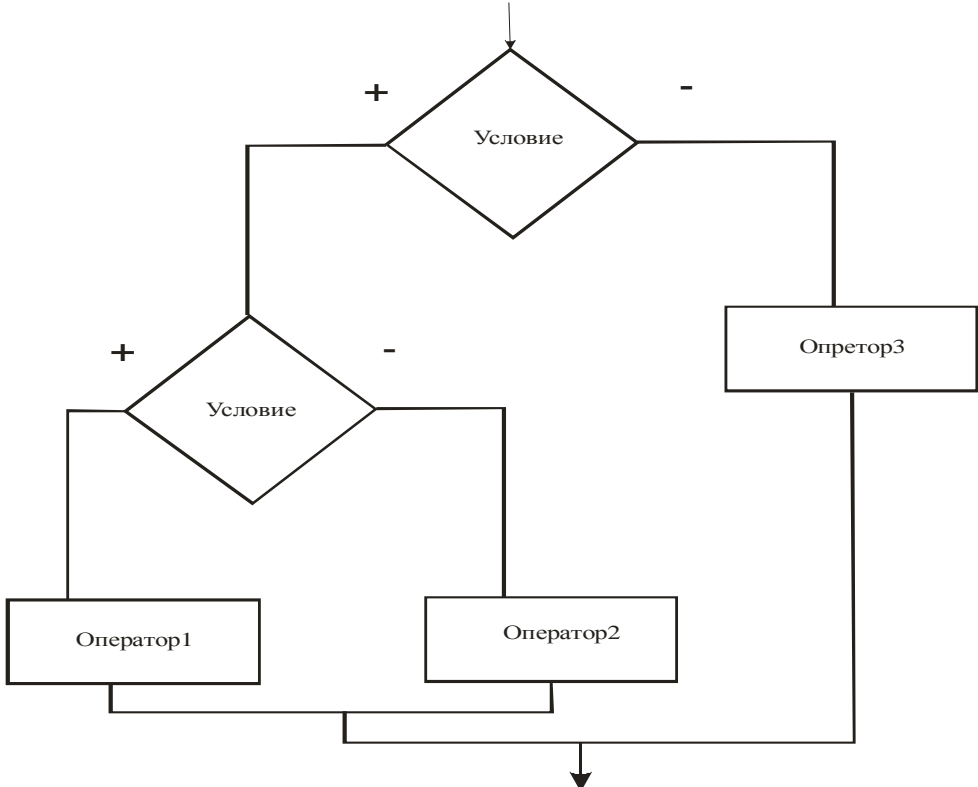




1.



2.



Примером разветвляющего вычислительного процесса может служить алгоритм вычисления корней квадратного уравнения, приведенного на рис.3.3. Из того алгоритма видно, что после проверки условия $d \geq 0$, в зависимости от результата, дальнейшие вычисления могут происходить по ветви (+) или ветви (-).

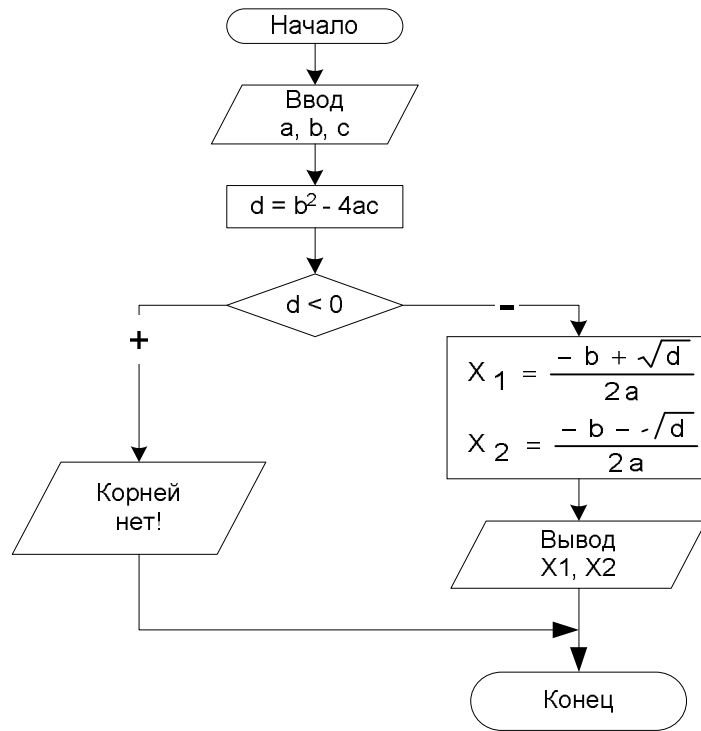


Рис. 3.3. Разветвляющийся вычислительный процесс

Алгоритмы циклической структуры

Циклическим называют такой вычислительный процесс, в котором многократно (циклически) повторяется некоторая операция или группа операций, называемая *телом цикла*.

Выполнение любого цикла требует реализации следующих этапов:

- *подготовка цикла*, предполагающая инициализацию (установления начального значения) регулярно изменяющейся переменной, называемой *параметром цикла*;
- *управление циклом*, предусматривающее проверку условия выполнения цикла;
- *выполнение тела цикла*, означающее выполнение одних и тех же операций для каждого нового значения параметра цикла;
- *продвижение цикла*, предписывающее изменение параметра цикла на одну и ту же величину, называемую *шагом*, перед каждым новым выполнением тела цикла.

Пример. Вычислить значения функции $y = ax^2$, если $a = 1.8$, а x изменяется от $x_n = -3.2$ до $x_k = 4.3$ с шагом $h = 0.1$.

Блок – схема вычислений (рис.7.4) предполагает использование циклической операции.

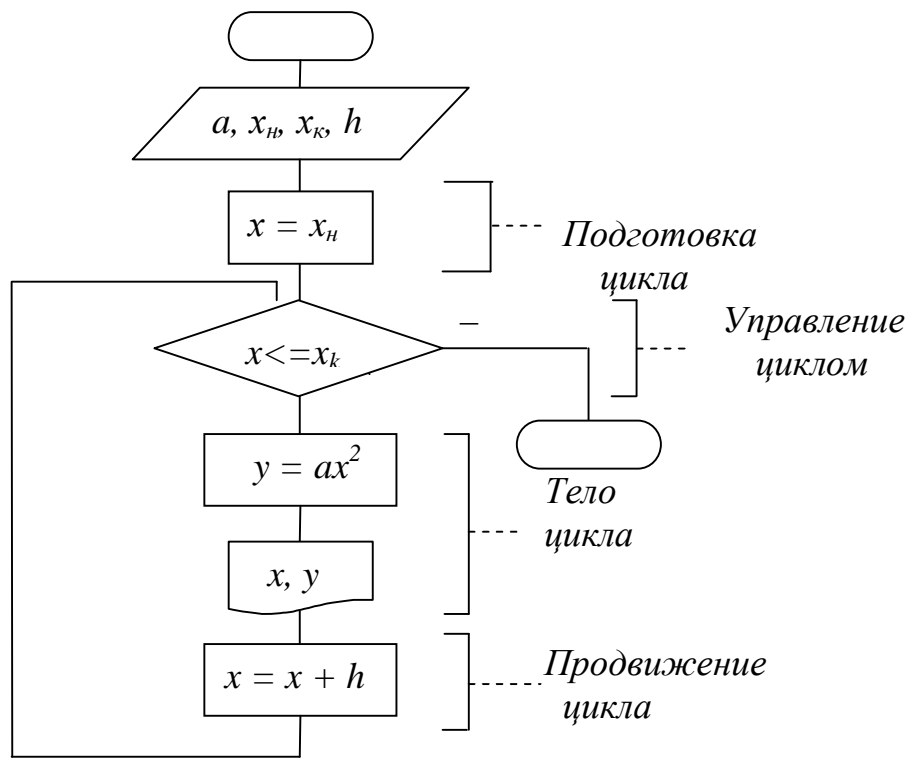


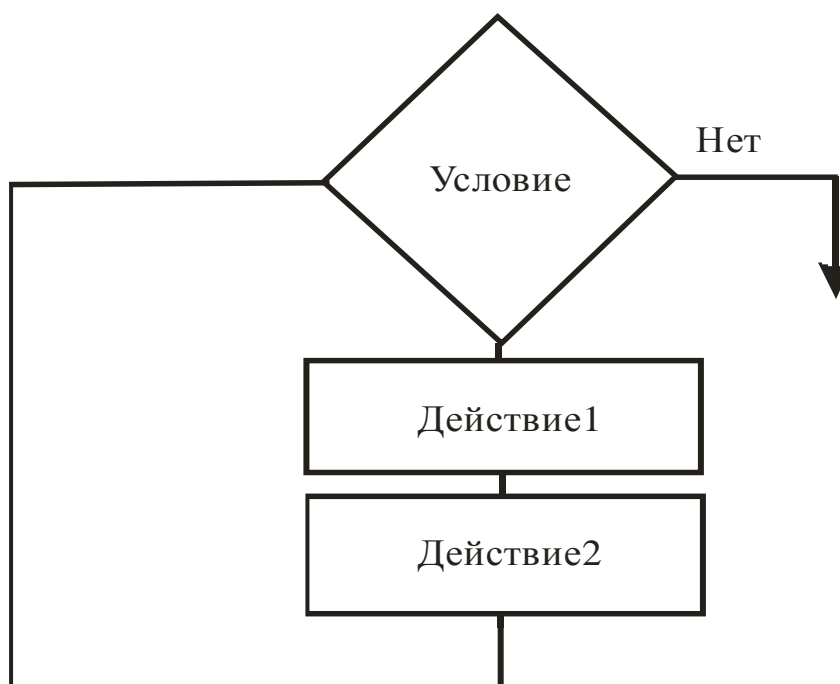
Рис. 3.4. Блок – схема циклического вычислительного процесс

Циклическая операция будет выполняться до тех пор, пока будет выполняться условие $x \leq x_k$.

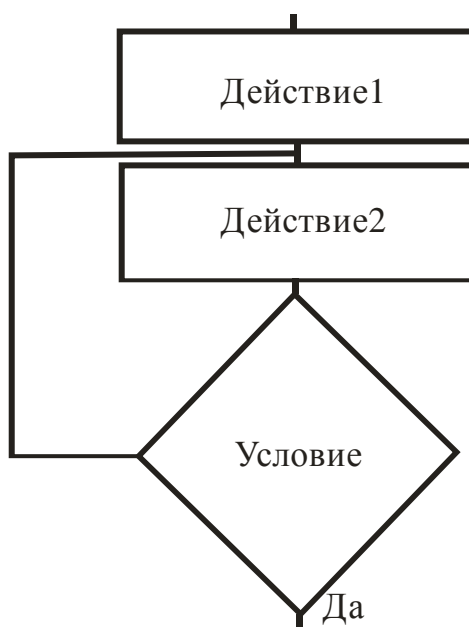
Следует отметить, что в большинстве реальных задач алгоритмы представляются как комбинация различных видов вычислительных процессов.



1. Цикл с предусловием



2. Цикл с постусловием



Отличие обоих циклов заключается в:

- В цикле с предусловием условие проверяется до тела цикла, в цикле с постусловием – после тела цикла
- В цикле с постусловием тело цикла выполняется хотя бы один раз, в цикле с предусловием может не выполняться вообще
- В цикле с предусловием проверяется условие продолжения цикла, в цикле с постусловием – условие выхода из цикла



Циклы с известным и неизвестным числом повторений

Циклы с неизвестным числом повторений характеризуются тем, что не известно сколько раз повторится тело цикла. Поэтому вместо конечного значения существует условие окончания цикла, то есть цикл заканчивается, если выполняется некоторое условие.

Пример 6

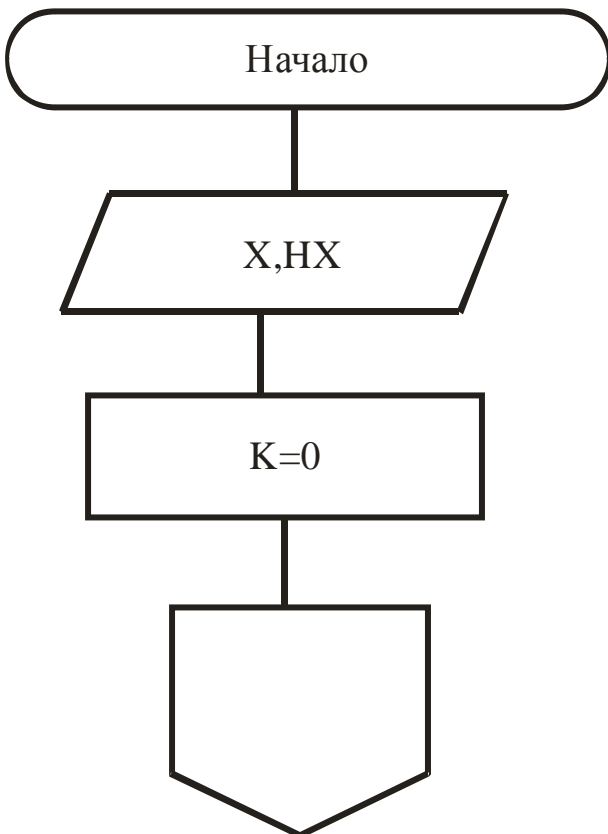
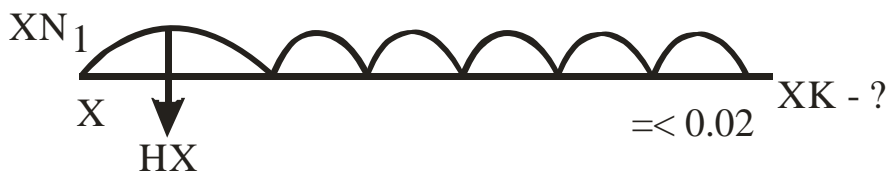
Вычислить значение Z по формуле:

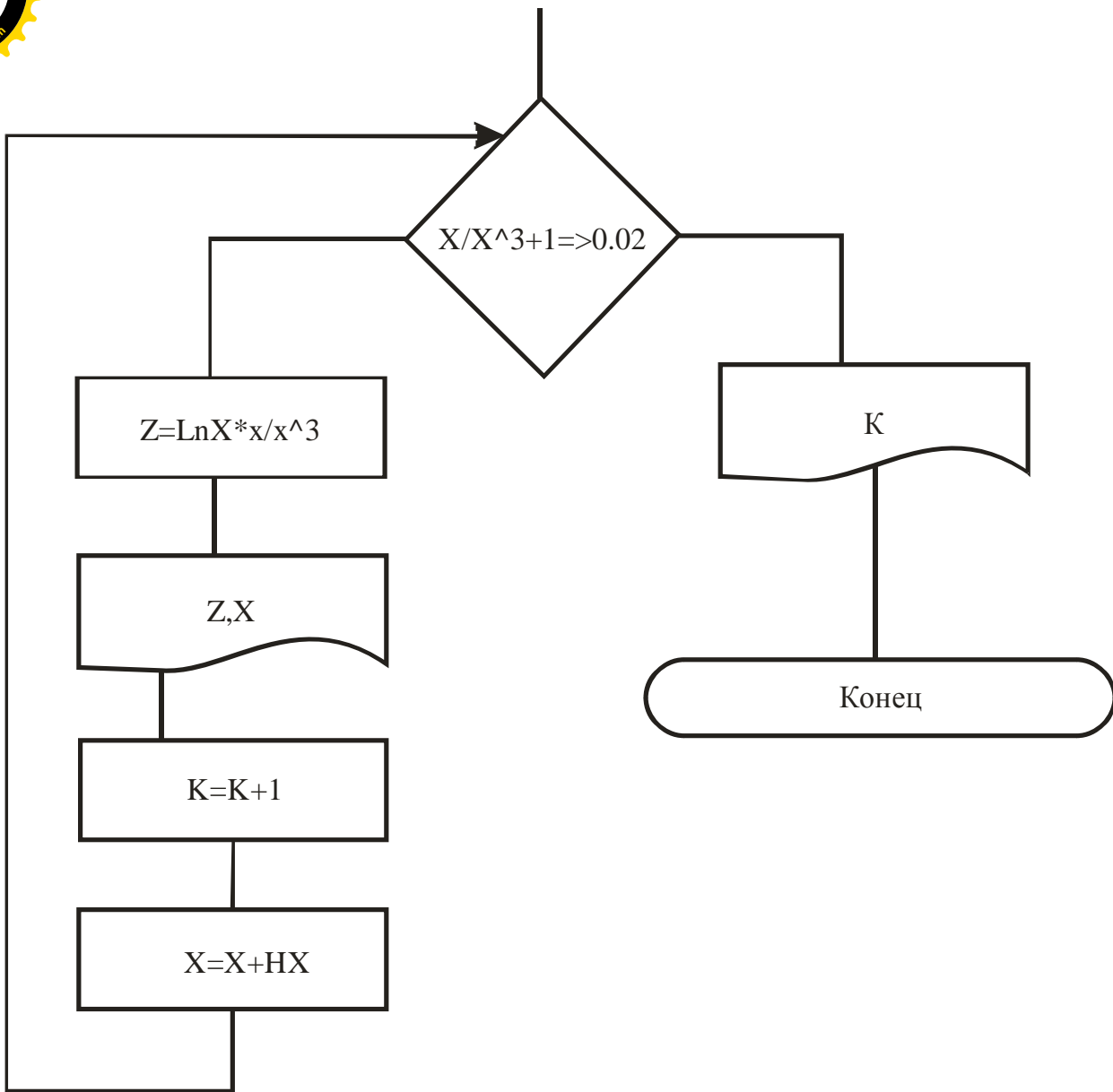
$$Z = \text{Ln}x * \sqrt{\frac{x}{x^3 + 1}}$$

$x = 1$; hx (шаг) = 0,5.

Считать Z до тех пор, пока подкоренное выражение будет $\Rightarrow 0,02$.

Определить величину K – количество вычисленных Z .





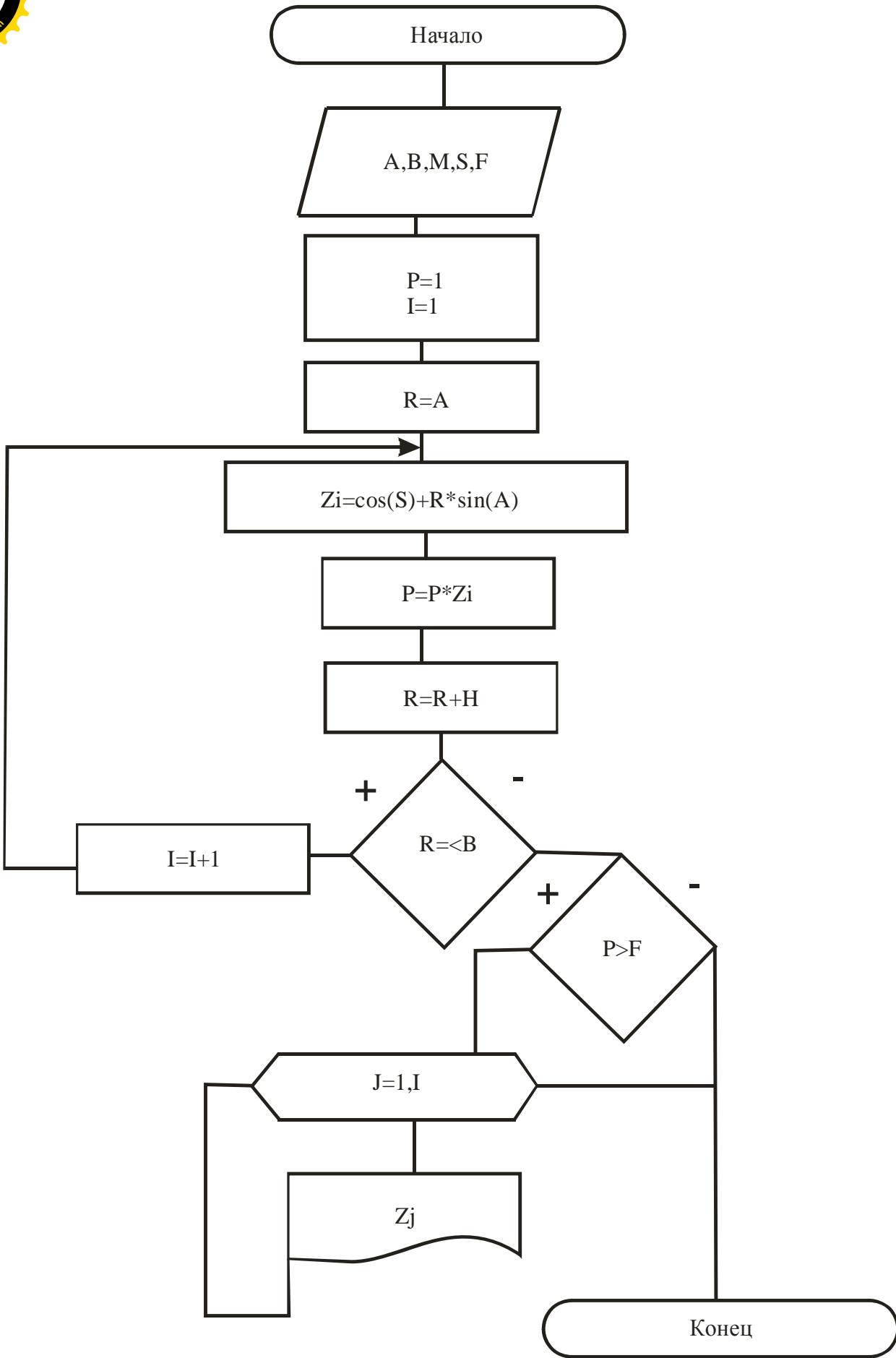
Целая величина R изменяется от A до B шагом H. Вычислить для каждого Z значение $Z = \cos(S) + R * \sin(A)$.

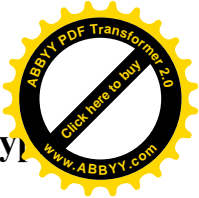
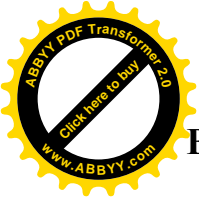
Найти произведение значений Z_i , если оно будет $>F$, то Z не печатать.

Начальные установки:

1. S. S=0 сумма
2. P. P=1 произведение
3. K=0

Если известен отрезок – цикл с известным числом повторений.





Вложенные структуры. Программирование процессов сложной структуры

$$y = \begin{cases} \sqrt[3]{x^2 - ax}, & b > 5, S = 0 \\ \frac{b}{x^2 + 1}, & b \leq 5 \end{cases}$$

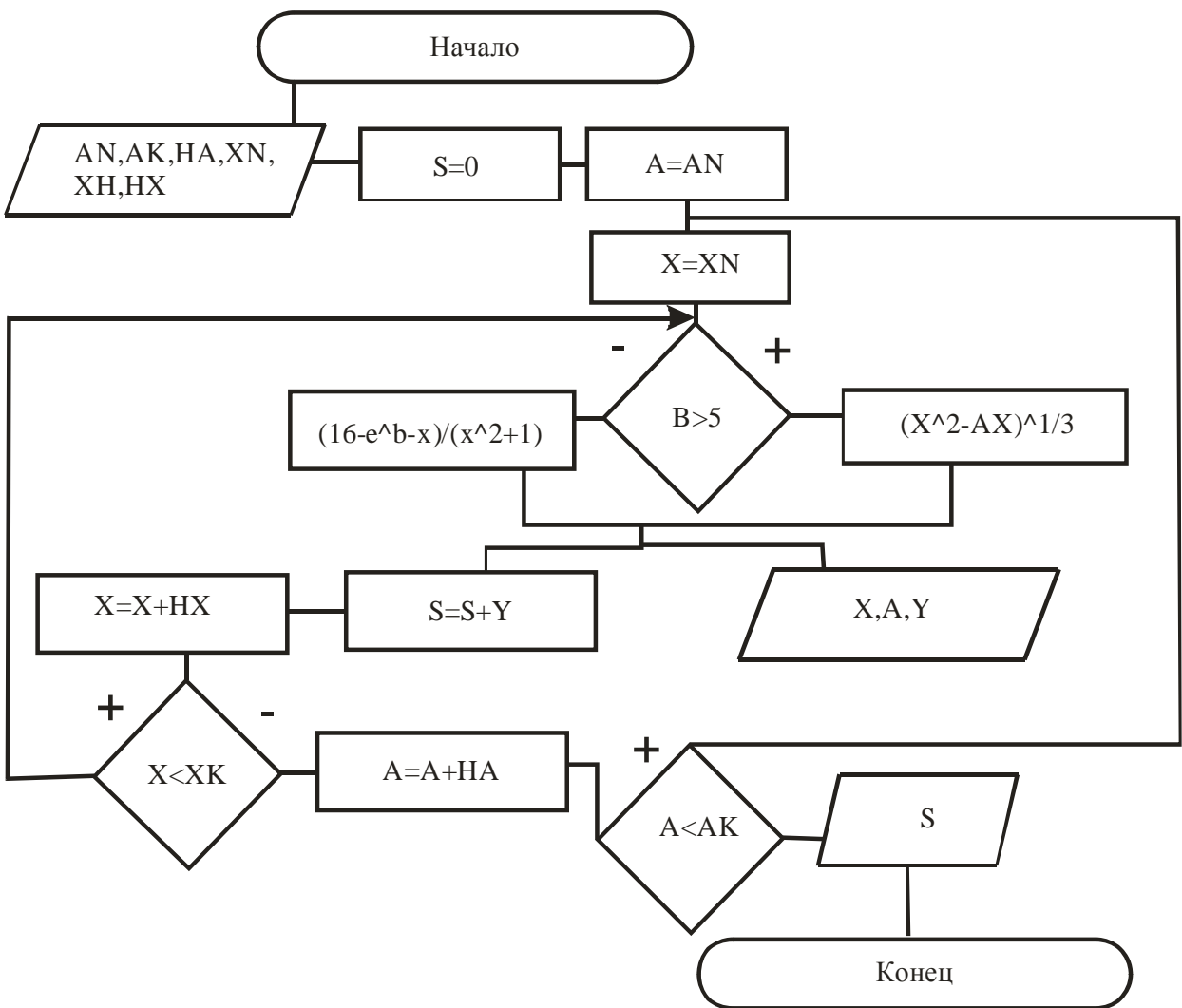
$$-4 \leq a \leq 2$$

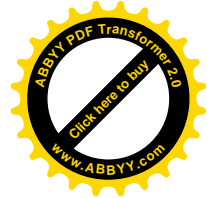
$$h_a = 0.5$$

$$2 \leq x \leq 6$$

$$h_x = 2$$

Найти $S = \sum y$





ЛЕКЦИЯ №4

4. МАССИВЫ. ТИПОВЫЕ АЛГОРИТМЫ

Алгоритмы большинства задач строятся на базе более простых, отработанных решений. Часто используемые алгоритмы называются *типовыми*. Таких алгоритмов не так уж и много, но их знание помогает облегчить разработку более сложных алгоритмов, фрагментами которых могут быть типовые. Это позволяет сэкономить время разработки и сделать алгоритмы оптимальными. Ниже рассматриваются *типовые* алгоритмы.

Большинство задач в вычислительной технике связано с группами чисел (ряд чисел, матрица чисел). Для удобства всю группу чисел принимают как единое целое, но, состоящее из нескольких элементов. Такую группу принято называть *массив*. Массив представляется как единая многоэлементная переменная. Как и любая переменная, массив имеет свое имя. Элементы массива также имеют имя. Оно принимается таким как и имя массива, но к нему добавляется индекс, который является порядковым номером элемента в составе массива. Следовательно, если группу из n чисел представить как массив с именем a , тогда отдельное число этого массива будет обозначаться как a_i . Здесь i – индекс, который может принимать любое значение в интервале от 1 до n , то есть обозначать порядковый номер числа. Например, обозначение a_7 означает, что это элемент массива с именем a , который находится на седьмом месте в его составе. Для хранения значений элементам массива выделяются в оперативной памяти компьютера ячейки, расположенные *поряд*. В большинстве приведенных алгоритмов используются *параметрические циклы*. У них параметр цикла имеет заданную область изменения (начальное и конечное значение). В таких циклах заголовок цикла представлен блоком *модификация*, в плоскости которого указывается имя параметра цикла, его область изменения и шаг.

4.1. Определение суммы чисел произвольного ряда

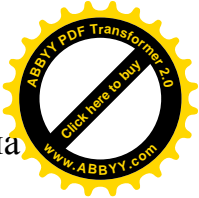
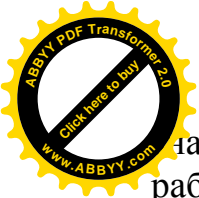
Дан: ряд произвольных чисел $a_1, a_2, a_3, \dots, a_n$.

Определить: сумму этих чисел $S = \sum_{i=1}^n a_i = a_1 + a_2 + a_3 + \dots + a_n$.

Алгоритм решения этой задачи приведен на рис.8.1. Этот алгоритм предусматривает использование циклической операции, в которой параметром цикла является индекс элемента ряда i (его порядковый номер). Тело цикла представлено рекуррентной формулой $S = S + a_i$

Рекуррентной называют такую формулу, с помощью которой последующее значение переменной (сумма S) вычисляется с использованием ее предыдущего значения.

Важной операцией в этом алгоритме является операция $S = 0$, которая предусматривает очистку ячейки оперативной памяти, выделенной для размещения



значений переменной S . Такая операция обязательна и совершается до начала работы цикла.

Дело в том, что при решении какой-либо задачи всем данным, участвующим в операциях вычислительного процесса (исходным, промежуточным, конечным), выделяются ячейки оперативной памяти. После завершения программы данные в используемых ячейках автоматически не удаляются (не обнуляются). Если после этого на компьютере будет решаться другая задача, то для ее переменных могут быть выделены ячейки памяти, которые использовались в предыдущей задаче, с сохранившимися в них данными. В случае, когда решается задача по представленному алгоритму, переменной S может быть выделена ячейка с "остатками" от предыдущей задачи (их еще называют "мусор"), и это может исказить результат решения. Во избежание этого ячейку памяти для переменной S вначале очищают от "мусора", устанавливая значение 0. Можно вместо нуля устанавливать значение первого слагаемого (первого числа ряда), но тогда параметр цикла i должен изменяться не от единицы, а от 2. Чистка ячеек памяти предусматривается только для тех переменных, значение которых вычисляется по рекуррентным формулам. При использовании других формул вычисления такая операция не обязательна.

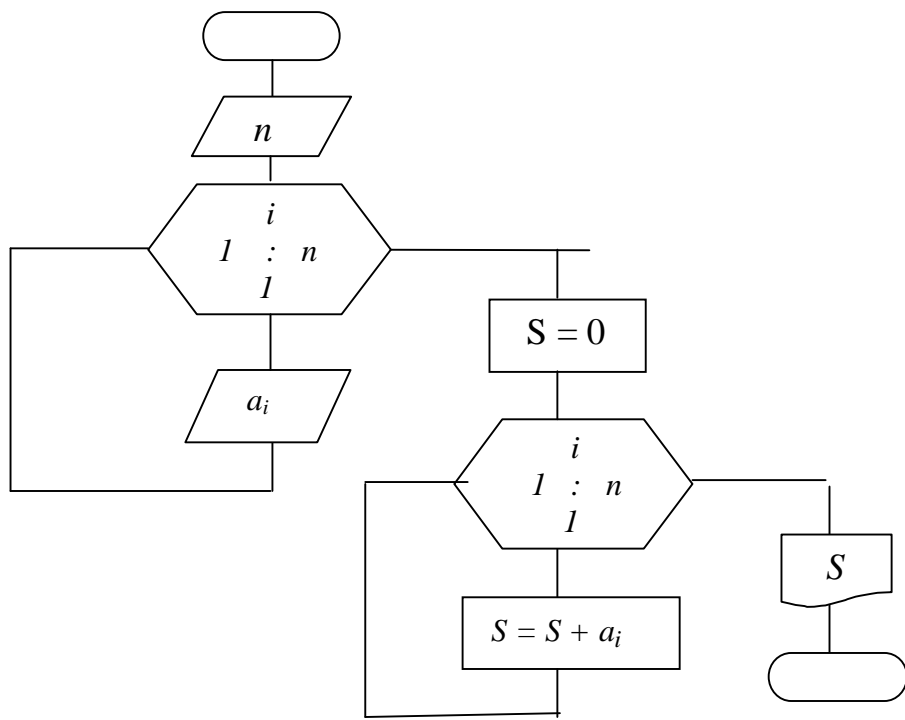


Рис. 4.1 Блок – схема вычисления суммы ряда чисел

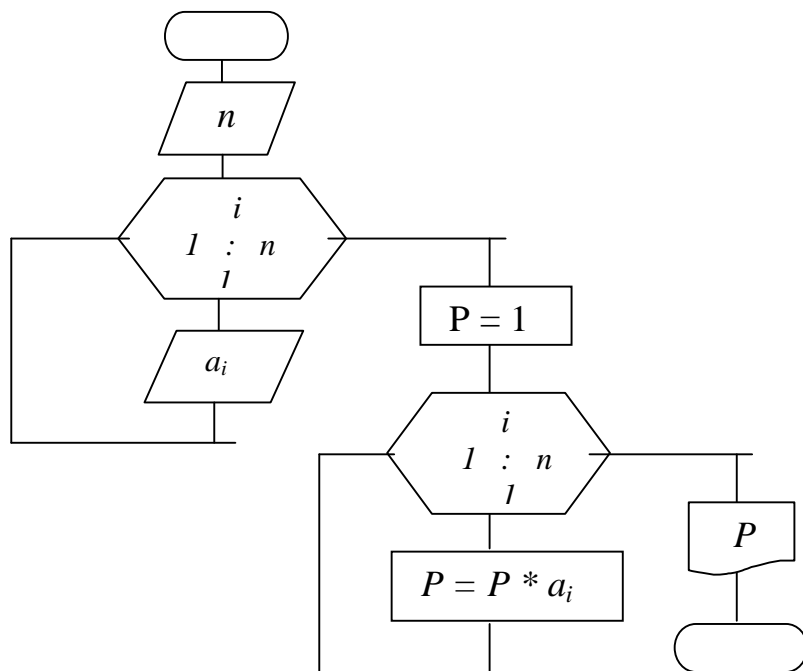


Рис. 4.2 Блок – схема вычисления произведения ряда чисел

4.2. Определение произведения чисел произвольного ряда

Дан: ряд произвольных чисел $a_1, a_2, a_3, \dots, a_n$.

Определить: произведение этих чисел

$$P = \prod_{i=1}^n a_i = a_1 * a_2 * a_3 * \dots * a_n$$

Блок-схема вычислительного процесса приведена на рис.4.2. Этот алгоритм подобен предыдущему. Отличием является только то, что вычисление произведения осуществляется по рекуррентной формуле $P = P * a_i$, а очистка ячейки памяти, выделенной под переменную P , осуществляется не нулем, а единицей по понятным причинам.

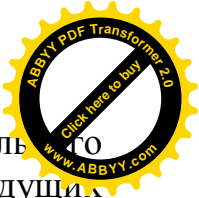
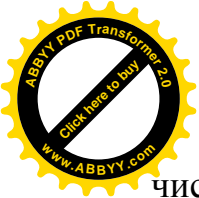
4.3. Алгоритм определения наибольшего (наименьшего) числа из заданного ряда чисел

Дан: ряд произвольных чисел $a_1, a_2, a_3, \dots, a_n$.

Определить: наибольшее число a_{max} и его место в ряде k .

Типовой алгоритм для решения таких задач приведен на рис.4.3.

Он основан на использовании циклической операции, в которой поочередно производится сравнение всех чисел ряда с числом a_{max} , играющим роль эталона для сравнения. В качестве начального значения a_{max} выбирается первый элемент ряда a_1 независимо от его истинного значения (наибольшее, наименьшее среди чисел ряда или какое-либо другое).



Переменная k предназначена для хранения текущего адреса максимального числа, на старте этот адрес – единица. Параметр цикла i , как и в предыдущих алгоритмах, определяет адрес (порядковый номер) числа ряда и имеет начальное значение 2, так, как первое число уже использовано в роли a_{max} . Далее выполняется сравнение значения текущего числа ряда a_i , имеющего адрес i , с числом, выполняющим на данный момент роль максимального a_{max} . Если в результате сравнения текущее число a_i окажется большим максимального a_{max} , то с этого момента роль максимального будет выполнять текущее число a_i , а его адрес будет зафиксирован в качестве значения переменной k . Если же результат сравнения будет иным, то никакие изменения не происходят, а сразу выполняется продвижение цикла, то есть устанавливается адрес i следующего числа ряда и производится переход к новому выполнению тела цикла. В результате перебора всех чисел ряда, определяется наибольшее число a_{max} , его адрес k , и их значения выводятся на экран.

Определение наименьшего числа осуществляется аналогично. Для этого в представленном алгоритме (рис.8.3) идентификатор a_{max} следует заменить a_{min} (поскольку этот идентификатор будет отвечать назначению алгоритма) и операцию сравнения $a_i > a_{max}$ заменить операцией $a_i < a_{min}$.

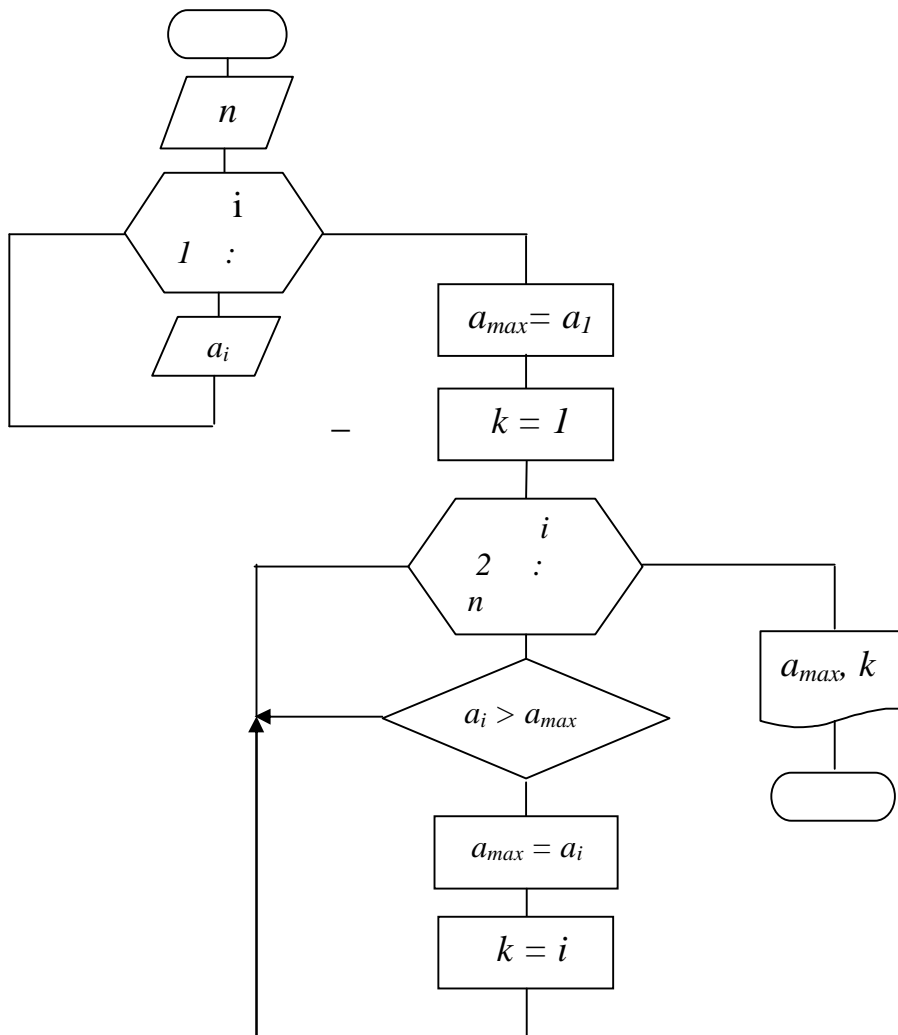


Рис. 4.3 Блок – схема определения наибольшего числа ряда



4.4. Алгоритм определения наибольшего (наименьшего) значения вычисляемой функции

Дана: функция $y = f(x)$, область изменения аргумента x : от x_H (начальное значение), до x_K (конечное значение), с шагом h .

Определить: наибольшее значение вычисляемой функции y_{max} на заданном интервале и значение аргумента x_{max} , при котором функция принимает максимальное значение.

Алгоритм решения этой задачи (рис.4.4) аналогичен алгоритму, приведенному на рис.4.3. Отличие состоит в том, что в предыдущей задаче значения чисел ряда уже были известны, а в этой их необходимо предварительно вычислять. В заголовке цикла, в отличие от предыдущего алгоритма, в качестве параметра цикла используется не адрес числа i , а значение аргумента x .

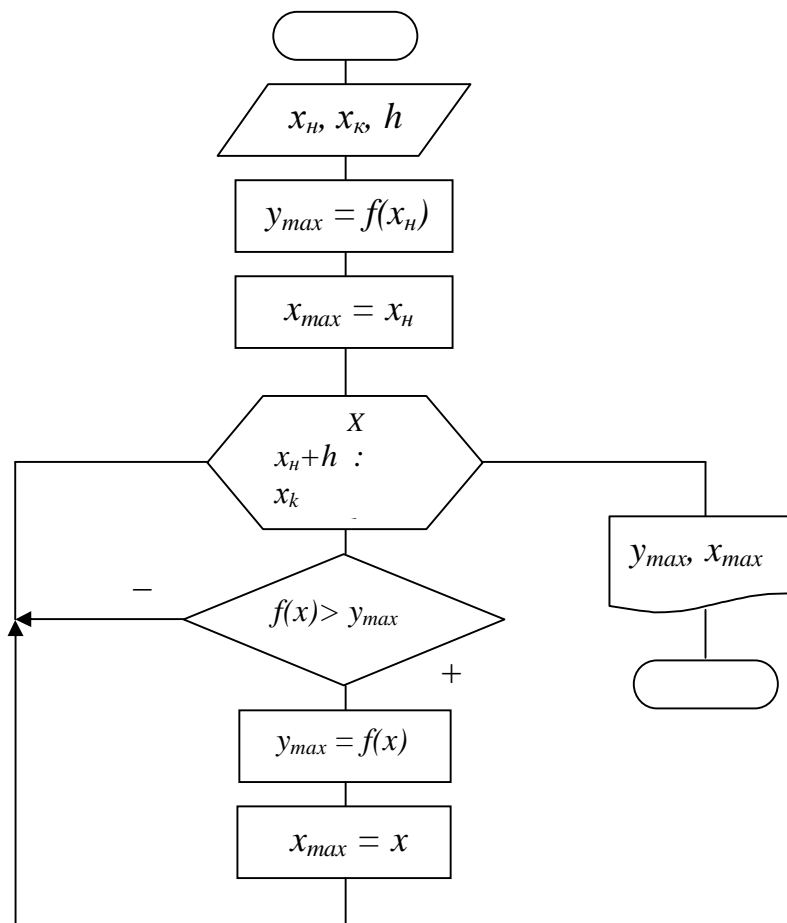
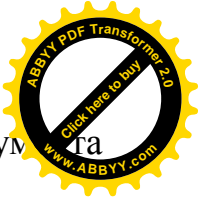
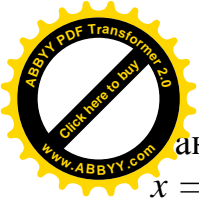


Рис. 4.4. Алгоритм определения наибольшего значения функции

Параметр цикла x изменяется не от начального значения, а от значения $x_H + h$, поскольку значение x_H уже было использовано в операции вычисления начального значения функции $y_{max} = f(x_H)$. Переменная x_{max} хранит текущее значение аргумента, при котором функция принимала максимальное значение. После завершения работы цикла значение переменной y_{max} будет равно



максимальному значению функции на заданном интервале при значении аргумента $x = x_{max}$.

Определение наименьшего значения вычисляемой функции производится аналогично. Для этого, в приведенной на рис.4.4 блок – схеме, идентификаторы y_{max} необходимо заменить y_{min} , x_{max} на x_{min} , а операцию сравнения $y > y_{max}$ – на операцию $y < y_{min}$.

4.5. Вычисление функции с одновременно изменяющимися несколькими аргументами

Дана: функция $F = f(x, y, z)$, аргументы x, y и z изменяются одновременно: x от x_n с шагом h_x , y от y_n до y_k с шагом h_y , z от z_n с шагом h_z .

Вычислить: значения функции.

Из условия задачи видно, что область изменения аргументов дана только для параметра y . Для параметров x и z конечные значения не даны. Это правильно. Если бы для них были бы установлены эти значения, то задачу решить бы было бы невозможно. Такая формулировка была бы неправильной. В самом деле, в каждом из этих параметров могут быть различные начальные значения, различные шаги изменения, различные конечные значения. При этом достичь заданных, в общем – то, различных конечных значений одновременно просто невозможно.

Для решения такого вида задач поступают следующим образом. Используют параметрический цикл и в его заголовок, в качестве параметра цикла, выносят параметр с заданной областью изменения. В приведенном примере – это y . Остальные аргументы (x и z) инициализируют до цикла. В теле цикла предусматривают изменение аргументов, не использованных в качестве параметров цикла (x и z). На рис. 4.5 приведена блок – схема решения этой задачи.

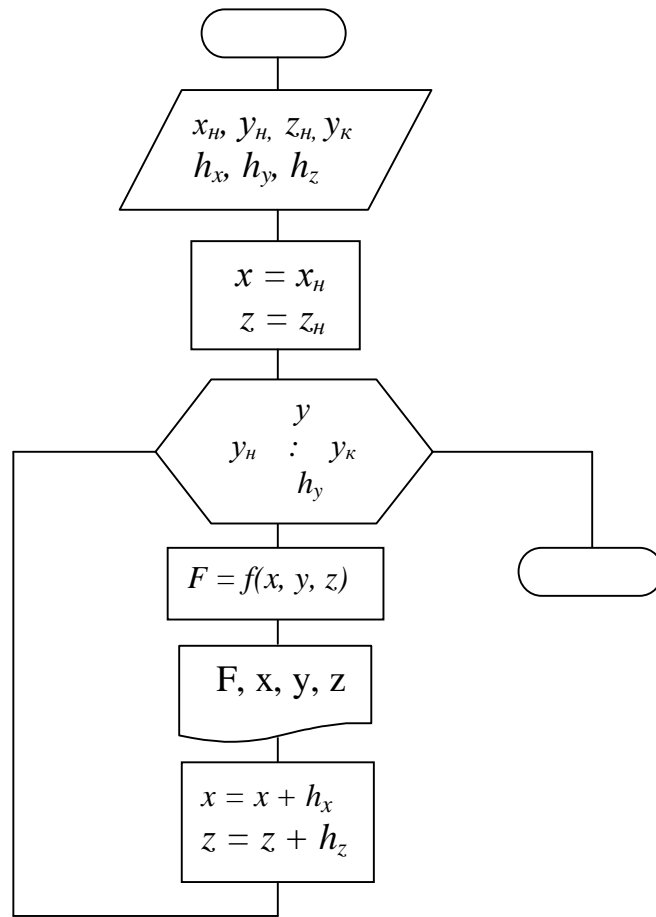


Рис. 4.5. Блок – схема вычисления функции с одновременно изменяющимися несколькими аргументами.

Очевидно, что этим методом можно решать такие задачи с любым количеством аргументов.

4.6. Итерационные циклы

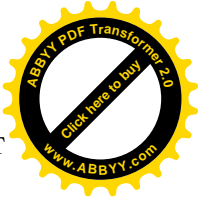
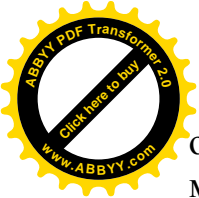
Итерационными называют такие циклы, у которых заранее неизвестно количество шагов. С помощью итерационных циклов решаются задачи, использующие метод *последовательных приближений*.

Примером, использующим итерационный цикл, может быть следующая задача.

Вычислить: сумму $S = 1 + \frac{1}{x} + \frac{1}{x^2} + \frac{1}{x^3} + \dots + \frac{1}{x^i} + \dots$, для любых $x > 1$.

Вычисление производить, пока $\frac{1}{x^i} > \varepsilon$. $\varepsilon = 10^{-5}$

Анализ выражения для вычисления суммы показывает, что знаменатель дроби постоянно возрастает, а значение дроби – уменьшается. Таким образом, каждое



очередное слагаемое будет все меньше и меньше. Если его значение станет меньше числа 10^{-5} , то изменение суммы будет ничтожно мало, и процесс вычисления можно прекратить.

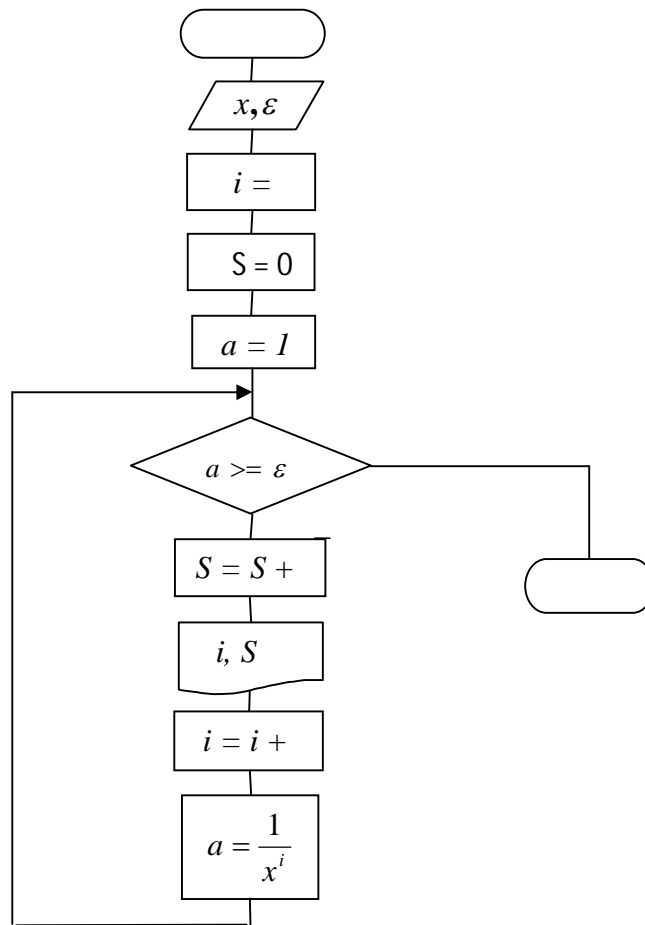
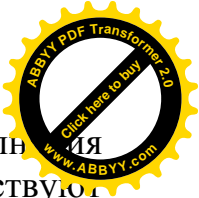
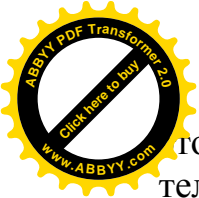


Рис.4.6. Блок – схема итерационного цикла с предусловием

В итерационных циклах основанием для завершения их работы является некое условие. Если оно выполняется, то цикл работает, в противном случае цикл прекращает работу. В представленной задаче цикл работает до тех пор, пока условие $\frac{1}{x^i} > \varepsilon$ выполняется. Итерационные циклы могут быть представлены с **предусловием** или с **постусловием**. Циклы с предусловием предусматривают *вначале проверку условия выполнения цикла*, а затем, если условие выполняется, осуществляется выполнение тела цикла.

В циклах с постусловием эта процедура производится в обратном порядке – *сначала выполняется тело цикла*, а после этого проверяется условие выполнения цикла. В этих циклах возможно *выполнение один раз* тела цикла даже тогда, когда



то не должно происходить. Об этом становится известно уже после выполнения тела цикла. Казалось бы, что циклы с постусловием бессмысленны, но существуют области задач, где их применение более предпочтительно.

На рис. 4.6 и 4.7 приведены блок – схемы вычислений с использованием циклов соответственно с предусловием и постусловием.

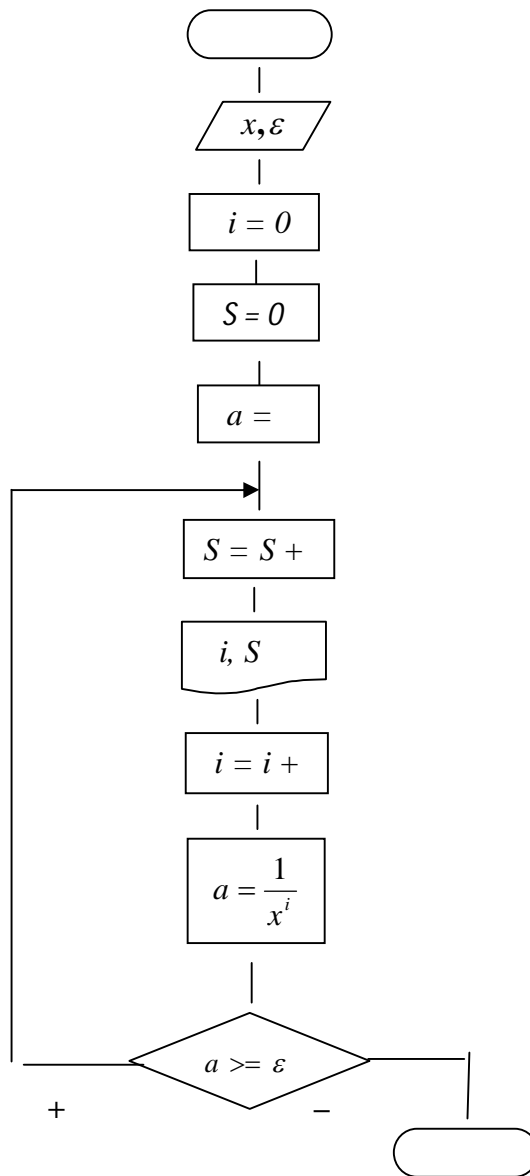


Рис.4.7. Блок – схема итерационного цикла с постусловием



ЛЕКЦИЯ №5

5. ТИПОВЫЕ АЛГОРИТМЫ

5.1. Сортировка ряда чисел

Сортировка – это упорядочение (размещение) чисел по возрастанию или по убыванию их значений.

Существует несколько различных способов решения этой задачи, однако чаще всего используются:

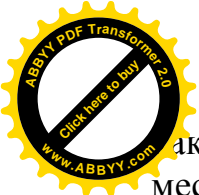
- *метод смежных пар* (метод "пузырька"),
- *метод поиска наименьшего (наибольшего) числа ряда.*

5.1.1. Метод смежных пар

Алгоритм, реализующий метод смежных пар, основан на сравнении двух соседних чисел с последующей их перестановкой в случае, если они расположены не в требуемой очередности.

На рис.5.1 приведен алгоритм сортировки заданного ряда чисел $a_1, a_2, a_3, \dots, a_n$ по возрастанию. Для регулярного сравнения двух соседних чисел предусматривается циклический процесс. В заголовке цикла, в качестве параметра цикла, используется адрес числа i с его областью изменения от 1 до $k = n - 1$. Здесь следует обратить внимание на то, что правая граница области изменения параметра цикла i равна не n , а $n - 1$. Это объясняется тем, что при сравнении двух соседних чисел адрес первого числа определяется значением параметра цикла i , а адрес второго числа вычисляется добавлением единицы ($i + 1$). Если бы параметр цикла имел бы свое последнее значение n , а не $n - 1$, то при $i = n$ адрес второго числа определялся бы как $i = n + 1$, а это уже выходит за пределы рассматриваемого ряда, и числа с таким адресом нет. Поэтому, чтобы исключить такую ситуацию, последнее значение параметра цикла принимается на единицу меньше.

В теле цикла происходит сравнение значений двух соседних чисел a_i и a_{i+1} , и в случае их неправильного расположения осуществляется их взаимная перестановка с помощью промежуточной переменной R . Переменная R используется с целью исключения потери значения одного из чисел при перестановке. Факт перестановки фиксируется с помощью переменной F , которая играет роль индикатора, путем установки ей значения равной единице. Если же числа расположены в требуемой очередности, то их перестановка не производится. После завершения каждого цикла, на последнее место перемещается самое большое число из чисел, рассматриваемых в этом цикле. Кроме этого, проверяется состояние индикатора F . Если его значение равно единице, то это означает, что хотя бы одна перестановка в завершившемся цикле была, и, следовательно, необходимо совершить еще один цикл для сравнения соседних чисел, но без последнего числа, которое и так уже расположено на нужном месте. Исключение последнего числа из рассмотрения, после завершения очередного цикла, осуществляется операцией $k = k - 1$. Это позволяет экономить машинное время, так



Так нет никакого смысла проверять размещение уже расположенного на нулевом месте числа.

После этого все начинается сначала. Индикатор F устанавливается в стартовое состояние со значением нуль, и в цикле производится поочередное сравнение двух соседних чисел. Такие циклы будут повторяться до тех пор, пока не обнаружится, что после завершения очередного цикла индикатор F не поменял своего начального значения 0 на 1 . И это будет означать, что в процессе циклической операции не было ни одной перестановки, а, следовательно, числа расположились в требуемой последовательности, то есть ряд отсортирован. Таким образом, алгоритм позволяет повторять циклические операции столько раз, сколько будет необходимо для завершения процесса сортировки.

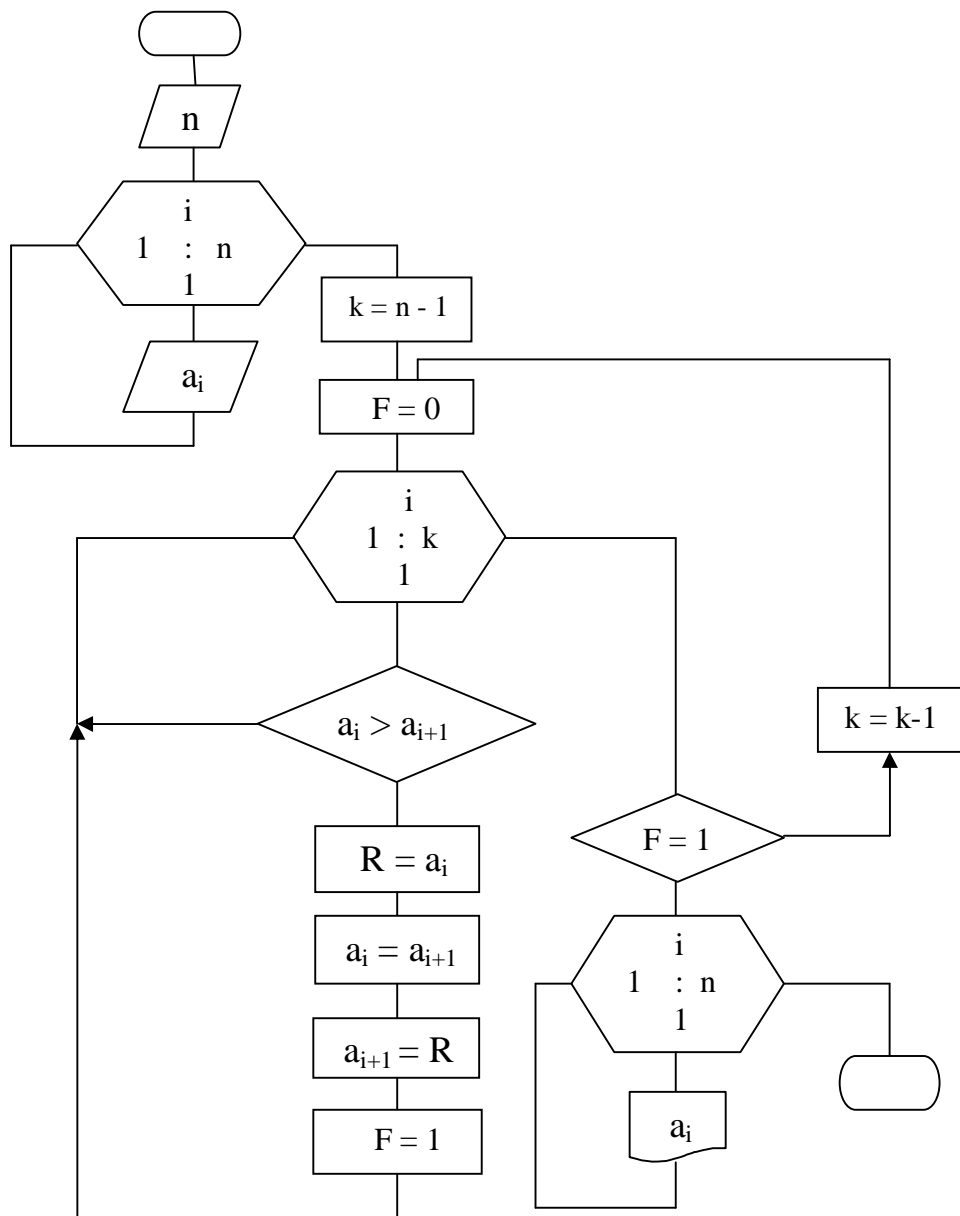
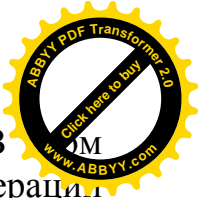
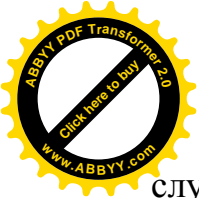


Рис. 5.1 Блок – схема сортировки чисел по возрастанию



Сортировка ряда чисел по убыванию выполняется аналогично. В случае в блок-схеме следует выполнить некоторые изменения. Так в операции сравнения соседних чисел необходимо знак " $>$ " заменить знаком " $<$ ".

Метод смежных пар экономичен с точки зрения расхода машинного времени. Если исходный ряд уже сортирован в требуемой последовательности, то достаточно выполнить только один проверочный цикл, чтобы убедиться, что индикатор F не поменял своего начального значения 0 .

5.1.2. Метод поиска наименьшего (наибольшего)

Идея метода состоит в том, что определяется место наименьшего числа при сортировке ряда по возрастанию (наибольшего числа при сортировке по убыванию), которое затем меняется местами с первым числом ряда. После этого ряд чисел рассматривается уже без первого числа. Найденное наименьшее (наибольшее) число выставляется аналогичным образом на второе место и в дальнейшем ряд чисел рассматривается уже без двух первых чисел, которые выставлены в нужном порядке. Это повторяется многократно до тех пор, пока останется одно число, после чего процесс сортировки прекращается.

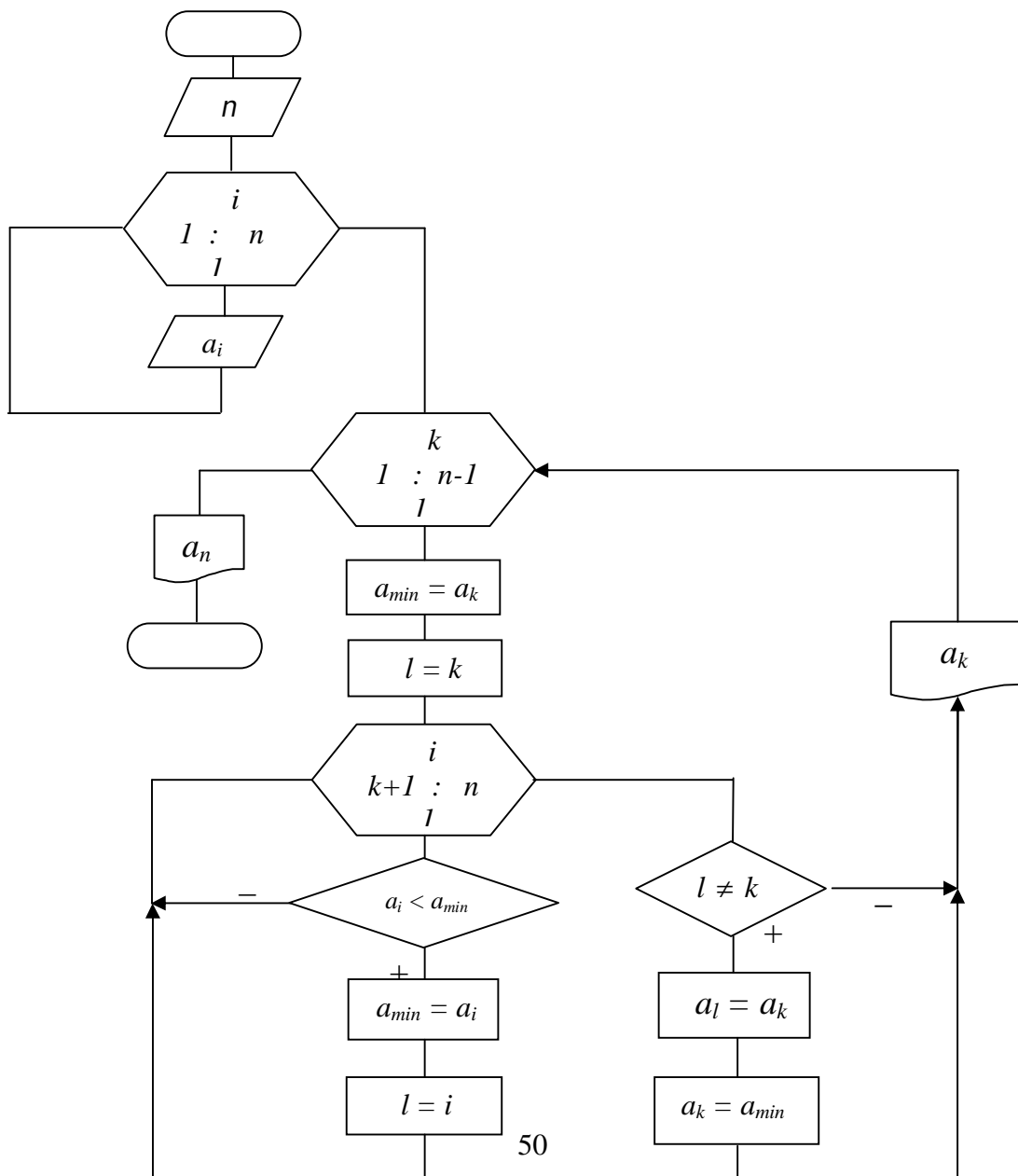




Рис. 5.2 Блок – схема сортировки чисел по возрастанию методом определения наименьшего числа

На рис.5.2 приведен алгоритм сортировки n чисел по возрастанию. Блок – схема представлена двумя вложенными циклами. Внешний цикл с параметром цикла k осуществляет назначение каждый раз числа на роль минимального a_{min} в рассматриваемом ряде чисел и одновременно отсекает уже выставленное вперед число после предыдущего цикла.

Вложенный цикл с параметром цикла i представляет собой типовой алгоритм определения наименьшего числа из ряда чисел. Здесь роль наименьшего числа a_{min} играет число a_k , которое изменяется последовательно от a_1 до a_{n-1} . Вложенный цикл рассматривает ряд чисел, начиная с соседнего с a_k , то есть с числа a_{k+1} . За пределами вложенного цикла производится обмен числами, наименьшего и первого в рассматриваемом ряде. Здесь роль переменной R , используемой в методе смежных пар, играет переменная a_{min} .

Если в роли минимального числа ряда осталось первое число a_k , то такой обмен отпадает. Этот факт устанавливается проверкой условия $l \neq k$.

В отличие от предыдущего метода, этот алгоритм осуществляет сортировку чисел по «жесткой» программе. То есть, независимо от того, сортирован ли исходный ряд или нет, тело внешнего цикла будет выполнено $n - 1$ раз. В этом случае непроизводительно расходуется машинное время, что определяет недостаток метода. Но, тем не менее, этот метод имеет свою область применения. Таким методом удобно решать задачи по определению нескольких (больше одного) минимальных чисел ряда. В этом случае достаточно установить правую границу параметра внешнего цикла k не $n-1$, а то число, которое определяет количество искомых минимальных элементов ряда.

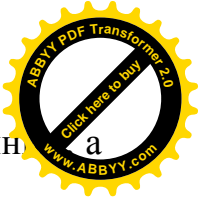
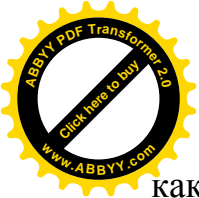
Сортировка по убыванию осуществляется аналогично. При этом в рассмотренной блок-схеме идентификатор a_{min} необходимо заменить идентификатором a_{max} , что вызывает правильные ассоциации, а операцию сравнения " $<$ " – на операцию " $>$ ".

5.2. Типовые алгоритмы решения задач с использованием матриц

Отличительной особенностью алгоритмов решения задач с использованием матриц (двухмерных массивов) от алгоритмов, ориентированных на работу с рядами чисел (векторами), является наличие двух вложенных циклов.

На рис.5.3 приведен алгоритм вычисления суммы элементов матрицы $a(m*n)$. Внешний цикл предназначен для установления текущего адреса i строки матрицы, а внутренний (вложенный) – текущего адреса j ее столбца.

Вычисление произведения P элементов матрицы осуществляется по аналогичной блок – схеме, в которой следует заменить символ S на P , операцию $S = 0$ на операцию $P = 1$, а операцию $S = S + a_{ij}$ - на операцию $P = P * a_{ij}$.



В некоторых случаях возникает вопрос, какой цикл должен быть внешним, какой – внутренним. Ответ на него следующий.

Если условием задачи предусматриваются какие-либо операции в строках, то внешний цикл должен определять текущий адрес строки, если же предусматриваются операции в столбцах, то внешний цикл должен определять текущий адрес столбца. Если же операции над элементами матрицы не акцентируются на строках или на столбцах, как в случае вычисления суммы или произведения элементов матрицы, то взаимное расположение циклов может быть любым.

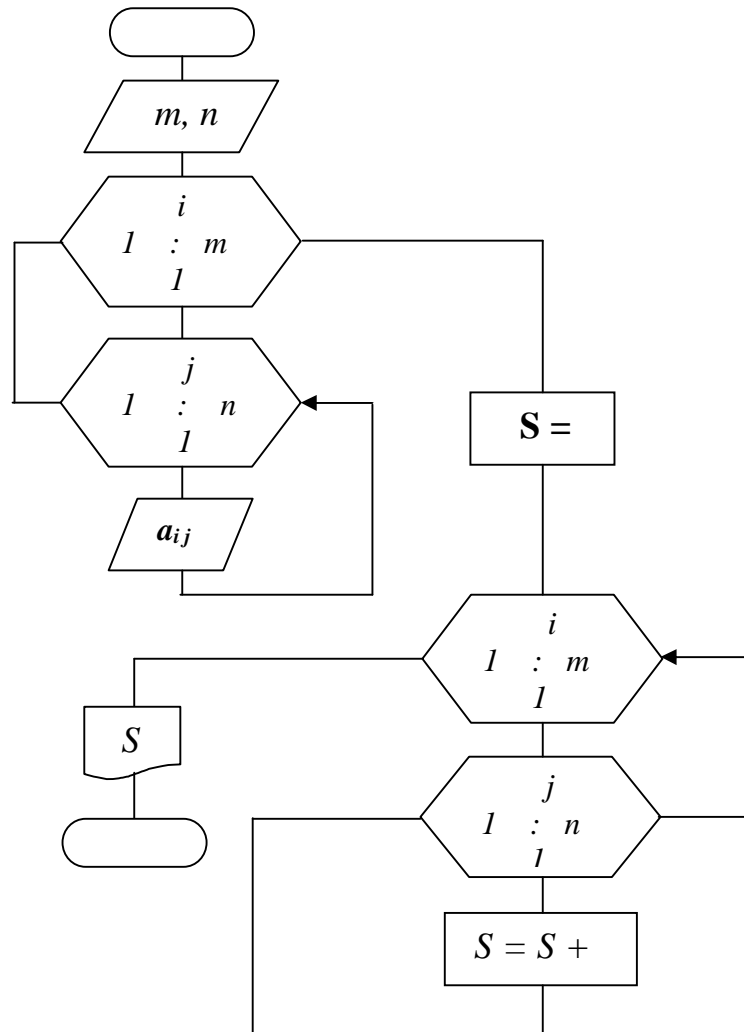
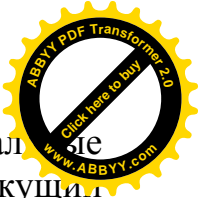
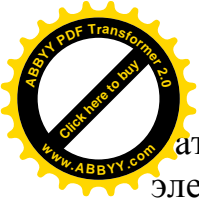


Рис. 5.3 Блок – схема вычисления суммы элементов матрицы

Пример 1. В заданной матрице $a(m*n)$ определить наибольшие элементы строк.

Алгоритм решения этой задачи представлен на рис.9.4. Здесь внешний цикл определяет текущий адрес строки. Получив адрес строки i , первый элемент ее a_{i1} (элемент первого столбца) берется в качестве начального значения переменной a_{max} . Вложенный цикл предусматривает определение текущего порядкового номера элемента в данной строке (номер столбца). В дальнейшем используется типовой алгоритм определения наибольшего числа в ряде чисел (в строке



матрицы). Следует подчеркнуть, что в задаче требуется определить максимальные элементы *в строках*, поэтому внешний цикл алгоритма устанавливает текущий адрес строки.

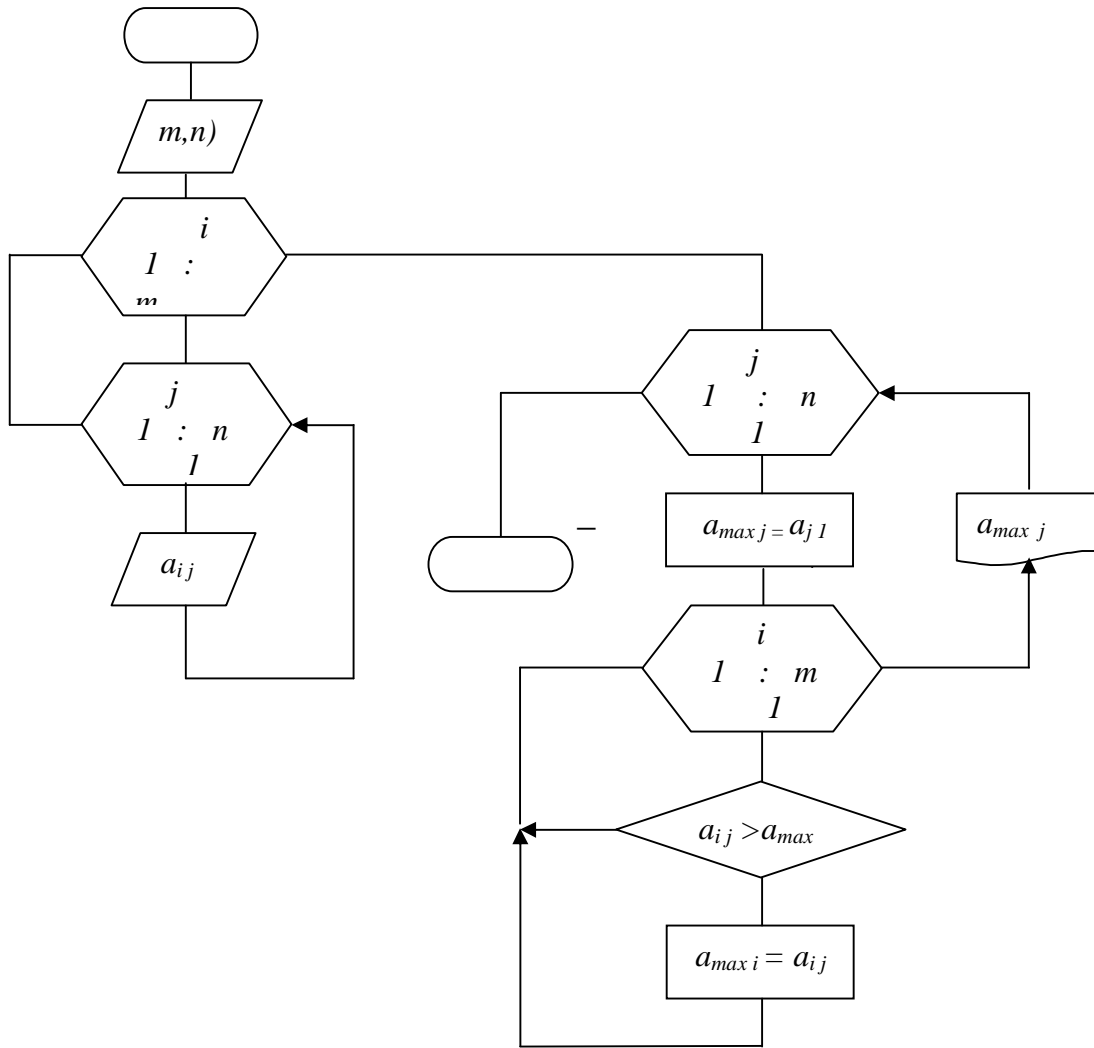


Рис.5.4. Блок-схема определения максимального элемента в строке матрицы

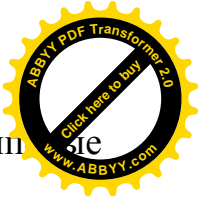
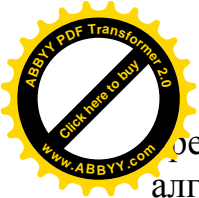
Пример 2. В заданной матрице $a(m*n)$ определить наибольший элемент в каждом столбце.

Алгоритм решения этой задачи приведен на рис.5.5. В отличие от предыдущего примера в этом алгоритме внешний цикл определяет текущий адрес столбца.

В случае примера с вычислением суммы всех элементов матрицы (Рис.5.3) внешний цикл может определять как текущий адрес строки, так и текущий адрес столбца, поскольку здесь не акцентируется место операций (строки или столбцы). Областью операций являются все элементы матрицы без уточнения, в каких строках или столбцах они расположены.

Приведенные алгоритмы составляют далеко не полный перечень алгоритмов с операциями над элементами матриц. Однако они могут быть использованы как фрагменты при разработке более сложных алгоритмов.

Следует хорошо понимать логику типовых алгоритмов, хорошо запомнить их и умело применять при решении различных задач. Это позволит экономить



время на разработку алгоритмов других задач, в которых используются типичные алгоритмы.

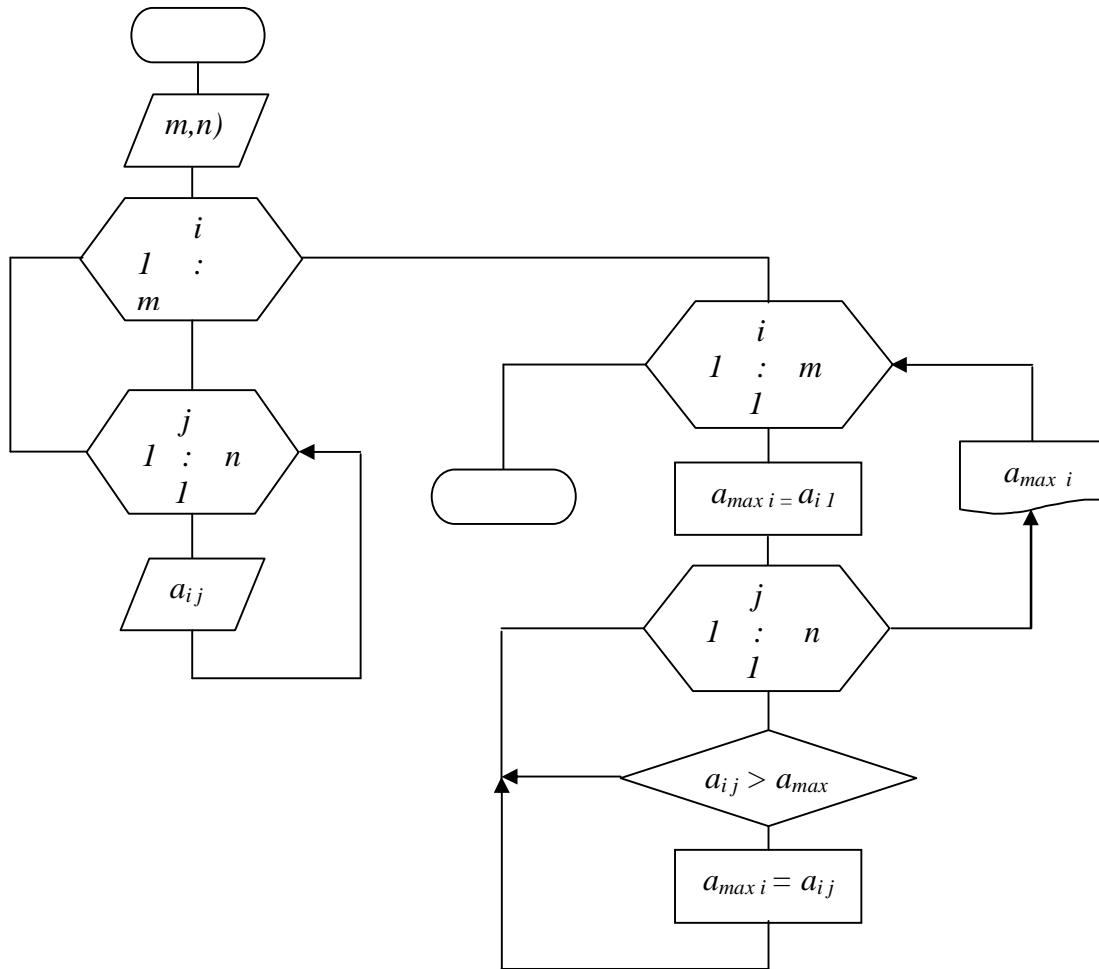


Рис. 5.5 Блок-схема определения максимального элемента в столбце матрицы



6. ОСНОВЫ ПРОГРАММИРОВАНИЯ В VBA

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

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

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

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

ОБЪЕКТ. СВОЙСТВО = ЗначениеСвойства.

Например,

.....

‘изменить имя текущего листа

ActiveSheet . Name = “Январь”

.....

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

ОБЪЕКТ.МЕТОД

Например,



.....
'удалить лист
WorkSheets("Обзор") . Delete
.....

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

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

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

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

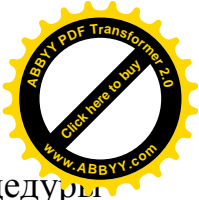
Код VBA

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

Каждое приложение MS Office хранит проекты по своему:

- Word сохраняет проекты в шаблонах документа (DOT) и в документах (DOC).
- Excel сохраняет проекты в рабочих книгах (XLS).
- Access сохраняет проекты в файлах базы данных (MDB).
- PowerPoint сохраняет проекты в презентациях (PPT).

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



Sub имя()

Это стандартный заголовок макроса. Завершающей строкой процедуры является строка

End Sub

Для запуска программы из редактора - клавиша F5. Для запуска программы из документа — alt+F8.

6.1. Элементы языка VBA

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

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

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

Примеры правильной записи имен:

H123, Price, Sale

Неправильно записанные имена:

21days начинается с цифры;

Price Sale содержит символ пробела;

Dim совпадает со служебным словом языка.

Дадим также несколько определений элементам VBA.

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

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

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



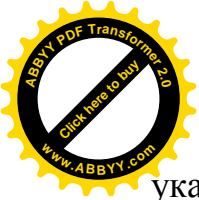
6.2. Типы данных языка VBA

Visual Basic имеет 11 встроенных типов данных:

| Тип данных | Размер в байтах | Разрядность цифр | | Диапазон значений |
|------------|---|------------------|---|---|
| Boolean | 2 | 1 | логический | True или False (да - нет, включен - отключен) |
| Integer | 2 | 5 | целое | От (-32768) до 32768 |
| Long | 4 | 10 | длинное целое | От (-2147483648) до 2147483647 |
| Single | 4 | 7 | действительное одинарной точности | От (-3.402823E+38) до (-1.401298E-45) и от 1.401298E-45 до 3.402823E+38 |
| Double | 8 | 15 | действительное двойной точности | От ± 1.79769313486232E+308 до ± 4.94065645841247E-324 |
| Currency | 8 | 19 | денежный | От (-922337203685477.5808) до 922337203685477.5807 |
| Date | 8 | - | для дат и времени | От 01.01.100 до 31.12.9999 |
| String | 1 | +1 | символьный | на любой символ - От 0 до 65535 символов |
| Object | 4 | - | | Любой объект |
| Array | Определяется количеством и размером элементов | - | массивы | - |
| Variant | Определяется записанными данными | - | | Любой из встроенных типов данных |

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

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



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

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

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

Currency предназначен для хранения действительного числа с фиксированной точкой, которая всегда располагается между 4 и 5 цифрами справа.

ПРИМЕР: 0.0021 46.7280 Этот тип специально предназначен для получения минимальных ошибок округления и чаще всего используется в бухгалтерских расчетах.

Date предназначен для хранения дат и времени с точностью до секунды.

Машиной даты записываются следующим образом: целая часть – количество дней, прошедших от начала отсчета 01.01.1900, а дробная часть – время.

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

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

Object предназначен для ссылок на любой объект VBA

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

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

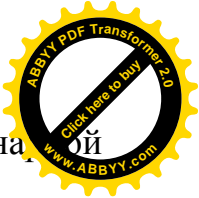
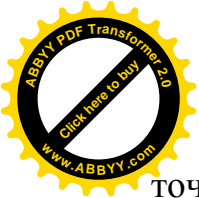
6.3. Переменные

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

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

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

Например:



Dim x As Single ‘переменная x описана как действительная одина- точности

Dim I As Integer ‘переменная I описан как целая

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

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

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

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

6.4. Константы

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

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

Примеры описания констант:

Const Pi as Double = 3.1415926 ‘описана константа Pi

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

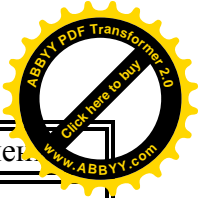
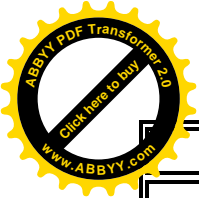
6.5.Операции

Операции производятся над выражениями, переменными, константами, литералами.

Арифметические операции

В качестве арифметических операций в языке Visual Basic применяются операции, описанные в таблице в порядке убывания приоритета.

| Математические и строковые операции VBA в порядке убывания приоритета Операция | Выполняемое действие |
|---|----------------------|
| ^ | Возведение в степень |
| - | Отрицание |
| * | Умножение |
| / | Деление 1 |



| | |
|-----|----------------------------------|
| \ | Целочисленное деление |
| Mod | Определение остатка от деления |
| + | Сложение |
| - | Вычитание |
| & | Объединение (конкатенация) строк |

Правила записи арифметических выражений в языке Бейсик.

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

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

с) Знаки арифметических операций опускать нельзя.

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

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

Логические операции

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

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

Операции отношения:

- = (равно)
- <> (не равно)
- < (меньше)
- > (больше)
- <= (меньше или равно)
- >= (больше или равно)

Например: $a > b$ и $b <= a + 2$.

Логические выражения могут быть связаны знаками *логических операций*:

- OR - Логическое сложение (ИЛИ),
- AND - Логическое умножение (И),
- NOT - Логическое отрицание (НЕ),
- XOR - Исключающее ИЛИ,
- EQV - Логическая эквивалентность,
- IMP - Логическая импликация .



Операция конкатенации

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

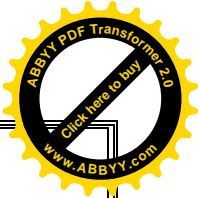
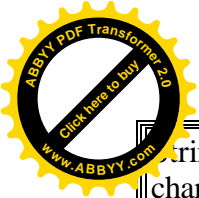
Например,
 операнд1="язык"
 операнд2="Visual Basic"
 результатом операции
 "язык" + "Visual Basic"
 будет строка
 "язык Visual Basic".

6.6. Математические функции VBA

| Функция | Выполняемое действие |
|---------------|--|
| Atn(x) | Возвращение арктангенса угла в радианах $\arctg x$ |
| Sin(x) | Возвращение синуса угла в радианах $\sin x$ |
| Cos(x) | Возвращение косинуса угла в радианах $\cos x$ |
| Tan(x) | Возвращение тангенса угла в радианах $tg x$ |
| Exp(x) | Возвращение значения e^x , где x - значение переменной или введенное число |
| Log(x) | Возвращение натурального логарифма числа $\ln x$ |
| Sqr(x) | Возвращение квадратного корень числа |
| Abs(x) | Возвращение абсолютного значения числа (по модулю) $ x $ |
| Fix(x) | Отбрасывает дробную часть числа и возвращает целую. В результате для положительных чисел получается число меньше, чем входное (Fix(2.5) возвратит 2), для отрицательных - большее (Fix(-2.5) возвратит -2) |
| Int(x) | Отбрасывает дробную часть числа и возвращает целую. Для положительных получается число меньше введенного (Int(2.5) возвратит 2), для отрицательных - так же меньше (Int(-2.5) возвратит -3). |
| Sgn() | Функция предназначена для определения знака числа. Если число положительное - она возвращает 1. Для нуля функция возвратит 0, для отрицательного числа -1. |

6.7. Строковые функции

| Функция | Описание |
|----------------------|--|
| Len(string) | Возвращает длину строки. Например, длина строки "Добрый день" составляет 11 символов - учитывая пробел. Выходное значение имеет тип Long |
| LCase(string) | Возвращает строку, все символы которой записаны в нижнем регистре. Например, строка "Привет" превратится в "привет" |
| UCase(string) | Возвращает строку, все символы которой записаны в верхнем регистре. Например, для "Привет" мы получим "ПРИВЕТ" |



| | |
|------------------------------|--|
| String(number, character) | Возвращает строку, состоящую из number символов character |
| Left(string, length) | Возвращает length символов, начиная с первого левого символа строки string |
| Right(string, length) | Возвращает length символов, начиная с самого правого символа строки string |
| LTrim(string) | Возвращает строку, в которой вырезаны все пробелы слева |
| RTrim(string) | Вырезает из строки все пробелы справа |
| Trim(string) | Вырезает из строки все пробелы слева и справа |
| Mid(string, start[, length]) | Вырезает из строки string с позиции start length символов |
| Asc(string) | Возвращает ASCII-код первого символа строки |
| Chr(charcode) | Возвращает символ, соответствующий коду символа |

6.8. Функции преобразования типов Str() и Val()

Str(число)

Возвращает значение типа Variant (String), являющееся строковым представлением числа. Обязательный аргумент *число* может задаваться любым допустимым числовым выражением.

При преобразовании числа в строку в начале строки обязательно резервируется позиция для знака числа. Если число является положительным, возвращенная строка будет содержать пробел на месте знака.

Примечание.

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

Val(строка)

Возвращает числа, содержащиеся в строке, как числовое значение соответствующего типа.

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

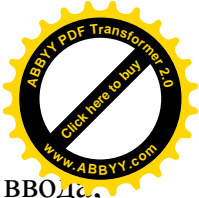
В следующем примере возвращается числовое значение 1615198:

```
Val(" 1615 198-я ул. Н.Й.")
```

6.9. Организация ввода/вывода информации

Диалоговые функции ввода/вывода информации

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



Функция *InputBox*

Выводит на экран диалоговое окно, содержащее сообщение и поле ввода, устанавливает режим ожидания ввода текста пользователем или нажатия кнопки, а затем возвращает значение типа String, содержащее текст, введенный в поле.

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

сообщение — текст в диалоговом окне.

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

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

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

Пример. Ввести значение X=0.5.

```
.....  
X=Val(InputBox("Введите X"))  
.....
```

В диалоговом окне ввести с клавиатуры 0.5.

Функция *MsgBox*

Функция *MsgBox* позволяет выводить результаты счета программы в диалоговом окне.

Синтаксис :

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

сообщение — строковое выражение, отображаемое в диалоговом окне.

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

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

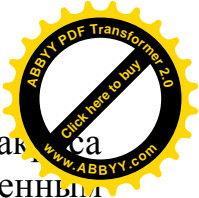
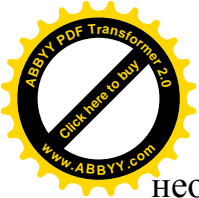
Будем использовать в качестве аргумента функции только *сообщение*, представляющее собой конкатенацию (сцепление)(+) двух строк: комментария к результату, взятого в кавычки, и строки, полученной из числового выражения с помощью функции Str.

Пример. Вывести результат на экран монитора.

```
.....  
MsgBox("Ответ=" + Str(y))  
.....
```

Использование окна отладки

Окно отладки можно вывести на экран при помощи команды Вид - Окно отладки или нажав клавиши <Ctrl>+<G> .



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

Debug.Print выражение.

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

.....
Debug.Print y
.....

Использование объектов Excel

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

Sheets (“Лист1”) . Select

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

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

X=Cells(2,1)

Пример.

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

.....
Sheets(“Лист3”).Select ‘переходим на третий лист
Cells(2,8)=s ‘выводим s
Cells(3,8)=p ‘выводим p
.....

Часто используется свойство *Value*, возвращающее содержимое ячейки. Например,

Cells(2,8).Value=s ‘выводим s



ЛЕКЦИЯ №7

7. ОПЕРАТОРЫ ЯЗЫКА VBA И РЕАЛИЗАЦИЯ ОСНОВНЫХ АЛГОРИТМИЧЕСКИХ КОНСТРУКЦИЙ

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

Оператор комментариев не выполняет в программе никаких действий и может содержать любой текст. Инструкция комментариев имеет два вида: это инструкция, начинающаяся с ключевого слова REM, за которым следует произвольный текст, или знак апострофа ('), который ставится в начале строки или после любого оператора, и за ним также можно писать любой текст.

Например:

```
REM произвольный текст комментария  
' произвольный текст комментария  
A1=5 ' комментарий к строке оператора
```

Оператор присваивания

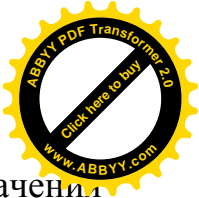
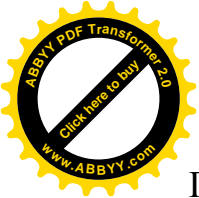
Присваивает выражение переменной или константе. Инструкции присвоения всегда включают знак равенства (=).

ПЕРЕМЕННАЯ = ВЫРАЖЕНИЕ

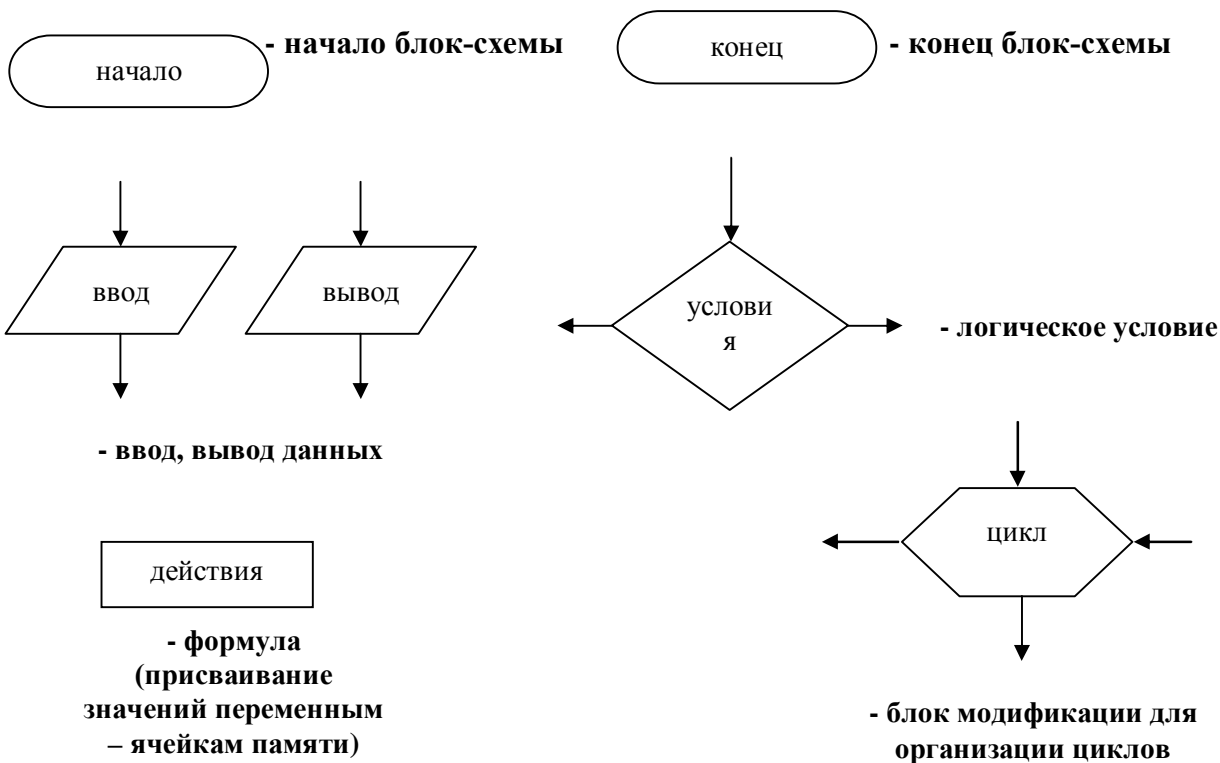
Вначале вычисляется выражение, а затем результат запоминается в переменной.

Выделяют следующие алгоритмические структуры:

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



ГОСТом установлены следующие геометрические фигуры для обозначения действий:



7.1. Программирование алгоритмов линейной структуры

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

При составлении линейных программ необходимо:

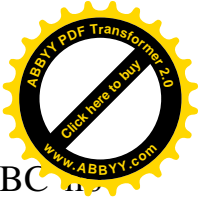
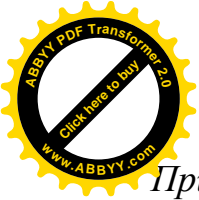
- выделить начальные данные;
- установить последовательность вычислений;
- определить исходные данные до момента их использования;
- желательно снабжать программу комментариями для удобства ее чтения.

Пример. Вычислить значение выражения для любых *a* и *b*.

Программа

```

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

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

Программа

```
Sub Geron()  
Dim a, b, c, p, s As Double  
a = Val(InputBox("Введите a")) 'ввод значения переменной a  
b = Val(InputBox("Введите b")) 'ввод значения переменной b  
c = Val(InputBox("Введите c")) 'ввод значения переменной c  
'P - полупериметр, S - площадь  
p = (a + b + c)  
s = Sqr(p * (p - a) * (p - b) * (p - c))  
Cells(1, 1) = "Площадь="  
Cells(1, 2) = s  
End Sub
```

7.2. Операторы управления порядком выполнения команд

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

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

Оператор безусловного перехода GoTo

Позволяет изменить порядок выполнения команд. Задаёт безусловный переход на указанную строку внутри той процедуры, в которой она находится.

Синтаксис инструкции GoTo:

Гото строка

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

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

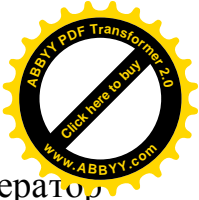
Примечание.

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

Оператор If... Then... Else

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

Синтаксис, т.е. правила записи строчной инструкции If...then...else следующий:



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

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

Обязательный параметр *условие* — представляет собой логическое выражение, которое возвращает значение (истина) TRUE или (ложь) FALSE. Если *условие* равно TRUE, то выполняется *оператор1*, если *условие* равно FALSE, то выполняется *оператор2*. Далее выполняется, если иное не предусмотрено программой в *операторах1* или 2, инструкция, следующая по порядку за строкой с инструкцией If... Then...Else.

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

If *условие* Then [*оператор*]

Примечание.

В квадратных скобках обозначены те элементы инструкции, которые не являются обязательными для записи.

В реальных случаях часто требуется обработка нескольких инструкций при выполнении условия. В этих случаях предпочтительнее блочный вид инструкции If, синтаксис которой выглядит следующим образом:

```
If условие Then
[Блок операторов]
[Else
[Блок операторов]]
End If
```

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

Оператор Select Case

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

```
Синтаксис инструкции
Select Case выражение
[Case список условий-1
[операторы-1]]
[Case список условий-2
[операторы-2]]
...
[Case список условий-n
```



```
[операторы-n]]  
[Case Else  
  [операторы_ else]]  
End Select
```

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

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

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

Выполняется оператор Select Case следующим образом — сначала вычисляется значение выражения, стоящего после ключевых слов Select Case, затем производится проверка на совпадение значения выражения и одного из Case условий, в случае совпадения выполняются операторы, помещенные за этим Case условием. В случае, если совпадения не обнаружено ни в одном из Case условий, то выполняются операторы, находящиеся после ключевых слов Case Else. Если список выражений в Case условии записан в первой форме, то необходимо, чтобы результат вычислений выражения равнялся одной константе из списка, если Case условие записано во второй форме, то необходимо, чтобы выполнялось условие отношения между значением выражения и константой. Если же в третьей форме, то необходимо выполнение соотношения $\text{КОНСТАНТА1} \leq \text{выражение} \leq \text{КОНСТАНТА2}$.

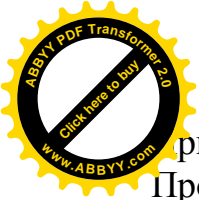
7.3. Программирование алгоритмов разветвляющейся структуры

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

Пример. Дано действительное число x . Вычислить y .

Программа.

```
Sub Raz2()  
Dim x, y As Double  
x = Val(InputBox("Введите x")) 'ввод значения переменной x  
If x > 0 Then y = Sin(x) Else y = 2 * x  
MsgBox ("Значение y=" + Str(y#))  
End Sub
```



Пример. Дано действительное число x . Вычислить y .

Программа

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

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

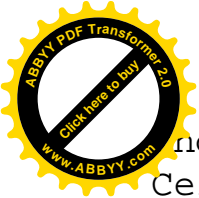
Программа

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

Пример. Дано действительное число x . Вычислить z, y, h , если известно, что

Программа

```
Sub Raz3()  
Dim x, z, y, h As Double  
x = Val(TextBox("Введите x"))  
If x > 0.8 Then  
z = 2 * Sin(x)  
y = Log(x) + 4 * x  
h = Cos(x)  
Else  
If x = 0.8 Then  
z = Sqr(Sin(x))  
y = Cos(x ^ 2) + x  
h = 2 * x  
Else  
z = Abs(x - 2)  
y = 2 + x ^ 2 * Sin(x)  
h = 0  
End If  
End Sub
```



```
End If  
Cells(1, 1) = "x=": Cells(1, 2) = x  
Cells(2, 1) = "z=": Cells(2, 2) = z  
Cells(3, 1) = "y=": Cells(3, 2) = y  
Cells(4, 1) = "h=": Cells(4, 2) = h  
End Sub
```

7.4. Программирование алгоритмов циклической структуры

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

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

Операторы цикла

Оператор цикла *While... Wend*

Оператор имеет синтаксис

While *условие*

[*операторы*]

Wend

Синтаксис инструкции *While... Wend* содержит следующие элементы:

условие — обязательный элемент. Числовое выражение или строковое выражение, которое имеет значение True или False.

операторы — необязательный элемент. Один или несколько операторов, выполняемых, пока условие имеет значение True.

Этот оператор называют оператором цикла с предусловием. Выполняется оператор *While... Wend* следующим образом. Если *условие* имеет значение True, выполняются все операторы до инструкции *Wend*. Затем управление возвращается инструкции *While* и вновь проверяется *условие*. Если *условие* по-прежнему имеет значение True, процесс повторяется. Если оно не имеет значение True, выполнение возобновляется с инструкции, следующей за инструкцией *Wend*. В связи с этим элемент *условие* здесь является условием выполнения цикла. Циклы *While... Wend* могут иметь любую глубину вложенности.

Оператор цикла *Do...Loop*

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

Do

[*операторы*]



Loop Until [условие]

Синтаксис инструкции Do Loop содержит следующие элементы:

условие — необязательный элемент. Числовое или строковое выражение, которое имеет значение True или False.

операторы— один или несколько операторов, выполнение которых повторяется, пока условие не приобретет значение True.

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

Оператор цикла **For...Next**.

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

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

Синтаксис инструкции For...Next

For *счетчик* = *начало* To *конец* [step *шаг*]

[*операторы*]

Next [*счетчик*]

счетчик — обязательный элемент. Это должна быть числовая переменная. Она не может иметь тип Boolean или быть элементом массива.

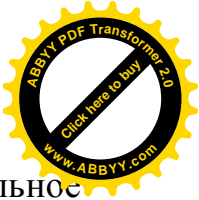
начало — обязательный элемент, содержит начальное значение переменной *счетчик*.

конец — обязательный элемент, содержит конечное значение переменной *счетчик*.

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

операторы— необязательный элемент. Один или несколько операторов между For и Next, которые выполняются указанное число раз.

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



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

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

Программа

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

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

Программа

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

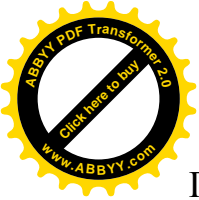
Пример. Вычислить произведение квадратов n чисел: 1,4,9,16,25,... . Число n задать произвольно при вводе.

Программа

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

Пример. Дана числовая последовательность:

Вычислить сумму данной последовательности при $n=\infty$ с точностью 0.000001.



Программа

```
Sub Iter()  
Dim a, e, s As Double  
a = 1 'первый член ряда  
s = a 'сумма ряда  
e = Val(InputBox("Введите точность вычислений"))  
Do 'начало цикла  
a = -a / 2 'вычисляем очередной член  
последовательности  
s = s + a 'накапливаем сумму  
Loop Until Abs(a) < e 'конец цикла  
MsgBox ("Сумма s=" + Str(s#))  
End Sub
```

Вложенные циклы

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

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

Программа

```
Sub V1Circle()  
Dim x, s, a, b, h As Double, i, n, k As Integer  
a = Val(InputBox("Введите a"))  
b = Val(InputBox("Введите b"))  
h = Val(InputBox("Введите шаг h"))  
n = Val(InputBox("Введите количество слагаемых n"))  
k = 1  
Cells(1, 1) = "X": Cells(1, 2) = "S"  
For x = a To b Step h  
s = 0  
For i = 1 To n  
s = s + x / i  
Next i  
k = k + 1  
Cells(k, 1) = x: Cells(k, 2) = s  
Next x  
End Sub
```




ЛЕКЦИЯ №8

8. МАССИВЫ

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

Оператор описания DIM

Оператор Dim применяется для описания переменных и выделения для них памяти.

Синтаксис инструкции

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

имя переменной — обязательный элемент, содержит имя переменной;

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

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

Например:

`Dim Num As Integer` явно объявляет переменную целого типа .

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

`Dim B(3,3) As Single`

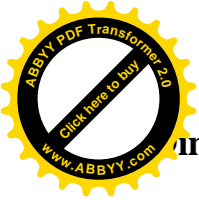
`Dim A(11) As Integer`

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

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

`Dim A(1 To 12) As Integer`

Массив в программе определяется поэлементно. Например:



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

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

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

8.1. Одномерные массивы

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

Dim A As Variant

A= Array(10,40,60)

Здесь A(0)=10, A(1)=40, A(2)=60.

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

Рассмотрим некоторые **алгоритмы обработки одномерных массивов.**

Поиск максимального элемента

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

Программа

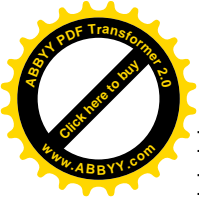
```
Sub Massiv()  
Dim a(5) As Single  
Dim max As Single, i, k As Integer  
'ввод массива  
For i = 1 To 5  
a(i) = Cells(1, i + 1)  
Next i  
max = a(1)  
k = 1 'индекс максимального элемента  
For i = 1 To 5  
If max < a(i) Then max = a(i): k = i  
Next i  
MsgBox ("Максимальный элемент=" + Str(max!))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

Поиск количества элементов по данному правилу

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

Программа

```
Sub Kol()
```



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

Нахождение элементов массива по элементам другого массива

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

Программа

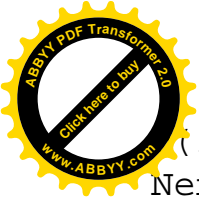
```
Sub NewMassiv()
Dim a(5), b(5) As Single
Dim i As Integer
'ввод массива a(5)
For i = 1 To 5
a(i) = Cells(1, i + 1)
Next i
For i = 1 To 5
b(i) = Sin(a(i))
Next i
'вывод нового массива b(5)
Cells(3, 1) = "Массив b(5)"
For i = 1 To 5
Cells(3, i + 1) = b(i)
Next i
End Sub
```

Умножение двух векторов

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

Программа

```
Sub SkalProiz()
Dim a(5), c(5) As Single
Dim i As Integer, s As Single
'ввод массива a(5)
For i = 1 To 5
a(i) = Cells(1, i + 1).Value
Next i
'ввод массива c(5)
For i = 1 To 5
```



```
a(i) = Cells(7, i + 1).Value  
Next i  
s = 0  
For i = 1 To 5  
s = s + a(i) * c(i)  
Next i  
MsgBox ("Скалярное произведение s=" + Str(s))  
End Sub
```

8.2. Некоторые алгоритмы обработки матриц

Поиск минимального элемента матрицы

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

Программа

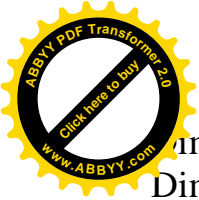
```
Sub MinMat()  
Dim a(2, 3) As Single  
Dim min As Single, i, j As Integer  
'ввод матрицы  
For i = 1 To 2  
For j = 1 To 3  
a(i, j) = Cells(i + 1, j)  
Next j  
Next i  
min = a(1, 1)  
For i = 1 To 2  
For j = 1 To 3  
If a(i, j) < min Then min = a(i, j)  
Next j  
Next i  
MsgBox ("Минимальный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

Умножение матрицы на вектор

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

Программа

```
Sub UmnMatNaVec()
```

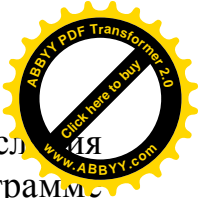
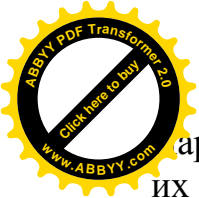


```
Dim a(2, 3) As Single, b(3) As Single, c(2) As Single
Dim s As Single, i, j As Integer
'ввод матрицы
For i = 1 To 2
For j = 1 To 3
a(i, j) = Cells(i + 1, j)
Next j
Next i
'ввод массива
For i = 1 To 3
b(i) = Cells(4, i + 1)
Next i
For i = 1 To 2
s = 0
For j = 1 To 3
s = s + a(i, j) * b(j)
Next j
c(i) = s
Next i
'вывод нового массива b(5)
Cells(6, 1) = "Массив c(2)"
For i = 1 To 2
Cells(6, i + 1) = c(i)
Next i
End Sub
```

ЛЕКЦИЯ №9

9. ПОДПРОГРАММЫ И ФУНКЦИИ

При разработке макросов часто требуется выполнять одни и те же законченные действия в различных его частях. Чтобы избежать многократного набора кода для этих действий, целесообразно описать их в виде подпрограммы и обращаться к ней по мере необходимости. Различают подпрограммы-процедуры и подпрограммы-функции. Описываются подпрограммы при помощи инструкций Sub...End Sub— это подпрограмма-процедура, инструкций Function...End Function — это подпрограмма-функция. Подпрограмма Function и подпрограмма Sub являются самостоятельной программой, которая может получать аргументы, выполнять последовательность операторов и изменять значения своих аргументов. Различие между процедурами и функциями состоит в том, что функция всегда возвращает значение и процедура Function может применяться в правой части выражения, как и любая другая встроенная функция, например, Sqr, Cos или Chr, а процедура Sub не может. Для работы подпрограммам иногда необходимо передать данные, которые носят названия параметров или аргументов подпрограммы.



параметры передаются в подпрограммы при ее вызове при помощи перечисления их после имени подпрограммы. При этом описание действий в подпрограмме осуществляется с использованием формальных параметров. Обращение к подпрограмме осуществляется с фактическими параметрами, которые должны соответствовать формальным по числу, типу и месту расположения.

Оператор Sub...End Sub

Описывает имя, аргументы и текст программы, составляющий тело процедуры Sub.

Синтаксис инструкции

Sub *имя* [(списка аргументов)]

операторы

End Sub

Здесь

имя — обязательный элемент, это имя процедуры Sub, удовлетворяющее стандартным правилам именования переменных.

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

операторы — любая группа операторов, выполняемых в процедуре Sub, находятся между Sub и End Sub.

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

Оператор вызова подпрограмм Call

Обращение к процедуре осуществляется с помощью оператора Call.

Синтаксис оператора:

Call *имя подпрограммы* [*параметры*]

Синтаксис инструкции Call содержит следующие элементы:

имя подпрограммы — обязательный элемент, содержит имя вызываемой процедуры.

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

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

Оператор Function...End Function

Описывает имя, аргументы и текст программы, составляющий тело подпрограммы- функции Function.

Синтаксис инструкции

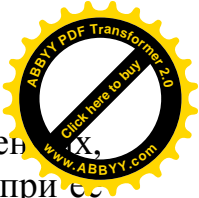
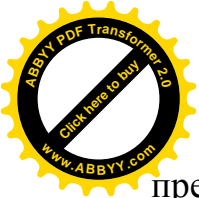
Function *имя* [(список аргументов)] [As *тип*]

операторы

[*имя* = *выражение*]

End Function

имя — обязательный элемент. Содержит имя подпрограммы-функции Function, удовлетворяющее стандартным правилам именования переменных;



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

тип — необязательный элемент. Тип данных значения, возвращаемого подпрограммой Function.

операторы— элемент, содержащий любую группу операторов, выполняемых внутри процедуры Function.

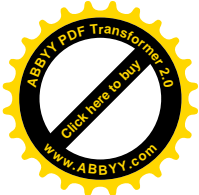
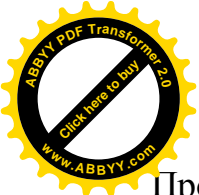
выражение — возвращаемое значение подпрограммой Function.

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

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

Программа

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



Пример. Решить систему уравнений методом Гаусса.

Программа

```
Sub Method()  
Dim a(3, 4), x(3), a1(3, 3)  
n = 3  
For i = 1 To n  
For j = 1 To n + 1  
a(i, j) = Cells(i + 1, j)  
Next j  
Next i  
For k = 1 To n - 1  
Call max(a(), n, k)  
Call gauss(a(), n, k)  
Next k  
Call obr(a(), n, x())  
Cells(6, 1) = "Вектор x"  
For i = 1 To n  
Cells(6, i + 1) = x(i)  
Next i:End Sub  
Sub max(a(), n, k)  
h = Abs(a(k, k))  
For i = k To n  
If Abs(a(i, k)) > h Then h = Abs(a(i, k)): l = i  
Next i  
For j = 1 To n + 1  
s = a(k, j): a(k, j) = a(l, j): a(l, j) = s  
Next j  
End Sub  
Sub gauss(a(), n, k)  
For i = k + 1 To n  
d = a(k, k)  
w = a(i, k)  
For j = k To n + 1  
a(i, j) = a(i, j) - a(k, j) / d * w  
Next j  
Next i  
End Sub  
Sub obr(a(), n, x())  
x(n) = a(n, n + 1) / a(n, n)  
For k = n - 1 To 1 Step -1  
s = 0  
For j = k + 1 To n  
s = s + a(k, j) * x(j)  
Next j  
x(k) = (a(k, n + 1) - s) / a(k, k)  
Next k  
End Sub
```




10. ОТЛАДКА ПРОГРАММ

10.1. Ошибки при создании программы

Можно выделить два типа ошибок, с которыми сталкивается программист. Во-первых - это ошибки, которые сопровождают создание программ, а во вторых - *ошибки времени выполнения*.

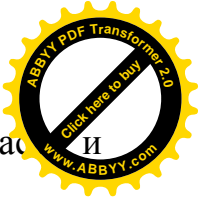
Если вы неправильно введете оператор или ключевое слово, если забудете указать часть выражения - ошибка сразу же будет заметна в редакторе. Такие ошибки относятся к *синтаксическим*. Также на этапе разработки можно обнаружить и устранить *логические ошибки*. Они связаны с неправильными формулами расчета показателей, неверным использованием переменных и т.д.

Приведем несколько примеров *синтаксических ошибок*.

- Неправильное ключевое слово. Такая ошибка сразу будет отмечена редактором. Например, если в конструкции цикла **For-Next** сделать ошибку в ключевом слове **For** - вы увидите сообщение об ошибке как только попытаетесь перейти к написанию следующей, после начала цикла, строки.
- Ошибка при объявлении переменной. Если вы забудете указать ключевое слово **Dim** при объявлении переменной - редактор выдаст сообщение об ошибке. Однако, сделает он это лишь при попытке запуска программы.

А вот - *логические ошибки*.

- Неправильное использование операторов. Например, вместо знака обычного деления вы случайно использовали знак целочисленного деления.
- Расчет какого-либо показателя по неправильной формуле. Например, если неточно расставить скобки в каком-либо выражении - это приведет к ошибке, хотя внешне все может выглядеть правильным.
- Неправильное использование функций - как встроенных, так и пользовательских. Например, используя функцию **Str** для получения строкового представления числа, вы не учли, что для положительных чисел эта функция добавляет в начало строки пробел. Далее вы попытались узнать первую цифру числа, вырезав первый символ полученной строки. Естественно, никакой цифры в этом случае не получится - лишь знак пробела для положительных или "минус" для отрицательных чисел.
- Неправильное использование переменных. Например, вы используете два вложенных цикла для обработки двумерного массива. Одна из цикловых переменных имеет имя **i**, вторая - **j**. Они довольно сильно похожи внешне, их можно случайно перепутать при указании индексов массива. К тому же, обрабатывая массив в цикле довольно легко перепутать место каждой из переменных при указании индекса массива. Использование понятных имен



- переменных (например - `my_Age` или `num_Vozrast` для хранения возраста и т.д.) позволяет эффективно бороться с такими ошибками.
- Случайное использование "новых" переменных. Например, вы предложили пользователю ввести некоторое значение и записали его в переменную `num_Inp`, а используя эту переменную в выражении, напечатали не `num_Inp`, а `num_Ihp`. Внешне они похожи, но, присмотревшись, вы можете обнаружить, что имена разные. Еще сложнее искать ошибки в латинских именах переменных, в которые "вкрались" русские буквы. Разницу между `my_Name` и `my_Name` вы не увидите, но это - разные переменные - в конце второй вместо латинской `e` использована русская `е`. Эффективно бороться с такими ошибками можно, если задать в редакторе опцию обязательного объявления переменных (**Option Explicit**). При появлении необъявленной переменной редактор даст знать об этом.
 - Неправильное использование оператора сравнения. Например, это может быть оператор, который сравнивает некие величины не так, как вы предполагали - вместо знака `<` вы случайно использовали `>` или, редактируя сравнение (скажем, скопировав похожий оператор сравнения из другого места программы для ускорения работы), поменяли местами сравниваемые переменные или выражения, не поменяв знака и т.д.

Ошибки логики можно найти лишь тщательно проверив и протестировав программу. Чтобы облегчить работу с кодом программы, рекомендуется снабжать ее комментариями. Вовсе необязательно комментировать каждую строчку, достаточно выделить крупные функциональные блоки.

10.2. Ошибки при выполнении программы

Такие ошибки называют еще *ошибками времени выполнения*. Происходят они, как правило, при неправильном вводе данных пользователем, при возникновении обстоятельств, делающих дальнейшую нормальную работу программы невозможной. Например, ошибку вызовет попытка использовать текстовые данные в арифметическом выражении, попытка сохранения файла в несуществующей директории, деление на ноль и т.д.

Ошибки времени выполнения возникают в нормально работающих программах, которые прошли проверку на синтаксическую и логическую правильность. С этими ошибками можно бороться используя один из двух методов. Первый - разработка программы таким образом, чтобы не допустить этих ошибок, создание программных конструкций, которые предотвращают возникновение ошибок. Второе - перехват ошибок и их обработка.

Давайте, для начала, рассмотрим способы поиска ошибок в процессе создания программ



10.3. Тестирование программ и поиск ошибок

Синтаксические ошибки, как уже было сказано, выявляются обычно на очень раннем этапе - сразу после набора текста в редакторе. А вот для поиска *логических ошибок* полезно протестировать программу.

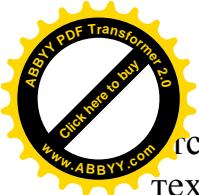
Для начала протестируйте вашу программу в обычном режиме работы. Например, если вы предлагаете ввести пользователю число - введите сами число и проверьте результаты работы. Если действия программы будут различаться в зависимости от введенных данных - попытайтесь перебрать все варианты этих данных - так, чтобы проверить все ветви программы.

Если обычные проверки закончились успешно - можно считать, что ваша программа корректно работает, не имеет *синтаксических* и *логических ошибок*.

После этого можно усложнить режим тестирования, чтобы как можно более полно симитировать действия пользователя программы и посмотреть на ее поведение. Еще лучше - попытайтесь целенаправленно вызвать ошибку. Гораздо лучше, когда программа даст сбой на вашем рабочем столе, и вы сможете оперативно устранить этот сбой, чем ждать, когда то же самое произойдет у другого пользователя.

Для проверки программы в реальных условиях попробуйте провести следующие эксперименты.

- Если программа запрашивает число - введите какое-нибудь слово, очень большое число, ноль, отрицательное число, оставьте поле ввода пустым, введите дробное число. Если вы не предусмотрели никаких специальных мер по обработке ошибок, при выполнении подобных операций вы почти гарантированно встретитесь с проблемами.
- Попробуйте запустить программу, открыв несколько окон с документами.
- Попробуйте прервать работу программы, а потом снова возобновить ее. Если в вашей программе есть участки, в течение выполнения которых нельзя допускать прерывания работы программы пользователем - вам следует подумать о том, чтобы запретить прерывание работы программы на этих участках.
- Попробуйте использовать вашу программу в более старой версии Microsoft Office, в нерусифицированной версии. Обратите внимание, например, на различия расширений имен файлов в Office 2007 и более старых версиях, на различия в объектных моделях. В общем случае программы из более старых версий Office будут работать в Office 2007, однако если они используют какие-то специфические особенности Office - такие программы нуждаются в проверке и обновлении. В то же время, макросы для Office 2007, использующие новые объектные модели, могут не работать или работать неправильно в старых версиях.
- Попробуйте поработать с вашей программой на чужом ПК. Вполне возможно, что при таком эксперименте вы столкнетесь с ошибкой. Например, вы программно работаете с файлами на вашем рабочем ПК - если эти файлы



отсутствуют на ПК другого пользователя или находятся в других директориях тех директорий, которые нужны вашей программе, нет на ПК другого пользователя - вы столкнетесь с ошибкой.

- Во время работы программы сделайте что-нибудь необычное. Как правило, от пользователей можно ожидать любых странных на первый взгляд действий. Если вы тестируете программу для MS Word, которая правит текст или занимается автоматическим созданием текста, попробуйте во время ее работы переключаться между документами, читать документ, вносить в него правки, выделять произвольные участки текста. То же самое касается MS Excel - во время работы программы попробуйте переключаться между открытыми книгами, между листами, выделять ячейки, попробуйте запустить программу, делая различные листы активными, открыв несколько книг. Результаты такого тестирования могут быть совершенно непредсказуемыми. Проанализировав их, вы можете прийти к выводу, что, например, на время выполнения программы нужно скрывать или блокировать документ, пользоваться альтернативными методами работы с документом. Яркий пример - объект **Selection** в MS Word, который чувствителен к смене выделения в процессе работы, и объект **Range**, который может работать совсем без создания выделения в тексте.

- Если ваша программа использует файлы, находящиеся в локальной сети, отключите сеть во время работы программы. Проверьте ее реакцию. То же самое можно сделать, если ваша программа работает с принтером - проверьте ее реакцию на выключенный принтер, на принтер, в котором нет бумаги.

- Наконец, представьте, что вы - пользователь программы и просто поработайте с ней. А еще лучше - попросите потенциального пользователя немного "пообщаться" с вашей программой. Этот способ позволяет протестировать программу в условиях, максимально приближенных к реальным.

Возможно, вам покажется, что такая проверка не нужна вашей программе. Если вы пишете небольшой макрос для собственного использования, скорее всего, так оно и есть. Но стоит вашему проекту хоть немного вырасти, методы поиска и устранения ошибок могут оказаться очень кстати.

Если вы встретитесь с ошибками и затрудняетесь определить, где именно они происходят - попробуйте выполнить программу в пошаговом режиме, используйте встроенные в редактор VBA средства для *отладки программ*.

10.4. Отладка программ в редакторе VBA

Основной метод отладки - это пошаговое исполнение программы с использованием точек останова (breakpoint).

Чтобы создать в программе точку останова, достаточно щелкнуть мышью в редакторе на серой панели напротив команды, на которой нужно остановить выполнение программы. Там появится большая красная точка (рис. 10.1.) - здесь программа будет остановлена в процессе выполнения. Строка будет подсвечена красным цветом.

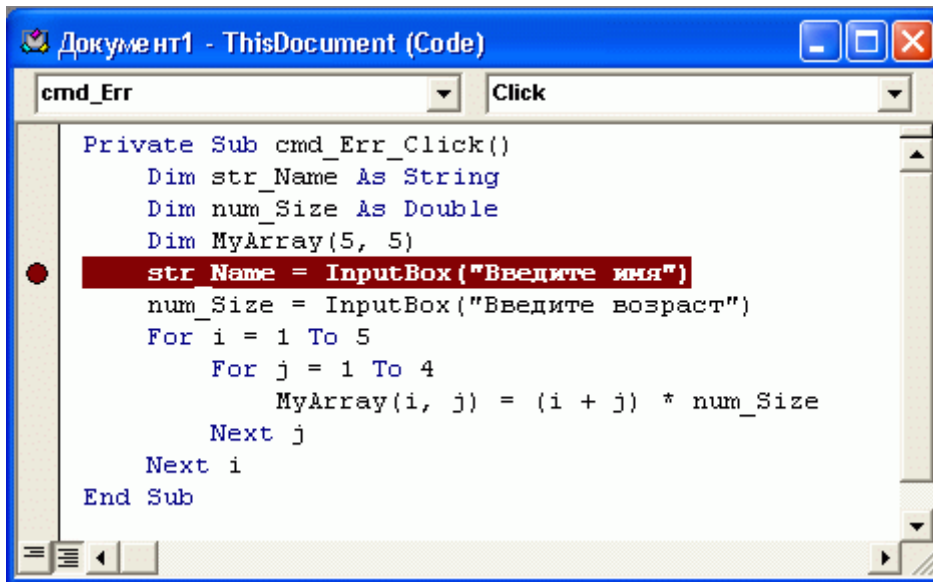


Рис. 10.1. Точка останова в программе

Следует понимать, что строка, подсвеченная при остановке программы еще не выполнялась - редактор указывает на нее, как бы говоря "Эта строка будет выполнена следующей".

Таких точек останова можно установить столько, сколько нужно - на тех строках программы, где вы подозреваете возникновение ошибки. Чтобы убрать точку останова, щелкните по ней мышью.

Установить точку останова в строку можно, выделив строку и выбрав команду **Debug o Toggle Breakpoint** (Отладка o Установить точку останова). Для удаления всех точек останова из программы можно воспользоваться командой **Debug o Clear All Breakpoints** (Отладка o Очистить точки останова).

Вместо точек останова, расставленных мышью или из меню, можно использовать оператор **Stop**. Он останавливает работу программы и переводит ее в режим *отладки*.

Так же режим *отладки* можно включить, нажав во время работы программы комбинацию клавиш **Ctrl + Pause Break** и нажав в появившемся окне кнопку **Debug**. Текущая строка будет выделена желтым цветом, напротив нее будет установлена желтая стрелочка. Однако такой способ обычно не позволяет точно "попасть" в то место программы, где находится предполагаемая ошибка (рис. 10.2.).

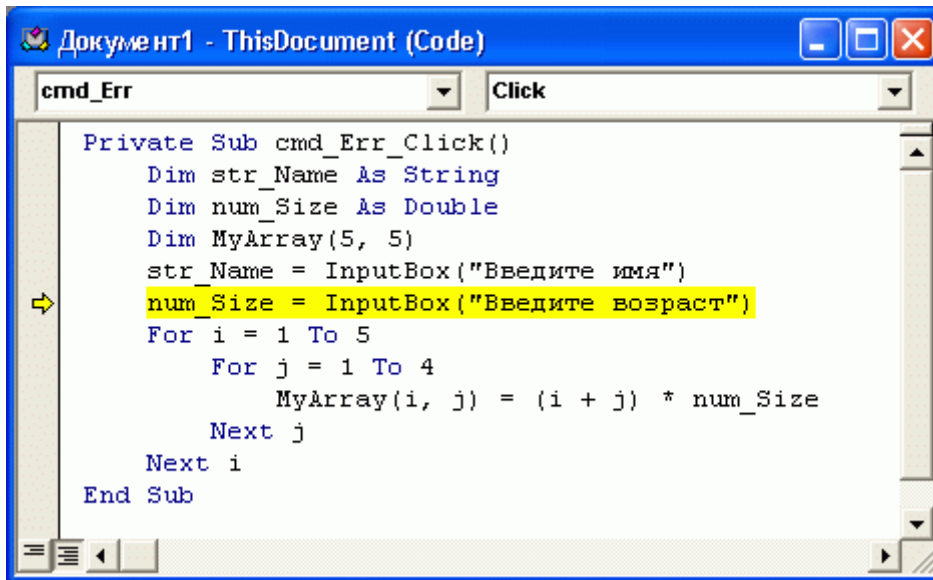


Рис. 10.2. Программа, остановленная во время выполнения

Чтобы запущенная программа останавливалась на каждой строке, можно запустить ее в режиме **Step Into** командой **Debug o Step Into** (Отладка o Пошаговое исполнение). Того же эффекта можно достичь, нажав клавишу **F8** на клавиатуре.

Когда программа остановлена, вы можете выполнить следующие действия

- Просмотреть значения переменных, наведя на них указатель мыши. Например, на рис. 10.3. вы видите всплывающее окно, которое содержит значение переменной.

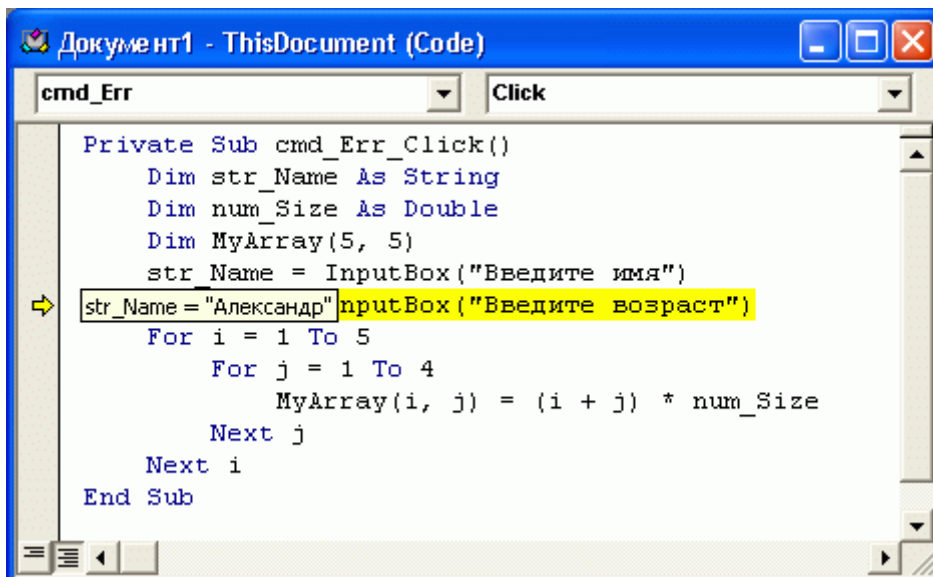
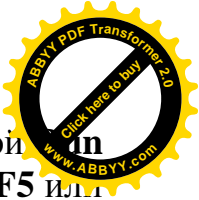
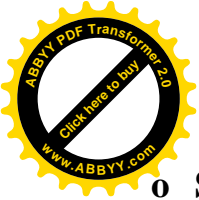


Рис. 10.3. Значение переменной во всплывающем окне

- Продолжить выполнение программы в режиме **Step Into** - выбрав соответствующую команду меню или нажав клавишу **F8**.
- Отредактировать программу.



- Продолжить исполнение программы в обычном режиме командой **Run o Sub/User Form** (Запустить o Процедуру/Форму), нажатием клавиши **F5** или соответствующей кнопкой на панели инструментов
- Остановить выполнение программы командой **Run o Reset** (Запустить o Перезагрузка) или кнопкой на панели инструментов
- Воспользоваться другими средствами *отладки* - окнами **Immediate, Locals, Watch**.

Помимо режима **Step Into** существуют следующие режимы отладки, доступные в меню **Debug**.

- **Step Over** (Перейти на следующую строку). Эта команда полезна при *отладке программы*, содержащей вызовы уже отлаженных процедур. В режиме **Step Over** отладчик не входит в процедуру, выполняя ее без *отладки*, после чего переходит на следующую строку. Например, вы выполняете программу в режиме **Step Into** и при очередной остановке видите, что подсвеченная строка содержит вызов процедуры, которую отлаживать не нужно. Вы выбираете команду **Step Over**, процедура выполняется без остановок на каждой ее строке, после чего следующая остановка происходит на строке вашей программы, которая идет за вызовом процедуры.

- **Step Out** (Выполнить процедуру) - эта команда позволяет выполнить текущую процедуру (например, вызванную из кода основной программы при обычной *отладке*) без остановки в каждой строке. Следующая остановка будет сделана на строке, которая следует за вызовом процедуры в основном тексте программы.

- **Run To Cursor** (Выполнить до курсора) - выполняет программу до позиции, на которой установлен курсор. Аналогично установке одиночной точки останова.

Помимо точек останова существуют и другие средства *отладки*. Они полезны при проверке значений переменных (ведь если переменных достаточно много - проверка их значений в коде программы может превратиться в утомительное и непродуктивное занятие), свойств объектов, которые могут вызвать ошибки и в других случаях.



ЛЕКЦИЯ №11

11. ПРИКЛАДНЫЕ ПРОГРАММЫ

В настоящее время большую популярность среди пользователей получил офисный пакет прикладных программ Microsoft Office, разработанный компанией Microsoft.

Этот пакет представляет собой комплекс совместимых между собой программ, призванный образовать для пользователя единую, комфортную производственную среду. В состав этого пакета входят:

- текстовый редактор Microsoft Word,
- электронные таблицы Microsoft Excel,
- система управления базами данных Microsoft Access,
- программа подготовки компьютерных презентаций Power Point,
- программа – планировщик рабочего времени (блокнот) Outlook,
- графический редактор Paint,
- коммуникационный пакет Microsoft Internet Explorer, включающий в

себя:

- программу для просмотра Интернета (браузер) Internet Explorer,
- программу электронной почты Outlook Express,
- программы для путешествия по Интернету Net – Meeting и

Windows Messenger, а иногда и программу для создания Web – страниц Front page и Front Page Express.

11.1. Текстовый редактор Microsoft Word

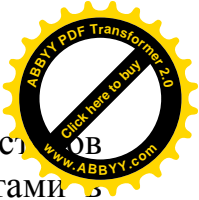
Очень часто инженеру и научному работнику в своей профессиональной деятельности приходится создавать различного рода документы (технические и научно – технические отчеты, статьи, инструкции и др.). Как правило, эти документы содержат текстовую часть с различными математическими формулами, таблицами, графиками, рисунками и др. При этом такая документация должна оформляться в соответствии с установленными требованиями (шрифты, нумерация страниц и разделов, отступы, поля и др.). Зачастую, при создании такого рода документов, в них приходится вносить какие – либо изменения (редактировать), тиражировать. Все эти действия и позволяет выполнять текстовый редактор Microsoft Word. В настоящее время этот редактор является самым мощным редактором в мире.

Назначение и основные возможности MS Word

Текстовый редактор MS Word предназначен для *создания, редактирования, сохранения и печати* текстовых документов различной степени сложности и на различных языках.

Основными возможностями текстового редактора являются:

- наличие мощной справочной системы и мощной контекстной справки;
- изменение форм текстового курсора и указателя мыши при изменении текущей ситуации;



- богатый набор заготовок (шаблонов) документов, наличие маск-подсистем, автоматизирующих работу над стандартными документами в стандартных условиях;
- ввод и редактирование текстов на большом количестве языков мира (в редакторе MS Word 2002 поддерживается более 80 языков);
- ввод текста с автоматическим формированием строк, абзацев и страниц;
- ввод, просмотр и редактирование документов практически произвольных размеров (ограничение – объем свободной дисковой памяти);
- возможность импорта и экспорта;
- наличие нескольких режимов работы с документом;
- большой выбор шрифтов, их стилей и размеров;
- большой выбор возможностей форматирования абзацев и страниц документа;
- поиск и замена подстрок;
- наличие буфера обмена;
- механизмы отмены и восстановления после последней отмены выполненных действий;
- возможность включать в документ обширный круг специальных символов, рисунков, иллюстраций, диаграмм и др.;
- наличие средств автоматизации работы с документами – автотекст, автозамена, автоформат, автоперенос и т. д.;
- автоматизация подготовки списков;
- автоматизация построения таблиц;
- возможность изменения ориентации текста (наклон, поворот, переворот);
- возможность подготовки рисунков средствами редактора;
- наличие средств ввода математических формул различной сложности;
- работа с многоколоночными документами (газетными, журнальными статьями);
- проверка орфографии с помощью встроенного словаря;
- проверка синтаксиса;
- подбор синонимов;
- одновременная работа с несколькими документами;
- рассылка документов по локальной сети;
- создание макросов – средств, обеспечивающих автоматизацию часто выполняемых однотипных операций.

Файлы, созданные в текстовом редакторе MS Word, сохраняются с расширением *.doc*. Этот редактор обеспечивает возможность работать с файлами в общепринятых форматах TXT и RTF, а также с файлами в форматах наиболее распространенных приложений.

Интерфейс редактора

Интерфейс – это способ общения пользователя с компьютером. В текстовом редакторе MS Word 2000 в качестве способа общения человека с компьютером приняты *окна*, как и в операционной системе Windows.

Окно редактора имеет стандартную, как и для окон Windows структуру. В верхней части окна располагается:



- строка заголовка окна, в которой указывается имя создаваемого редактируемого файла;
- строка операционного меню, которая состоит из названий видов выполняемых операций и услуг, предоставляемых редактором;
- панель инструментов, представляющая собой одну или несколько строк, состоящих из кнопок, обозначающих различные часто выполняемые операции. Панель инструментов не является обязательным атрибутом окна. Она может показываться или нет по желанию пользователя.

Далее следует рабочая зона окна, в которой располагается, создаваемый или редактируемый текст. Вверху и слева этого окна могут выводиться *линейки*, создающие дополнительные удобства пользователю в части контроля размера текста на странице по горизонтали и вертикали. Справа и внизу располагаются *полосы прокрутки*, позволяющие перемещать текст по вертикали и горизонтали.

Ниже основного окна располагается *строка состояния*. В нее выводится вся текущая информация о создаваемом или редактируемом документе. В режиме ввода текста строка состояния содержит следующие управляющие и информационные элементы:

- номер текущей страницы документа (**Стр. 5**);
- номер текущего раздела (**Разд. 1**);
- количество страниц от начала документа до текущей страницы и общее количество страниц в документе (**3/18**);
- расстояние от верхней кромки текущего листа до строки, в которой находится текстовый курсор (**На 24,5см**);
- текущая позиция текстового курсора: номер строки и колонки (**Стр. 41 Кол 6**);
- указатели режимов записи макрокоманд **ЗАП**, исправления документа **ИСПР**, расширенного выделения **ВДЛ**, вставки/замены **ЗАМ**;
- указатель настройки языка (русский, английский, украинский и др.);
- указатель проверки правописания.

Указатели режимов **ЗАП**, **ИСПР**, **ВДЛ**, **ЗАМ** могут быть заблокированными, то есть выключенными (представляются блеклым шрифтом) или активными, то есть включенными (представляются ярким шрифтом).

Основное меню

Основное меню содержит перечень всех видов работ с документом. Оно содержит следующие пункты:

- *Файл* – содержит стандартные команды выполнения файловых операций;
- *Правка* – содержит команды, необходимые для редактирования документа;
- *Вид* – состоит из команд, определяющих внешний вид окна редактора и способ изображения документа в окне;
- *Вставка* – содержит команды, позволяющие производить различные вставки в документ от номера страницы до математических формул;
- *Формат* – содержит различные команды форматирования документа;
- *Сервис* – содержит команды проверки орфографии, включения автопереноса и ряд других;



- *Таблица* – содержит команды, необходимые для создания и редактирования таблиц;
- *Окно* – обеспечивает работу с окнами документов;
- *Справка* – служит для обращения к справочной подсистеме редактора.

Панель инструментов

Панель инструментов не является обязательным элементом окна редактора, но ее присутствие облегчает выполнение различных редакционных операций. Эта панель представляет собой набор различных кнопок. Каждая кнопка имеет свое назначение и обеспечивает определенную операцию. «Нажатие» (одинарный щелчок мышью) на какую – либо кнопку вызывает выполнение этой операции. Реализацию таких операций можно осуществлять с помощью пунктов меню, но этот путь все-таки длиннее, чем с помощью кнопки. Набор кнопок может быть *определен пользователем по своему усмотрению.*

В редакторе MS Word предусмотрено большое количество стандартных панелей, но наиболее часто используются следующие из них:

- *стандартная*, содержащая часто используемые элементы управления (кнопки) (открытие и сохранение файла, работа с буфером обмена, откат и накат, вызов справки и др.);
- *форматирование*, содержит кнопки, которые вызывают различные действия по оформлению символов, абзацев, и страниц текста;
- *таблицы и границы*, состоит из кнопок по форматированию рамок, таблиц и закрашиванию участков текста;
- *рисование*, содержит кнопки, обеспечивающие выполнение операций по созданию рисунков, схем в редактируемом документе.

В стандартно настроенном окне редактора MS Word присутствует две панели: *стандартная* и *форматирование*.

11.2. Электронные таблицы. MS Excel. Основные представления

При решении задач, связанных с большим объемом обрабатываемой информации, удобно представлять данные в виде *таблиц*, состоящих из *строк* и *столбцов*. Например, один столбец таблицы может быть заполнен списком типов электродвигателей, другой их мощностью P , третий напряжением U и т. п. Некоторые данные, помещаемые в другие столбцы, могут вычисляться по формулам, используя числа уже заполненных столбцов. Например, вычисление тока $I = P/U$. Иногда приходится вычислять сумму чисел в столбце. Например, общее число отказов электрооборудования за сутки, неделю, месяц и др. Подобные задачи удобно решать с помощью программ, называемых *электронными таблицами*. Существует большое количество различных электронных таблиц, но наиболее распространенной среди них следует считать Microsoft Excel, входящую в пакет программ Microsoft Office.

Эта таблица позволяет:

- выполнять вычисления с использованием обширной библиотеки встроенных функций;
- строить различного вида графики и диаграммы;
- производить анализ статистических данных;



- исследовать влияние различных факторов;
- решать задачи оптимизации;
- импортировать и экспортировать документы или их фрагменты, созданные в других распространенных пакетах.

Файлы, созданные с использованием программы MS Excel, по умолчанию получают расширение *.xls*, чтобы отличать их от других файлов.

Интерфейс программы MS Excel

Интерфейс этой программы имеет все те же особенности, что и программы текстового редактора MS Word. Он имеет:

- строку заголовка, создаваемого или редактируемого документа;
- строку основного меню с теми же пунктами, что и в MS Word;
- строки панелей инструментов (*стандартная* и *форматирование*);
- рабочее окно;
- строку состояния;
- строку формул, размещаемой под строками панели инструментов.

Рабочее окно представляет собой не чистое поле, а пустую таблицу, состоящую из столбцов и строк. Максимальный размер таблицы:

- 2^8 (256) столбцов,
- 2^{16} (65536) строк.

Верхняя строка таблицы содержит заголовки столбцов, в виде букв латинского алфавита: *A . . . Z, AA . . . AZ, BA . . . BZ, IA . . . IZ*. Левый столбец содержит порядковые номера строк: *1...65536*. Такая маркировка таблицы позволяет устанавливать адрес *ячейкам*, располагающимся на пересечении столбцов и строк. Адрес каждой ячейки *составной* и образуется из имени столбца и строки, на пересечении которых и размещается ячейка. Например, *A8, C26* и т.д.

Файл программы MS Excel может содержать не одну, а несколько таблиц, образуя *книгу* из нескольких *листов*. Поэтому документом программы MS Excel является *рабочая книга*. Максимальный размер рабочей книги – 255 листов (таблиц). При создании нового файла, по умолчанию, *образуется книга из трех листов*. Внизу рабочего окна располагаются закладки в виде наименования листов книги: *Лист1, Лист2, Лист3*. Щелкая мышью по этим закладкам, можно переходить на тот или иной лист, как бы листовая книга. Листы в рабочей книге можно добавлять, используя пункт меню **ВСТАВКА – ЛИСТ**. При этом добавленные листы будут отображаться в виде закладок *слева*, отодвигая имеющиеся закладки листов *вправо*.

Операции с ячейками

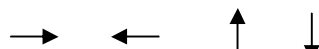
Любая ячейка может содержать до 255 символов, которые трактуются как *содержимое ячейки*. В общем случае содержимым ячейки может быть

- *текст*,
- *число*,
- *формула*.

График или диаграмма обычно занимает группу ячеек.

Выделение ячеек

Для того, чтобы в ячейку поместить какую-либо информацию, ее следует *активизировать* или, как иначе говорят, *выделить*. Эта операция осуществляется





щелчком мыши в плоскости выделяемой ячейки. Признаком выделения ячейки является появление вокруг нее жирного контура с небольшим квадратиком в правом нижнем углу. Этот квадратик называется маркером заполнения. Выделить нужную ячейку можно и с помощью клавиш управления курсором, перемещаясь к выделяемой ячейке.

Курсор, находящийся в поле таблицы, имеет вид крестика, который может перемещаться с помощью мыши.

Кроме выделения одиночных ячеек иногда возникает необходимость в выделении группы ячеек.

Выделение группы смежных ячеек

Группу смежных ячеек еще называют *интервалом* или *диапазоном*. Для выделения интервала ячеек следует установить курсор мыши в верхнюю, левую ячейку интервала, нажать левую клавишу мыши и, не отпуская ее, перемещать указатель мыши до правой нижней ячейки интервала. По ходу перемещения указателя, выделяемые ячейки закрашиваются голубым цветом. После завершения выделения левую клавишу мыши следует отпустить. Выделенный интервал остается закрашенным, кроме ячейки в верхнем левом углу.

Выделение нескольких несмежных интервалов

Для выделения *нескольких несмежных интервалов* необходимо нажать на клавиатуре клавишу *Ctrl* и, не отпуская ее, производить выделение всех несмежных интервалов так же, как это делается при выделении одного интервала. После завершения выделения, следует отпустить клавишу мыши и клавишу *Ctrl* на клавиатуре. Все выделенные интервалы, кроме последнего, полностью закрашиваются голубым цветом. У последнего интервала не закрашивается ячейка в верхнем левом углу.

Выделение отдельной строки и отдельного столбца

Для выделения *отдельной строки* следует щелкнуть мышью по ее заголовку.

Для выделения *отдельного столбца* следует щелкнуть мышью по его заголовку.

После выделения интервала ячеек заголовки строк и столбцов, чьи ячейки образуют интервал, изменяют свой шрифт на *жирный*.

Для *снятия выделения* интервала ячеек или группы интервалов следует щелкнуть мышью за пределами выделенных интервалов.

Синтаксис представления интервала ячеек в тексте или в формулах имеет вид: **A2:C8**. Здесь **A2** – адрес ячейки, размещенной в верхнем левом углу интервала, а **C8** – адрес ячейки, размещенной в нижнем правом углу интервала.

Более лаконично операции по выделению объектов таблицы MS Excel представлены в табл. 11.1.

В любой момент времени может быть активна *только одна книга, один лист, одна ячейка* или *одна группа ячеек*.

Имя активной книги *располагается в заголовке окна*, активный лист выделяется *белым цветом*.



Таблица 11.1

| Объект | Операции |
|-------------------------------|---|
| Ячейка | Щелкнуть мышью по выделяемой ячейке. |
| Строка | Щелкнуть мышью по заголовку строки. |
| Столбец | Щелкнуть мышью по заголовку столбца. |
| Интервал ячеек | Щелкнуть по ячейке в верхнем левом углу интервала и при нажатой левой клавише мыши перемещать мышь к ячейке в правом нижнем углу интервала, затем отпустить клавишу мыши. |
| Группа несмежных ячеек | Выделить один интервал, нажать клавишу Ctrl и, не отпуская ее, выделять все остальные интервалы. |
| Лист | Щелкнуть по закладке (ярлыку) выделяемого листа. |
| Книга | Щелкнуть пункт меню ОКНО , в раскрывшемся дополнительном меню, щелкнуть по имени выделяемой книги. |

ЛЕКЦИЯ № 12

12. ЭЛЕКТРОННАЯ ТАБЛИЦА MS EXCEL. ДАННЫЕ

12.1. Запуск и завершение программы MS Excel

Запуск программы осуществляется одним из приведенных способов:

- с помощью пунктов меню *ПРОГРАММЫ – MS Excel*;
- щелчком по ярлыку *MS Excel* на рабочем столе (если он выведен);
- щелчком по имени файла *excel.exe*, размещение которого следует предварительно найти.

Выход из программы *MS Excel* производится одним из способов:

- с помощью пунктов меню таблицы *ФАЙЛ – ВЫХОД*;
- нажатием комбинации клавиш *ALT + F4*.

12.2. Данные. Ввод данных

В качестве данных в MS Excel могут использоваться:

- *числа*;
- *текст*;

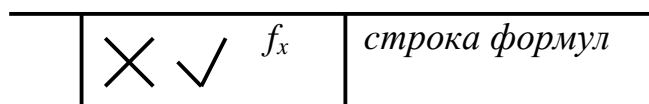


- даты;
- время;
- последовательные ряды данных.

Данные можно вводить двумя способами:

- непосредственно в выделенную ячейку,
- в поле панели «Строка формул».

Вид строки формул следующий:



Для ввода данных в строку формул необходимо щелкнуть по строке формул и после появления в ней текстового курсора начать ввод.

Ввод числовых данных

Максимальное количество цифр в числах, используемых в таблице – 15.

Числовые данные могут вводиться в виде:

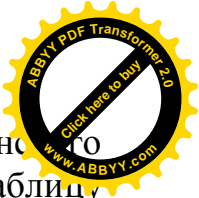
- целых чисел: 39; -126; 30721;
- десятичной дроби: 0,29; -5,67; 78,8042; Здесь целая часть отделяется от дробной запятой;
- обыкновенной дроби: 4 1/3 (четыре целых и одна третья); 0 5/6 (ноль целых и пять шестых). Обыкновенные дроби всегда должны содержать целую часть, даже нулевую. Это важно, потому что запись в виде 5/6 будет восприниматься как дата.
- действительные числа в экспоненциальной форме: 8,9E- 07; 1,5E+ 5;

Числовые данные выравниваются по **правому краю** ячеек.

Если количество цифр в дробной части вводимых чисел одинаково, то для облегчения ввода чисел целесообразно воспользоваться режимом **автоматического формирования дробной части чисел**. Для этого в окне **ПАРАМЕТРЫ** на вкладке **ПРАВКА** следует установить флажок **ФИКСИРОВАННЫЙ ДЕСЯТИЧНЫЙ ФОРМАТ ПРИ ВВОДЕ**, а затем в расположенном ниже настраиваемом поле **десятичных разрядов** указать *количество разрядов в дробной части числа*. После этого можно вводить все цифры числа подряд, не задумываясь над тем, где ставить запятую. Она будет устанавливаться автоматически. Например, если в поле десятичных разрядов установлено количество разрядов 3, то ввод последовательности цифр 35719 приведет к формированию числа 35,719. Можно указывать и *отрицательное количество разрядов в дробной части*. Это будет означать, что к введенным цифрам числа следует дописать соответствующее количество нулей. Например, в поле десятичных разрядов установлено (-4). Тогда к введенной последовательности цифр 47297 будет дописано четыре нуля и получится число 472 970 000. Такой режим облегчает ввод чисел.

Ввод текста

Текстовые данные состоят из букв, цифр и других символов. *Начинаться текстовые данные должны с буквы или апострофа*. Длина текста в ячейке не должна превышать 256 символов. Текстовые данные выравниваются по *левому краю*. Если текст должен начинаться с цифры, знака плюс или минус, то перед ними *должен устанавливаться апостроф*. Например, '396.



Тексты могут состоять не только из символов русского или латинского алфавита, но и из других символов. Вставлять их можно, используя таблицу символов. Для получения этой таблицы следует воспользоваться пунктами меню **ВСТАВКА – СИМВОЛ**, с последующим использованием предоставленной таблицы символов (как это делается в текстовом редакторе MS Word).

Режим автозаполнения

В MS Excel предусмотрен полезный режим *автозаполнения значений ячеек*. Под *автозаполнением* подразумевают *автоматическое копирование* содержимого одной или нескольких ячеек в выделенную группу ячеек или автоматическое заполнение последовательности ячеек какой-либо *регулярной последовательностью названий, чисел, дат и др.* Чтобы этот режим включить, следует выполнить цепь операций:

СЕРВИС–ПАРАМЕТРЫ

путем установки флажка перед строкой **АВТОЗАПОЛНЕНИЕ ЗНАЧЕНИЙ ЯЧЕЙКИ**.

Если этот режим включен, то программа сравнивает вводимый текст с тем, который вводился ранее. И если несколько первых символов вводимого текста совпадают с соответствующими символами ранее вводимого текста, то в ячейку копируется весь ранее вводимый текст. Дополненная часть текста остается выделенной. Если автозаполнение ячейки устраивает пользователя, он может закончить на этом ввод текста. В противном случае пользователю следует стандартным образом продолжать ввод текста, и первый же несовпадающий символ вводимого текста удалит из ячейки, выделенное автозаполнение.

Включение этого режима полезно в тех случаях, когда различные ячейки таблицы следует заполнять одним и тем же текстом. Для этого необходимо:

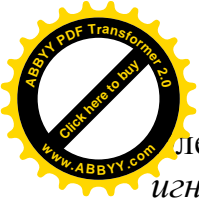
- при нажатой клавише **Ctrl** выделить все ячейки, в которые должно быть записано одинаковое значение,
- ввести это значение в последнюю выделенную ячейку,
- нажать комбинацию клавиш **Ctrl + Enter**.

Автозаполнением можно пользоваться при создании ряда ячеек с *автоматическим наращиванием значений дат, времени, дней недели, месяцев и др.* Если какую-либо ячейку заполнить текстом «1 квартал», а затем переместить маркер выделения на соседние три ячейки, то они заполнятся текстами «2 квартал», «3 квартал», «4 квартал». Если одну ячейку заполнить текстом «Январь», а затем перетащить маркер на соседние одиннадцать ячеек, то они заполнятся последующими месяцами года.

Для автозаполнения таких названий как дни недели, месяца, года программа MS Excel использует готовые (стандартные) списки. Вместе с тем пользователю предоставляется возможность сформировать свой список названий для автозаполнения. Это выполняется следующей последовательностью операций:

СЕРВИС – ПАРАМЕТРЫ – СПИСКИ

В последнем окне следует щелкнуть по кнопке **ДОБАВИТЬ**, затем в открывшемся окне **ЭЛЕМЕНТЫ СПИСКА** ввести необходимую последовательность названий, отделяя их друг от друга, нажатием клавиши **Enter**. При этом каждое название не должно превышать 80 символов, а весь список 2000 символов.



Элементы списка не должны начинаться с цифры, ошибочные значения и формы *игнорируются* (выпадают из списка).

Ввод даты и времени

В программе MS Excel предусмотрено несколько стандартных форм записи *дат* и *времени*.

Даты можно вводить в следующих форматах:

1.1.7 1-1-7 1/1/7 1- янв – 09

все эти записи обозначают одну и ту же дату: 1 января 2009 года.

Время представляется таким форматом:

20:15 – 20 часов 15 мин

15:18:47 – 15 часов 18 мин 47 сек

Здесь часы, минуты и секунды отделяются двоеточиями.

Ввод последовательных рядов чисел

Последовательный ряд чисел – это числа, отличающиеся между собой на единицу или другое постоянное число, называемое *шагом*.

Для облегчения ввода таких рядов чисел в MS Excel используется прием называемый *распространением* или *автозаполнением*. Существует два приема создания последовательного ряда чисел.

Создание ряда чисел с шагом 1.

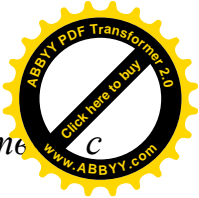
Для реализации этой операции необходимо выполнить следующую последовательность действий:

- ввести первое число ряда в первую ячейку,
- установить указатель мыши на маркер заполнения, выделенной первой ячейки (квадратик в нижнем правом углу) так, чтобы указатель мыши принял форму крестика,
- нажать клавишу *Ctrl*, что вызывает превращение крестика в двойной крестик,
- при нажатой клавиши *Ctrl* и левой клавиши мыши перемещать мышь до последней ячейки,
- отпустить левую клавишу мыши,
- отпустить клавишу *Ctrl*.
- снять выделение интервала.

Создание ряда чисел с шагом отличным от 1.

- ввести первое число ряда в первую ячейку,
- ввести второе число ряда во вторую ячейку,
- выделить эти ячейки,
- установить курсор мыши на маркер заполнения второй из выделенных ячеек так, чтобы появился крестик,
- при нажатой левой клавиши мыши перемещать мышь до последней ячейки интервала,
- отпустить левую клавишу мыши,
- снять выделение интервала.

Эти операции позволяют создать ряд чисел в *арифметической прогрессии*. Такие ряды можно создавать и в *геометрической прогрессии*. Для этого



необходимо ввести первое число в ячейку и выполнить следующие действия с использованием меню:

СЕРВИС – ЗАПОЛНИТЬ – ПРОГРЕССИЯ

В открывшемся дополнительном окне выбрать вид прогрессии, шаг, расположение (по строкам, по столбцам), предельное значение, затем нажать кнопку **ОК** этого окна.

ЛЕКЦИЯ № 13

13.1. Редактирование данных

В MS Excel вносить изменения можно как в отдельно взятой ячейке, так и в группе ячеек.


Редактировать содержимое ячеек можно двумя способами:

- непосредственно в ячейке,
- в строке формул.

Для внесения изменений непосредственно в ячейке необходимо:

- дважды щелкнуть по редактируемой ячейке или после ее выделения нажать клавишу *F2*;
- внести изменения;
- нажать клавишу *Enter*.

При редактировании содержимого ячейки в строке формул следует:

- выделить нужную ячейку, после этого ее содержимое отобразится в строке формул;
- щелкнуть по содержимому ячейки в строке формул, что приведет к появлению текстового курсора, и выполнить необходимые изменения
- щелкнуть по кнопке ввод  в строке формул или нажать клавишу *Enter*.

Копирование, перемещение и удаление данных

Копировать можно содержимое одной ячейки, строки, столбца, группы ячеек.

При этом копируемый диапазон должен быть *связанным*. Копирование возможно как в пределах одного листа, книги, так и на другой лист, книгу. Имеется два способа копирования:

- с помощью буфера обмена,
- путем буксировки.

Перед копированием следует убедиться, что та область, где будет создаваться копия, свободна от данных. В противном случае, имеющиеся в этой области данные, будут замещены, создаваемой копией. Это приведет к потере этих данных.

Копирование с помощью буфера обмена

Последовательность операций при копировании с помощью буфера обмена следующая:

- выделить ячейку или группу ячеек, содержимое которых подлежит копированию;
- использовать один из вариантов образования копии в буфере обмена:
 - нажать клавиши *Ctrl + C*,



- щелкнуть по кнопке **КОПИРОВАНИЕ** на панели инструментов,
- выполнить команду **ПРАВКА – КОПИРОВАТЬ**.

После выполнения одной из этих операций вокруг выделенных ячеек появится *бегущий пунктир*, а содержимое ячейки (ячеек) скопируется в буфер обмена.

- выделить ячейки, куда должна помещаться копия,
- выполнить одну из команд:
 - нажать клавиши **Ctrl + V**,
 - щелкнуть по кнопке **ВСТАВИТЬ** на панели инструментов,
 - выполнить операции меню **ПРАВКА – ВСТАВИТЬ**.

Перемещение выполняется аналогичным образом с тем лишь отличием, что операция **КОПИРОВАТЬ** заменяется операцией **ВЫРЕЗАТЬ** одним из способов:

- щелкнуть по кнопке **ВЫРЕЗАТЬ** на панели инструментов,
- выполнить операции меню **ПРАВКА – ВЫРЕЗАТЬ**.

Удаление выполняется в следующей последовательности:

- выделить ячейку или группу ячеек,
- выполнить одну из операций:
 - щелкнуть по кнопке **УДАЛИТЬ** на панели инструментов,
 - нажать комбинации клавиш **Ctrl + X**,
 - выполнить последовательность команд меню **ПРАВКА – УДАЛИТЬ**.

Перемещение и копирование с помощью приема буксировки

Перемещение и **копирование** можно производить с помощью еще одного приема, называемого **буксировкой**.

Перемещение осуществляется следующей последовательностью операций:

- выделить ячейку или группу ячеек,
- установить указатель мыши на границу выделенного участка так, чтобы указатель мыши принял вид двойной стрелки ↖ ,
- при нажатой левой клавише мыши переместить указатель мыши на новое место,
- отпустить левую клавишу мыши.

Копирование выполняется аналогично **перемещению**, но при нажатой клавише **Ctrl**. После завершения операции копирования *сначала отпускается клавиша мыши, а затем клавиша Ctrl, а не наоборот*. Если клавиши отпускать в обратной последовательности, то произойдет не *создание ряда чисел, а копирование содержимого выделенной ячейки (ячеек)*.

13.2.Операции с данными

Сортировка

Под **сортировкой** понимают изменение порядка следования строк таблицы с целью их упорядочения по какому-либо признаку. Например, по *убыванию* или *возрастанию* значений данных. Для выполнения простейшей сортировки



достаточно выделить фрагмент таблицы, подлежащий сортировке, и щелкнуть панели инструментов по кнопке **СОРТИРОВКА ПО ВОЗРАСТАНИЮ** или **СОРТИРОВКА ПО УБЫВАНИЮ**. Для выполнения сортировки в общем случае следует выполнить последовательность команд **ДАННЫЕ – СОРТИРОВКА**, а затем в раскрывшемся диалоговом окне сделать необходимые установки.

Автосуммирование

Суммирование чисел, расположенных в заданном диапазоне ячеек, является одной из самых распространенных операций в таблицах. В MS Excel предусмотрено два способа автоматизации этой операции:

- с использованием строки состояния,
- с использованием кнопки **АВТОСУММА**.

Способ с использованием *строки состояния* относится к наиболее простым. Для автоматизации операции суммирования таким способом необходимо выполнить предварительную подготовку:

- щелкнуть правой кнопкой мыши по строке состояния таблицы,
- в появившемся контекстном меню, содержащем пункты
 - Нет
 - Среднее
 - Количество значений
 - Количество чисел
 - Максимум
 - Минимум
 - Сумма

установить флажок перед пунктом Сумма

- щелкнуть за пределами контекстного меню (убрать меню с экрана).

После этого, при выделении любого интервала ячеек, содержащих числа, в строке состояния будет автоматически отображаться их сумма. Разумеется, что, установив флажок перед одним из других пунктов контекстного меню, можно автоматизировать и другие операции. Пункт «Нет» позволяет исключить автоматическое выполнение каких-либо операций над числами из выделенного интервала.

Второй способ заключается в использовании кнопки **Атосумма** панели **Стандартная**. Для его применения необходимо:

- выделить ячейку, куда следует помещать результат суммирования. Обычно ее располагают под столбцом выделенных чисел или справа выделенной строки.
- Щелкнуть по кнопке **Автосумма**. Программа попытается самостоятельно выделить диапазон ячеек для суммирования, обозначив его бегущим пунктиром.
- Выделить требуемый интервал ячеек, (в случае несогласия с «предложением» программы).
- Щелкнуть по кнопке **Автосумма**.

После этих операций в выделенной под результат ячейке отобразится сумма чисел.



13.3. Работа с формулами

В ячейках таблицы могут размещаться не только числа и текст, но и *формулы*.

Формула представляет собой выражение, состоящее из чисел, знаков, математических операций, адресов ячеек, названий функций, круглых скобок и т. д.

Отличительным признаком того, что в таблице MS Excel представлена формула, а не какой-либо текст, является *присутствие перед ней знака равенства* (=), знака плюс (+) или знака минус (-).

Запрещается:

- использовать скобки для отрицательного числа,
- вводить пробел для разделения тысяч,
- вводить денежные обозначения (р.),
- использовать кириллицу (русский алфавит) для обозначения адресов ячеек.

Порядок создания формул

Для создания формулы необходимо:

- выделить ячейку, в которой требуется поместить формулу,
- в строке формул установить знак =,
- после знака = в строке формул ввести текст формулы,
- щелкнуть по значку **ВВОД** в строке формул или нажать клавишу **Enter**.

После этих операций формула отобразится в выделенной ячейке. Например, пусть требуется сложить числа 29 и 278, а результат вывести в ячейку A2. Для этого необходимо выделить ячейку A2 и в нее ввести формулу = 29 + 278. Одновременно эта формула отобразится в строке формул. После выделения ячейки A2 эту формулу можно ввести и в строке формул, и тогда она отобразится в выделенной ячейке.

13.4. Использование в формулах адресов ячеек

Часто в формулах используются не значения, содержащиеся в ячейках, а адреса этих ячеек. Адреса ячеек принято называть *ссылками* на ячейки. Например, = A1 + B3. При такой записи будут взяты числа из ячейки A1 и B3 и выполнится вычисление их суммы. Кроме этого существует еще два вида ссылок: *внешние и удаленные*.

Внешние ссылки

Разрешается ссылаться не только на ячейки, расположенные на текущем листе, но и на ячейки других листов рабочей книги. Такие ссылки называются *внешними*. Синтаксис таких ссылок виден из примера:

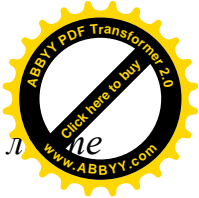
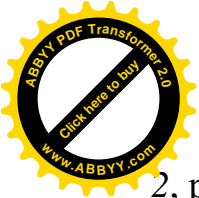
$$= \text{Лист2! B17} + B1$$

В этом примере одно число берется из ячейки B17, расположенной на листе 2, а другое число – из ячейки B1, расположенной на текущем листе.

Удаленные ссылки

Ссылки на данные других книг называют *удаленными*. Формулы с использованием удаленных ссылок имеют вид:

$$= "c:\EAPU\T. XLS] \text{Лист2!}" C3 * A1 + B3$$



Здесь используется удаленная ссылка на ячейку *C3*, расположенную на листе 2, рабочей книги *T. XLS*, путь к которой указан как *c:\EAPU*.

Символы *!* и *[]* устанавливаются автоматически. Редактирование формул выполняется таким же образом, что и редактирование содержимого ячеек.

По умолчанию программа MS Excel обеспечивает вывод на экран не самой формулы, а результат вычисления с помощью этой формулы. Вместе с тем имеется способ отображения на экране всех формул таблицы. Для этого необходимо выполнить следующие операции:

СЕРВИС – ПАРАМЕТРЫ – ВИД – ПАРАМЕТРЫ ОКНА – ФОРМУЛЫ

В строке **ФОРМУЛЫ** установить флажок. После этого в ячейках будут отображаться формулы, а сами ячейки будут увеличиваться до размеров, позволяющих помещать в них формулы.

Операции копирования, перемещения или распространения формул осуществляется так же, как это делается для ячеек.

ЛЕКЦИЯ №14

14. ЭЛЕКТРОННАЯ ТАБЛИЦА MS EXCEL. ФУНКЦИИ

14.1. Операции и функции

Программа MS Excel допускает использование в формулах знаков математических и логических операций при работе с числами, а также операции конкатенации, объединяющие текстовые строки, при работе с текстом. Если в одном выражении используется несколько различных операций, то последовательность их выполнения осуществляется в соответствии с установленными приоритетами (старшинством).

В формулах могут использоваться различные функции. В MS Excel имеется библиотека, содержащая более 200 всевозможных функций. По своему назначению их можно разделить на:

- математические,
- статистические,
- финансовые,
- для работы с базами данных и др.

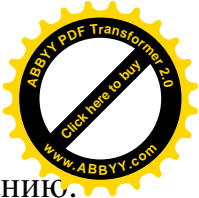
Синтаксис записи функции в MS Excel следующий:

| |
|--|
| <i>Имя функции (список аргументов)</i> |
|--|

Список аргументов представляет собой перечень чисел, адресов ячеек, других функций, разделенных точкой с запятой.

Например,

СУММ(A1; A2; A3)



Здесь *СУММ* – имя функции, осуществляющей суммирование чисел.

A1, A2, A3 – адреса ячеек, содержащих числа, подлежащие суммированию.

Для более компактной записи списка аргументов MS Excel позволяет вместо перечня адресов, *рядом расположенных* ячеек, указывать их *диапазон*. Такой диапазон представляется *адресом первой его ячейки и отделенного от него двоеточием адресом последней ячейки*. Например, *СУММ(A1:A8)*. При такой записи выполнится сложение чисел, расположенных в соседних ячейках интервала *A1 – A8*. Правильной была бы и такая запись:

СУММ(A1; A2; A3; A4; A5; A6; A7; A8),

но совершенно очевидно, что она более громоздкая, чем запись

СУММ(A1: A8).

Допускается представлять список аргументов из нескольких различных диапазонов ячеек. Например,

СУММ(A1:A6; B5:C12).

Обобщая изложенное выше, следует использовать следующие правила при записи списка аргументов функций:

- *аргументы функций должны заключаться в круглые скобки. При этом нельзя ставить пробелы ни после открывающейся скобки, ни перед закрывающейся скобкой;*
- *в качестве аргументов можно использовать числа, логические значения, массивы, текст, ссылки и другие формулы;*
- *в качестве разделителей аргументов функции в русифицированной версии MS Excel используется знак (;) – точка с запятой.*

Если в качестве аргумента одной функции используется другая функция, то она называется *вложенной*. MS Excel допускает не более *семи уровней вложения*. Пример использования вложенной функции:

= СТЕПЕНЬ (СУММ(A4;C10);2)

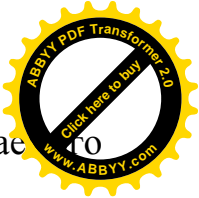
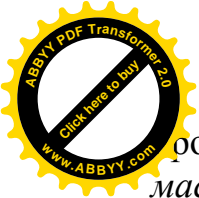
В этом примере используется стандартная функция СТЕПЕНЬ, в которой в качестве первого аргумента записывается основание степени, а второго – показатель степени. Здесь основанием степени предполагается число, вычисляемое с помощью другой (вложенной) функции СУММ, как результат сложения чисел, расположенных в ячейках A4 и C10. Показатель степени – число 2.

При создании текста функций адреса ячеек можно:


- *вводить, набирая с клавиатуры,*
- *щелкая мышью по ячейке, адрес которой должен отображаться в списке аргументов на месте текстового курсора.*

Очевидно, что последний способ в большинстве случаев предпочтительней, так как при щелчке по нужной ячейке *ее адрес автоматически отобразится* в списке аргументов функции.

Для посимвольного отображения в формуле какой-либо функции необходимо точно знать ее имя, состав списка аргументов и последовательность их записи. Это требует определенного умственного напряжения. Для облегчения этого процесса



Программа MS Excel предлагает своеобразного помощника, называемого **мастером вставки функций**. Мастер вставки функций вызывается:

- щелчком мыши по кнопке  **Вставка функций**, расположенной на панели инструментов;
- щелчком по кнопке с таким же названием в строке формул;
- последовательностью команд основного меню **Вставка – Функция**.

Работа мастера состоит из двух шагов. Для каждого шага используется отдельное окно.

В диалоговом окне *первого шага* с названием **Мастер функций – шаг 1 из 2**, в списке **Категория**, выбирается нужный вид функций, а в раскрывшемся дополнительном окне выбирается имя требуемой для вычисления функции и производится щелчок по кнопке **ОК** окна **Мастер функций – шаг 1 из 2**.

В окне *второго шага* с названием **Мастер функций – шаг 2 из 2** в окне **Аргументы функции** создается список аргументов.

Из описанного выше можно представить обобщенный порядок включения функции в формулу:

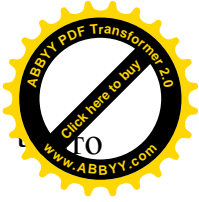
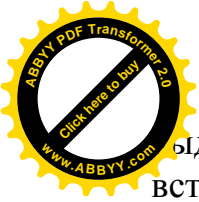
- выделить ячейку, в которой должна размещаться формула;
- установить в ней знак равно (=);
- начать посимвольно создавать формулу до того места, где надо включить функцию. Если формула не начинается с функции, то этот пункт исключается;
- выполнить одну из операций:
 - щелкнуть мышью по кнопке f_x строки формул;
 - щелкнуть мышью по кнопке f_x панели инструментов;
 - выполнить последовательность команд основного меню **ВСТАВКА – ФУНКЦИЯ**;
- в образовавшемся окне **Мастер функций – шаг 1 из 2** выбрать название требуемой функции;
- в появившемся окне **Аргументы функции** указать ее аргументы;
- продолжить ввод других элементов формулы в обычном порядке, если они предусмотрены;
- для завершения ввода формулы щелкнуть по кнопке **Ввод строки формул** или нажать клавишу **Enter**.

Если функцию необходимо вставить в уже существующую формулу, то вначале следует:

- выделить ячейку с формулой (текст формулы отобразится в строке формул);
- щелкнуть в строке формул в любом месте, создаваемой формулы (появится текстовый курсор);
- переместить текстовый курсор к месту вставки функции;
- вызвав мастера вставки функций, произвести ее вставку по уже описанной схеме.

Ошибки в формулах

При работе с формулами могут возникнуть ошибки, связанные с использованием пустых адресов или удаленных ячеек, а также неправильным вводом аргументов и функций. В каждом из этих случаев программа MS Excel



ыдает сообщения об ошибке. В табл. 14.1 приведены наиболее встречающиеся ошибки и их описание.

Таблица 14.1. Сообщения об ошибках в формулах

| Вид сообщения | Смысл ошибки |
|----------------------|---|
| # ДЕЛ/0! | Попытка деления на ноль. |
| # Н/Д! | Отсутствие данных (возможно ячейка пуста). |
| # ИМЯ! | Ссылка на несуществующее имя. |
| # ЧИСЛО! | Использован недопустимый числовой аргумент. |
| # ССЫЛКА! | Неправильно указан адрес ячейки. |
| # ЗНАЧ! | Тип значения не совпадает с типом данных, допустимых для данного аргумента. |

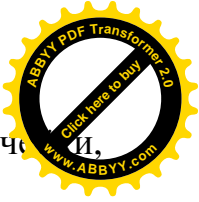
Автопересчет

Если в ячейках, адреса которых указаны в формулах, вносить изменения, то программа MS Excel *автоматически* произведет перерасчет по формуле с учетом новых значений. В тех случаях, когда таблица велика, то при ее редактировании каждый раз будет включаться перерасчет, что займет достаточно много времени. Поэтому, с целью сокращения времени отладки такой таблицы, на время ее отладки автоматический перерасчет можно исключить. Это можно сделать, выполнив следующую последовательность операций с пунктами основного меню: **СЕРВИС – ПАРАМЕТРЫ**. На вкладке **ВЫЧИСЛЕНИЯ** окна **ПАРАМЕТРЫ** необходимо включить переключатель **ВРУЧНУЮ**. После этого перерасчет значений возможен только при нажатии клавиши **F9**.

14.2. Относительная и абсолютная адресация ячеек

Часто приходится многократно выполнять расчеты по одной и той же формуле, но при различных значениях аргументов. Например, вычисление функции вида $y = ax^2$ для ряда значений аргумента x . Если подходить формально, следуя логике вычислений по формулам принятой в MS Excel, то в каждой ячейке, куда будет помещаться результат вычисления, необходимо создавать формулу с указанием адреса ячейки, содержащей значение x и адреса ячейки, содержащей значение коэффициента a . Такой подход возможен, но он требует больших затрат времени и энергии.

Программа MS Excel предлагает и другой способ решения таких проблем. Он состоит в том, что *создается единая формула вычисления в одной ячейке и затем она копируется во все остальные ячейки*. При этом *автоматически* изменяется адрес ячейки, в которых размещается значение аргумента. Применительно к приведенному примеру *автоматически* должен меняться адрес ячейки, хранящей



чередное значение аргумента x и оставаться неизменным адрес ячейки, содержащей значение коэффициента a .

Для реализации такого способа вычислений по формулам в MS Excel предусмотрены два вида адресации:

- *относительная,*
- *абсолютная.*

При *относительной адресации* адрес ячейки, содержащей значение аргумента, при перемещении копии формулы в другую ячейку, *автоматически* изменяется *относительно* адреса ячейки, содержащей значение аргумента в оригинале формулы. Например, пусть значения аргумента x в приведенной формуле размещены в ячейках $A1:A10$, значение коэффициента a – в ячейке $C1$. В ячейке $B1$ создана формула

$$= C1*A1*A1$$

Предполагается все последующие вычисляемые значения функции y по приведенной формуле последовательно помещать в ячейках интервала $B2:B10$. Следовательно, копии формулы следует поочередно создавать во всех ячейках интервала $B2:B10$. Если копия формулы помещается в ячейку $B2$, то адрес коэффициента $C1$ должен остаться неизменным, а адрес аргумента должен соответственно измениться. Копия должна принять следующий вид $= C1*A2*A2$. В ячейке $B10$ формула принимает вид $= C1*A10*A10$. Как видно, здесь аргумент синхронно меняет свой адрес *относительно* своего адреса, указанного в оригинале формулы. В то же время адрес коэффициента $C1$ должен оставаться неизменным, то есть *абсолютным*. Иными словами новый адрес аргумента меняется на столько, на сколько ячейка с копией формулы отдалена от исходной по вертикали и по горизонтали.

Бывают случаи, когда при перемещении копии формулы адреса некоторых операндов должны оставаться неизменными. В приведенном примере это адрес ячейки, где хранится значение коэффициента a . Для такого операнда устанавливается другая адресация – *абсолютная*.

Признаком абсолютной адресации является наличие знака \$ перед значением координаты в адресе ячейки. Если в формуле адрес ячейки представлен как $\$A2$, то это означает, что адрес столбца A является *абсолютным* и остается неизменным, а адрес строки воспринимается как *относительный* и может изменяться. В случае обозначения адреса в виде $\$A\5 – адрес ячейки $A5$ воспринимается как *абсолютный* и остается неизменным во всех случаях. Адресация вида $A\$9$ говорит о том, что адрес столбца *относительный*, а адрес строки *абсолютный*. Сочетание адресации может быть *различной*.

Абсолютная адресация чаще всего используется тогда, когда в формулах используются константы, размещенные в определенных ячейках или данные, размещенные в отдельных столбцах или строках.

Порядок создания абсолютного адреса следующий:

- *выделить в формуле адрес, подлежащий редактированию, таким же способом, как выделяется фрагмент текста в текстовом редакторе MS Word, то есть установить указатель мыши на первый символ адреса и при нажатой ее левой клавиши перемещать мышью до последнего символа*



выделяемого адреса, после чего отпустить клавишу (выделенный фрагмент окрасится темным цветом);

- нажать на клавиатуре клавишу $F4$, после чего адрес примет вид $\$B\6 , то есть будет установлена абсолютная адресация и для столбца B и для строки 6 . Вторичное нажатие клавиши $F4$ приведет к изменению адреса к виду $B\$6$, последующее ее нажатие – к виду $\$B6$. Нажимая таким образом клавишу $F4$, можно устанавливать желаемую абсолютную адресацию.

ЛЕКЦИЯ №15

15. ПОСТРОЕНИЕ ДИАГРАММ В СРЕДЕ EXCEL

Программа MS Excel позволяет по числовым данным строить различные графики и диаграммы. Графическое представление данных создает наглядность и существенно облегчает их анализ. В MS Excel могут строиться диаграммы следующего вида:

- линейные,
- столбчатые,
- круговые,
- гистограммы,
- объемные и др.

В одной диаграмме можно совместить несколько графических изображений, построенных по нескольким зависимостям, размещенным в нескольких строках или столбцах. Такие зависимости принято называть *рядами*. Каждый ряд формирует отдельную линию на диаграмме. Основными элементами, с помощью которых строятся ряды, являются *маркеры*. В зависимости от типа диаграммы используются различного вида *маркеры*:

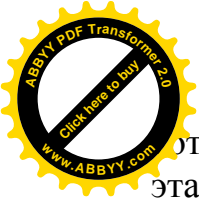
- точки,
- линии,
- полосы,
- столбцы,
- секторы.

Все диаграммы, за исключением *круговой* и *объемной*, строятся на плоскости, в прямоугольной системе координат. Горизонтальную ось принято называть *осью категорий*, а вертикальную ось – *осью значений*. В объемных диаграммах используется третья ось – *ось рядов*.

Включаемые в документ диаграммы, по желанию пользователя, могут сопровождаться *заголовками*, *разметкой* и *названием осей*, *координатной сеткой* и так называемой *легендой*, представляющей собой список использованных в диаграмме условных обозначений.

Общий порядок построения диаграмм

Для облегчения процесса построения диаграмм в MS Excel предусмотрен *мастер построения диаграмм*, который «руководит» этим процессом, предлагая



от или иной ход действий. В работе мастера диаграмм предусматривается чье-либо действие на этапе (*шага*). Для построения диаграммы по числовым данным, размещенным в строках или столбцах таблицы, следует:

- выделить строки или столбцы таблицы с данными, на основании которых предполагается построение диаграммы. В этом случае данные, располагаемые в ячейках, должны составлять прямоугольные блоки. При необходимости, в выделяемый фрагмент можно включать название строк (*категорий*) или столбцов (*значений*);
- щелкнуть по кнопке **Мастер диаграмм** на панели инструментов или выбрать из меню команду **Вставка – Диаграмма**.

На экране появляется диалоговое окно *первого шага* мастера диаграмм. Это окно имеет две вкладки: **Стандартные**, **Нестандартные**.

На вкладке **Стандартные** в окне **Тип** приводятся названия 14 стандартных типов диаграмм (гистограмма, линейная, график, круговая и др.).

На вкладке **Нестандартные** в окне **Тип** приводятся названия 20 нестандартных типов диаграмм, которые могут быть построены мастером.

После выбора подходящего типа диаграммы на каждой из этих вкладок, справа от окна **Тип**, раскрывается окно **Вид**, в котором отображаются образцы диаграмм, выбранного типа.

Чтобы посмотреть, как выглядит диаграмма, построенная по конкретным данным, следует щелкнуть мышью по кнопке **Просмотр результата** окна **Мастер диаграмм [шаг 1 из 4]**. Если представленный вид диаграммы устраивает пользователя, то можно завершить процесс построения щелчком мыши по кнопке окна мастера **Готово**. Если же требуется добавить некоторые детали оформления, то следует щелкнуть мышью по кнопке **Далее** и перейти ко *второму шагу* мастера диаграмм.

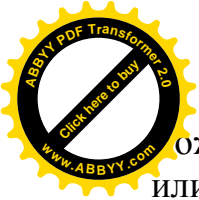
В раскрывшемся окне *второго шага*, **Мастер диаграмм [шаг 2 из 4]**, на вкладке **Диапазон данных** изображается примерный вид диаграммы, построенной по типу, выбранному на первом шаге. В поле ввода **Диапазон**: можно уточнить или изменить диапазон ячеек, содержащих данные для построения диаграммы.

Программа автоматически формирует ряды данных по строкам или по столбцам, исходя из того, что на диаграмме должно быть меньше рядов данных, чем чисел в каждом ряду. Если в выделенной области столбцов больше, чем строк, то в качестве *рядов* по умолчанию выбираются *строки*, в противном случае – *столбцы*. Если пользователя не устраивает такое предложение, то он может не согласиться, и с помощью переключателя **Ряды в:** в этом же окне установить свой вариант (*в строках* или *в столбцах*).

На вкладке **Ряд** этого окна размещены элементы управления строящейся диаграммы. С помощью кнопок **Добавить** и **Удалить** можно изменить первоначально выделенное количество рядов на диаграмме.

Для удаления достаточно выделить в списке **Ряд** этой вкладки строку соответствующую удаляемому ряду диаграммы и щелкнуть по кнопке **Удалить**.

Для добавления ряда вначале следует щелкнуть по кнопке **Добавить**, а затем ввести в поля **Имя**, **Значения**, **Подписи** оси **X** адреса или диапазоны адресов ячеек таблицы, содержащие соответственно *название ряда*, *числа ряда* и *подписи* в разметке оси категорий. С помощью кнопок в правом конце каждого из этих полей



можно переходить к исходной таблице и в ней выделять соответствующие ячейки или диапазоны. Для перехода к *третьему шагу* мастера диаграмм следует щелкнуть по кнопке **Далее**. После этого на экране появляется окно *третьего шага*, **Мастер диаграмм [шаг 3 из 4]**.

Окно *третьего шага* содержит шесть вкладок: **Подписи данных**, **Таблица данных**, **Заголовки**, **Оси**, **Линии сетки**, **Легенда**.

На вкладке **Подписи** данных можно выбрать способ указания на диаграмме чисел, соответствующих маркерам рядов.

На вкладке **Таблица** данных можно включить флажок вывода на экран, как самой диаграммы, так и таблицы данных, по которым диаграмма строится.

На вкладке **Заголовки** расположены поля, в которые можно помещать название диаграммы и осей.

На вкладке **Оси** имеются флажки, с помощью которых можно включать или выключать режимы разметки осей *категорий* и *значений*.

На вкладке **Линии сетки** находятся флажки, управляющие режимом вывода координатной сетки отдельно по каждой из осей диаграммы.

На вкладке **Легенда** находится флажок и переключатели, управляющие выводом и расположением легенды относительно диаграммы. Завершается работа мастера диаграмм на третьем шаге щелчком по кнопке **Далее**. Это вызывает появление окна **Мастер диаграмм [шаг 4 из 4]**.

На *четвертом шаге* мастера диаграмм, с помощью переключателя, выбирается место расположения диаграммы (на *отдельном* или *имеющемся* листе). Завершается работа с мастером диаграмм щелчком по кнопке **Готово**.

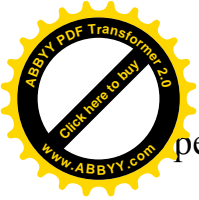
Если диаграмма размещена на текущем листе, то ее размеры можно изменять, используя *маркеры изменения размеров*, перемещать, используя стандартные приемы.

Диаграмму можно редактировать. Для этого необходимо по ней сделать двойной щелчок и, используя контекстное меню или основное меню, обратиться к тем или иным окнам мастера.

Тренды.

Тренд – это функция заданного вида с помощью, которой можно аппроксимировать построение по данному графику. Тренд служит для выявления тенденции развития процесса, представленного в виде диаграммы и обеспечивает прогноз на заданный период. В Excel предусмотрены стандартные типы:

- линейный
- логарифмический
- степенной
- экспоненциальный
- полиномиальный
- скользящее среднее



Тренд можно строить для диаграмм типа:

- линейный график
- гистограмма
- диаграмма с областями
- точечная диаграмма

Для построения линии тренда нужно активировать график и правой кнопкой мыши вызвать контекстное меню, в котором выполнить команду – добавить линию тренда.

ЛЕКЦИЯ №16

16. РЕШЕНИЕ МАТЕМАТИЧЕСКИХ ЗАДАЧ СРЕДСТВАМИ MS EXCEL

16.1. Операции с матрицами

Средства MS Excel могут быть весьма полезны для решения задач линейной алгебры, прежде всего для операций с матрицами и решения систем линейных уравнений.

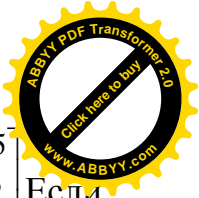
Значительная часть математических моделей различных объектов, в частности автоматизированного электропривода, электрических сетей, записывается в достаточно простой и компактной матричной форме. В связи с этим возникает необходимость совершать различные операции с ними.

К операциям с матрицами относятся:

- *транспонирование,*
- *вычисление определителя,*
- *нахождение обратной матрицы,*
- *сложение и вычитание,*
- *умножение.*

16.1.1. Транспонирование матриц

Под транспонированием матрицы понимают замену столбцов матрицы ее соответствующими строками. Это означает, что первая строка исходной матрицы становится первым столбцом транспонированной матрицы и т. д. Например, для исходной матрицы



$$A = \begin{bmatrix} 3 & -7 & 11 \\ 18 & 13 & 39 \\ -5 & 92 & 86 \end{bmatrix} \quad \text{транспонированной является} \quad A^T = \begin{bmatrix} 3 & 18 & -5 \\ -7 & 13 & 92 \\ 11 & 39 & 86 \end{bmatrix} \quad \text{Если}$$

исходная матрица A имеет размер $m \times n$, то транспонированная матрица A^T будет иметь размер $n \times m$.

Для выполнения операции транспонирования в MS Excel используется функция ТРАНСП, которая позволяет менять ориентацию массива (вектора чисел) на рабочем листе с горизонтальной на вертикальную и наоборот.

Порядок действий при транспонировании матрицы следующий:

выделить соответствующего размера блок ячеек под транспонированную матрицу,

- нажать на панели инструментов кнопку (f_x) **Вставка функции**,
- в раскрывшемся окне **Мастер функций** в поле **Категория** выбрать строку **Ссылки и массивы**,
- в другом поле **Функция** выбрать имя функции **ТРАНСП**,
- щелкнуть по кнопке **ОК** окна **Мастер функций**,
- в появившемся диалоговом окне **ТРАНСП** в строке **Массив** ввести диапазон ячеек, занимаемой исходной матрицей. Если диалоговое окно закрывает исходную матрицу, его можно перетащить в сторону.
- нажать сочетание клавиш **Ctrl + Shift + Enter**.

В результате этих действий, в выделенном ранее диапазоне ячеек под транспонированную матрицу, появятся ее значения.

16.1.2. Вычисление определителя матрицы

Важной характеристикой квадратных матриц является их определитель. **Определитель матрицы** – это число, вычисляемое на основе ее элементов. Обычно определитель обозначают знаком Δ с индексом, определяющим порядок определителя. Существуют правила вычисления определителей, но вычисление их вручную представляет существенные трудности. Это объясняется тем, что с ростом порядка n матрицы (ее размеров) существенно увеличивается количество вычисляемых членов определителя ($n!$).

Программа MS Excel позволяет производить вычисление определителей квадратной матрицы. Для этого необходимо выполнить следующие операции:

- установить табличный курсор в ячейку, в которую должно помещаться значение вычисленного определителя,
- щелкнуть по кнопке **Вставка функции**,
- в раскрывшемся диалоговом окне **Мастер функций**, в рабочем поле **Категория** выбрать строку **Математические** и щелкнуть по ней,
- в поле **Функция** щелкнуть по строке **МОПРЕД**,
- щелкнуть по кнопке **ОК** окна **Мастер функций**,
- в строке **Массив**, раскрывшегося диалогового окна **МОПРЕД**, ввести диапазон ячеек, в которых размещена исходная матрица,
- щелкнуть по кнопке **ОК**.

После этих действий в ранее выбранной ячейке появится значение определителя.



16.1.3. Нахождение обратной матрицы

В математике для каждого числа a , не равного нулю, существует ему *обратное число* a^{-1} . Аналогично этому для *квадратных* матриц вводится понятие – *обратная матрица*. Обратные матрицы широко используются для решения систем уравнений. Для матрицы A ей обратная обозначается как A^{-1} . Матрица A^{-1} называется обратной по отношению к квадратной матрице A , если при ее умножении на исходную матрицу *слева* или *справа* получается *единичная матрица*. То есть $A \times A^{-1} = A^{-1} \times A = E$.

Обратная матрица является квадратной матрицей того же порядка, что и исходная. Обратную матрицу можно получить только тогда, когда *определитель исходной матрицы не равен нулю* (исходная матрица *невырожденная*). Для вырожденной матрицы обратная матрица не существует.

Для получения обратной матрицы средствами программы MS Excel необходимо выполнить следующие действия:

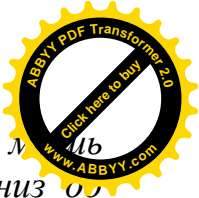
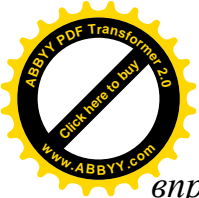
- выделить область ячеек для обратной матрицы. Размер области такой же как и у исходной матрицы;
- щелкнуть на панели инструментов по кнопке **Вставка функции**,
- в раскрывшемся окне **Мастер функций**, в поле **Категория**, выбрать строку **Математические**, а в поле **Функция** – строку **МОБР**,
- щелкнуть по кнопке **ОК**,
- в появившемся окне **МОБР**, в поле **Массив**, ввести диапазон ячеек, занимаемых исходной матрицей,
- нажать комбинацию клавиш **Ctrl + Shift + Enter**.

После этих операций в заранее выбранной области ячеек отобразится обратная матрица.

16.1.4. Сложение и вычитание матриц

Операции сложения и вычитания матриц могут производиться *только с матрицами одного размера*. При этом результирующая матрица будет такого же размера, как и исходные матрицы. Действия для выполнения операций сложения и вычитания матриц – схожие и представляются следующей последовательностью:

- установить *табличный курсор* в ячейку, где будет размещаться *первый элемент результирующей матрицы* (ее верхний левый угол), например, $P1$,
- в выбранную ячейку ввести формулу вычисления первого элемента результирующей матрицы. При сложении матриц такой формулой будет $=A1+F1$ (при вычитании $=A1-F1$). Здесь $A1$ и $F1$ – это адреса, взятых в качестве примера, первых элементов, участвующих в операции матриц,
- скопировать формулу в остальные ячейки результирующей матрицы. Для этого необходимо:
 - выделить ячейку, где записана формула,
 - установить курсор мыши на маркер заполнения этой ячейки (затемненный квадратик в правом нижнем углу ячейки) таким образом, чтобы указатель мыши принял вид тонкого крестика,



– нажать левую клавишу мыши и, не отпуская ее, перемещать мышь вправо по ячейкам первой строки результирующей матрицы, а затем вниз до правого нижнего угла этой матрицы.

В результате этих действий в ячейках результирующей матрицы появятся ее значения, равные сумме или разности соответствующих элементов, участвующих в операции матриц.

16.1.5. Умножение матрицы на число

Для умножения матрицы на число используются операции аналогичные сложению или вычитанию матриц. Для этого в ячейку, выделенную для первого элемента результирующей матрицы, записывается формула умножения первого элемента исходной матрицы на число. В этой формуле записывается адрес первого элемента, а в качестве числа записывается или его значение или его адрес (если число помещено в ячейку). Далее используется способ копирования формулы во все последующие ячейки результирующей матрицы.

16.1.6. Умножение матриц

Произведение двух матриц возможно в случае, когда количество столбцов первой матрицы равно количеству строк второй матрицы. Например: $C = A \times B$, где

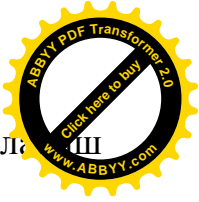
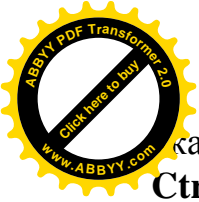
$$A = \begin{bmatrix} 1 & 3 & 4 & 2 \\ 3 & 2 & 0 & -1 \\ 0 & 1 & -1 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & 3 \\ 2 & 2 \\ 10 & 0 \\ 12 & -1 \end{bmatrix}$$

Здесь четыре столбца матрицы A равны четырём строкам матрицы B .

Для умножения матриц необходимо выполнить следующие операции:

- выделить область ячеек под результирующую матрицу. Для этого необходимо установить размер этой матрицы. Он определяется как $m \times p$, где m – количество строк матрицы A , а p – количество столбцов матрицы B , что соответствует 3×2 .
- Щелкнуть по кнопке **Вставка функции**,
- в раскрывшемся окне **Мастер функций** в поле **Категория** выбрать строку **Математические**, а в поле **Функция** – имя функции **МУМНОЖ**,
- щелкнуть по кнопке **ОК**,
- в появившемся диалоговом окне **МУМНОЖ** в строке **Массив1** указать диапазон ячеек, занимаемых матрицей A , а в строке **Массив2** – диапазон ячеек матрицы B ,
- нажать сочетание клавиш **Ctrl+Shift+Enter**.

В результате этих действий в ячейках, выделенных под результирующую матрицу, появятся ее значения. Если этого не произойдет, то следует щелкнуть



казателем мыши в строке формул и еще раз нажать комбинацию клавиш **Ctrl+Shift+Enter**.

ЛЕКЦИЯ №17

17. РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ УРАВНЕНИЙ

Большое количество прикладных задач в технике, и в электротехнике в частности, решаются с использованием систем линейных уравнений. Поэтому весьма важно владеть методикой их решения. Существует достаточное количество как ручных способов их решения, так и способов решения с помощью компьютера. Имеется ряд прикладных программ таких как Matlab, Matcad и других, позволяющих достаточно просто решать такую задачу. Но вместе с тем и программа MS Excel предлагает свои возможности для решения систем n линейных уравнений с n неизвестными.

Запись вида

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases}$$

где a_{ij} – произвольные числа, являющиеся коэффициентами при неизвестных, x_i – неизвестные, подлежащие вычислению, b_i – произвольные числа, представляющие собой свободные члены.

Такая форма записи системы линейных уравнений называется *нормальной формой*.

Если система уравнений имеет хотя бы одно решение, то она называется *совместной*, в противном случае – она *несовместна*.

Если система уравнений имеет только одно решение, то она называется *определенной*, если она имеет несколько решений, то ее называют *неопределенной*.

Приведенную систему уравнений можно представлять и в матричном виде:

$$A \times X = B$$

где A – матрица коэффициентов или матрица системы:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}$$

X – матрица – столбец (вектор) неизвестных



$$X = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$$

B – матрица – столбец (вектор) свободных членов

$$B = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}$$

В развернутом виде систему можно записать следующим образом:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \times \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}$$

Существует ряд методов решения таких систем, в частности, метод Крамера, Гаусса, обратной матрицы и др.

Наиболее удобным методом решения такого вида систем уравнений с помощью программы MS Excel является метод обратной матрицы.

17.1. Метод обратной матрицы

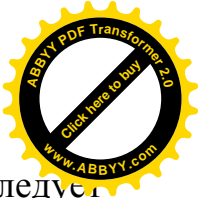
Решение системы с помощью этого способа представляется в виде:

$$X = A^{-1} \times B .$$

Программа MS Excel располагает необходимыми средствами для нахождения обратной матрицы и умножения матриц, что обеспечивает решения системы уравнений.

Порядок операций для решения системы линейных уравнений средствами MS Excel следующий:

- ввести (разместить) матрицу коэффициентов A в определенной области электронной таблицы,
- ввести матрицу – столбец свободных членов B ,
- выделить область ячеек под обратную матрицу (такого же размера, как и под матрицу коэффициентов A),
- щелкнуть по кнопке **Вставки функции**,
- в раскрывшемся окне **Мастер функций**, в поле **Категория**, выбрать строку **Математические**, а в поле **Функция** – имя функции **МОБР**,
- щелкнуть по кнопке **ОК**,
- в раскрывшемся окне **МОБР**, в рабочем поле **Массив**, указать диапазон ячеек, занимаемый исходной матрицей A ,



– нажать сочетание клавиш **Ctrl+Shift+Enter**.

Если обратная матрица не появилась в указанном диапазоне, то следует щелкнуть мышью в строке формул и повторить нажатие клавиш **Ctrl+Shift+Enter**.

- выделить диапазон ячеек под матрицу – столбец (из n ячеек) неизвестных X ,
- щелкнуть по кнопке **Вставки функции**,
- в раскрывшемся окне **Мастер функций**, в поле **Категория**, выбрать строку **Математические**, а в поле **Функция** – имя функции **МУМНОЖ**,
- щелкнуть по кнопке **ОК**,
- в раскрывшемся окне **МУМНОЖ**, в рабочем поле **Массив1** ввести диапазон ячеек, занимаемой обратной матрицей A^{-1} , а в рабочем поле **Массив2** ввести диапазон ячеек, занимаемой обратной матрицей – столбцом свободных членов B ,
- нажать сочетание клавиш **Ctrl+Shift+Enter**.

Если вектор X не появится в указанном диапазоне, то следует щелкнуть мышью в строке формул и повторить нажатие клавиш **Ctrl+Shift+Enter**.

После получения решения следует произвести проверку результата, выполнив операцию $A \times X$. Полученный результат должен быть равен вектору B .

ПРИМЕР 17.1. Решить систему методом обратной матрицы:

$$\begin{cases} x_2 - 13x_3 + 4x_4 = -5 \\ x_1 - 2x_3 + 3x_4 = -4 \\ 3x_1 + 21x_2 - 5x_4 = 2 \\ 4x_1 + 3x_2 - 5x_3 = 5 \end{cases}$$

В этом случае матрица коэффициентов **A** и вектор свободных коэффициентов **b** имеют вид:

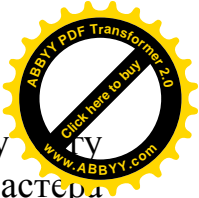
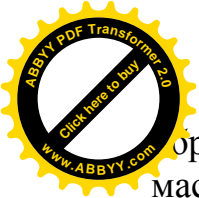
$$A = \begin{pmatrix} 0 & 1 & -13 & 4 \\ 1 & 0 & -2 & 3 \\ 3 & 21 & 0 & -5 \\ 4 & 3 & -5 & 0 \end{pmatrix} \quad b = \begin{pmatrix} -5 \\ -4 \\ 2 \\ 3 \end{pmatrix}$$

Введём матрицу **A** и вектор **b** в рабочий лист MS Excel (рис. 17.1).

| | A | B | C | D | E | F | G | H |
|---|----|---|----|-----|----|----|----|---|
| 1 | | 0 | 1 | -13 | 4 | | -5 | |
| 2 | A= | 1 | 0 | -2 | 3 | b= | -4 | |
| 3 | | 3 | 21 | 0 | -5 | | 2 | |
| 4 | | 4 | 3 | -5 | 0 | | 3 | |
| 5 | | | | | | | | |

Рис. 17.1

В нашем случае матрица **A** находится в ячейках **B1:E4**, а вектор **b** в диапазоне **G1:G4**. Для решения системы методом обратной матрицы необходимо вычислить матрицу, обратную к **A**. Для этого выделим ячейки для хранения обратной матрицы (это нужно сделать обязательно!!!); пусть в нашем случае это будут ячейки **B6:E9**. Теперь обратимся к мастеру функций, и в категории **Математические** выберем функцию **МОБР**, предназначенную для вычисления



обратной матрицы (рис. 3.2), щелкнув по кнопке **ОК**, перейдем ко второму мастеру функций. В диалоговом окне, появляющемся на втором шаге мастера функций, необходимо заполнить поле ввода Массив (рис. 17.3). Это поле должно содержать диапазон ячеек, в котором хранится исходная матрица - в нашем случае **B1:E4**. Данные в поле ввода Массив можно ввести, используя клавиатуру или выделив их на рабочем листе, удерживая левую кнопку мыши.

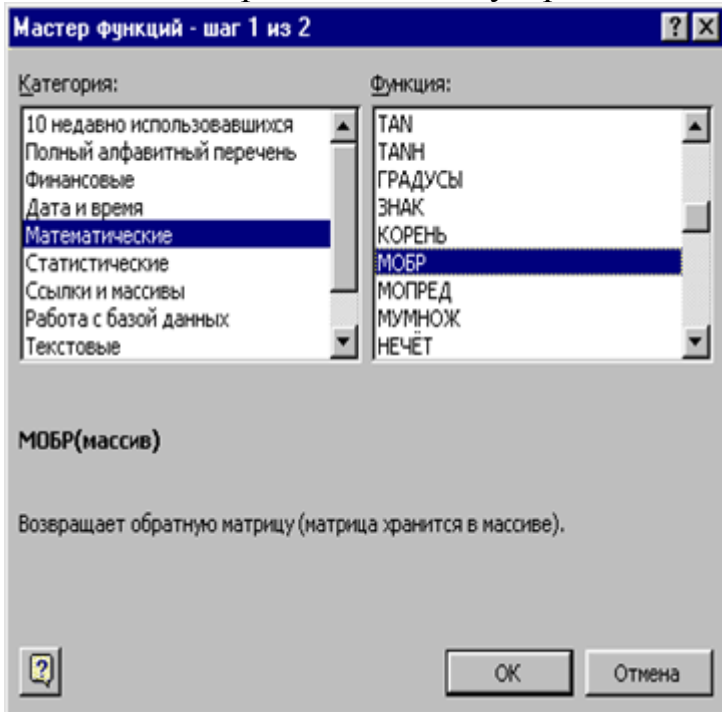


Рис. 17.2

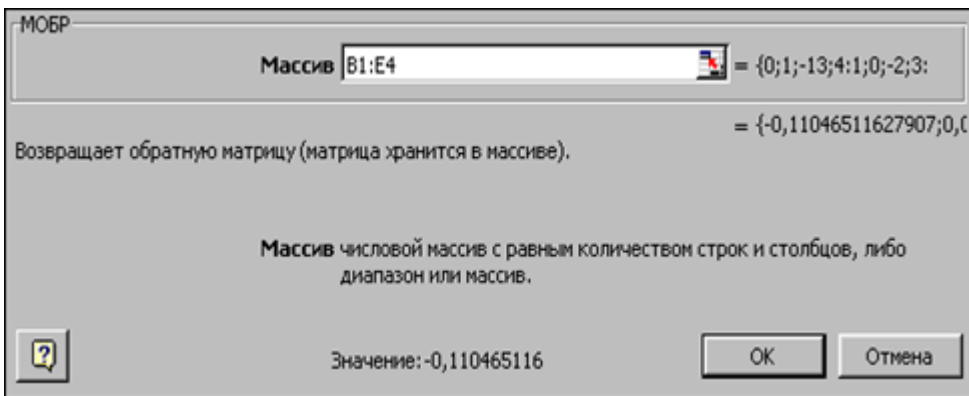
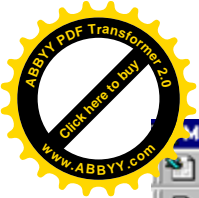


Рис. 17.3

Если поле Массив заполнено, можно нажать кнопку **ОК**. В первой ячейке, выделенного под обратную матрицу диапазона, появится некое число. Для того чтобы получить всю обратную матрицу, необходимо нажать клавишу **F2** для перехода в режим редактирования, а затем одновременно клавиши **Ctrl+Shift+Enter**. В нашем случае рабочая книга MS Excel примет вид изображенный на рис. 17.4.



Microsoft Excel - Книга1

Файл Правка Вид Вставка Формат Сервис Данные Окно ?

Arial Cyr 10 Ж К Ц

D7 = (=МОБР(B1:E4))

| | A | B | C | D | E | F | G |
|----|----|----------|----------|----------|----------|----|----|
| 1 | | 0 | 1 | -13 | 4 | | -5 |
| 2 | A= | 1 | 0 | -2 | 3 | b= | -4 |
| 3 | | 3 | 21 | 0 | -5 | | 2 |
| 4 | | 4 | 3 | -5 | 0 | | 3 |
| 5 | | | | | | | |
| 6 | | -0,11047 | 0,096899 | -0,03023 | 0,24845 | | |
| 7 | | 0,011628 | 0,077519 | 0,055814 | -0,06124 | | |
| 8 | | -0,0814 | 0,124031 | 0,009302 | -0,03798 | | |
| 9 | | -0,01744 | 0,383721 | 0,016279 | -0,10814 | | |
| 10 | | | | | | | |

Рис. 17.4

Теперь необходимо умножить полученную обратную матрицу на вектор **b**. Выделим ячейки для хранения результирующего вектора, например **H6:H9**. Обратимся к мастеру функций, и в категории **Математические** выберем функцию **МУМНОЖ**, которая предназначена для умножения матриц. Напомним, что умножение матриц происходит по правилу строка на столбец и матрицу **A** можно умножить на матрицу **B** только в том случае, если количество столбцов матрицы **A** равно количеству строк матрицы **B**. Кроме того, при умножении матриц важен порядок сомножителей, т.е. **AB≠BA**.

Перейдем ко второму шагу мастера функций. Появившееся диалоговое окно (рис. 17.5) содержит два поля ввода **Массив1** и **Массив2**. В поле **Массив1** необходимо ввести диапазон ячеек, в котором содержится первая из перемножаемых матриц, в нашем случае **B6:E9** (обратная матрица), а в поле **Массив2** ячейки, содержащие вторую матрицу, в нашем случае **G1:G4** (вектор **b**).

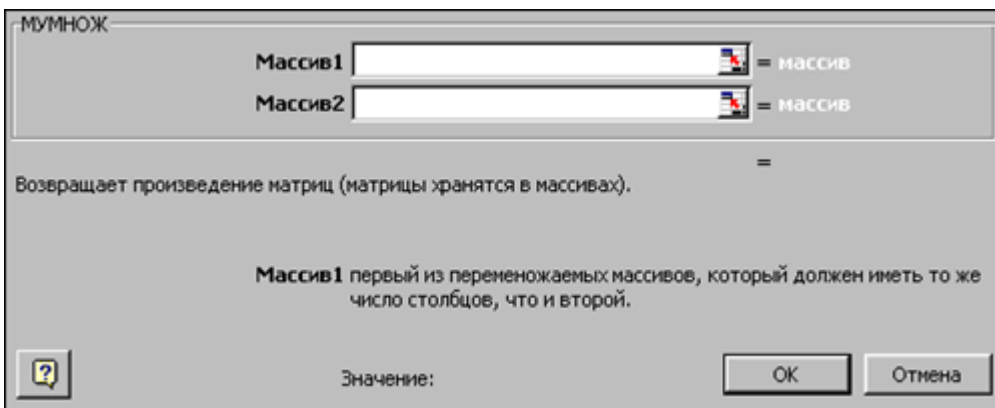
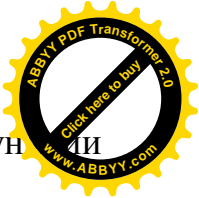
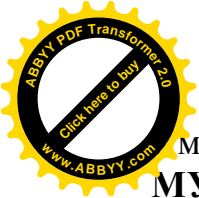


Рис. 17.5

Если поля ввода заполнены, можно нажать кнопку **ОК**. В первой ячейке выделенного диапазона появится соответствующее число результирующего вектора. Для того чтобы получить весь вектор, необходимо нажать клавишу **F2**, а затем одновременно клавиши **Ctrl+Shift+Enter**. В нашем случае результаты вычислений (вектор **x**), находится в ячейках **H6:H9**.

Для того чтобы проверить, правильно ли решена система уравнений, необходимо умножить матрицу **A** на вектор **x** и получить в результате вектор **b**.



множение матрицы **A** на вектор **x** осуществляется при помощи функции **МУМНОЖ(В1:Е4;Н6:Н9)**, так как было описанной выше.

В результате проведенных вычислений рабочий лист примет вид изображенный на рис. 17.6.

| | A | B | C | D | E | F | G | H | I | J |
|---|----|----------|----------|----------|----------|----|----|----------|----------|----|
| 1 | | 0 | 1 | -13 | 4 | | -5 | | | -5 |
| 2 | A= | 1 | 0 | -2 | 3 | b= | -4 | | Проверка | -4 |
| 3 | | 3 | 21 | 0 | -5 | | 2 | | | |
| 4 | | 4 | 3 | -5 | 0 | | 3 | | | |
| 5 | | | | | | | | | | |
| 6 | | -0,11047 | 0,096899 | -0,03023 | 0,24845 | | | 0,849612 | | |
| 7 | | 0,011628 | 0,077519 | 0,055814 | -0,06124 | | x= | -0,44031 | | |
| 8 | | -0,0814 | 0,124031 | 0,009302 | -0,03798 | | | -0,1845 | | |
| 9 | | -0,01744 | 0,383721 | 0,016279 | -0,10814 | | | -1,73953 | | |
| | | | | | | | | | | |

Рис. 17.6

ПРИМЕР 17.2. Решить систему из ПРИМЕРА 17.1 методом Крамера.

Введём матрицу **A** и вектор **b** на рабочий лист. Кроме того, сформируем четыре вспомогательные матрицы, заменяя последовательно столбцы матрицы **A** на столбец вектора **b** (рис. 17.7).

Для дальнейшего решения необходимо вычислить определитель матрицы **A**. Установим курсор в ячейку **I10** и обратимся к мастеру функций. В категории **Математические** выберем функцию **МОПРЕД**, предназначенную для вычисления определителя матрицы, и перейдём ко второму шагу мастера функций. Диалоговое окно, появляющееся на втором шаге содержит поле ввода **Массив**. В этом поле указывают диапазон матрицы, определитель которой вычисляют. В нашем случае это ячейки **B1:E4**.

Для вычисления вспомогательных определителей введем формулы:

$$I11=МОПРЕД(B6:E9),$$

$$I12=МОПРЕД(B11:E14),$$

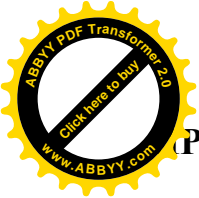
$$I13=МОПРЕД(B16:E19), I14=МОПРЕД(B21:E24).$$

В результате в ячейке **I10** хранится главный определитель, а в ячейках **I11:I14** - вспомогательные.

Воспользуемся формулами Крамера и разделим последовательно вспомогательные определители на главный. В ячейку **K11** введём формулу $=I11/I10$. Затем скопируем её содержимое в ячейки **K12**, **K13** и **K14**. Система решена.

| | A | B | C | D | E | F | G | H | I | J | K |
|----|-----|----|----|-----|----|----|----|-----|-------|----|----------|
| 1 | | 0 | 1 | -13 | 4 | | -5 | | | | |
| 2 | A= | 1 | 0 | -2 | 3 | b= | -4 | | | | |
| 3 | | 3 | 21 | 0 | -5 | | 2 | | | | |
| 4 | | 4 | 3 | -5 | 0 | | 3 | | | | |
| 5 | | | | | | | | | | | |
| 6 | | -5 | 1 | -13 | 4 | | | | | | |
| 7 | A1= | -4 | 0 | -2 | 3 | | | | | | |
| 8 | | 2 | 21 | 0 | -5 | | | | | | |
| 9 | | 3 | 3 | -5 | 0 | | | | | | |
| 10 | | | | | | | | | | | |
| 11 | | 0 | -5 | -13 | 4 | | | d= | 2580 | | |
| 12 | A2= | 1 | -4 | -2 | 3 | | | d1= | 2192 | x= | 0,849612 |
| 13 | | 3 | 2 | 0 | -5 | | | d2= | -1136 | | -0,44031 |
| 14 | | 4 | 3 | -5 | 0 | | | d3= | -476 | | -0,1845 |
| 15 | | | | | | | | d4= | -4488 | | -1,73953 |
| 16 | | 0 | 1 | -5 | 4 | | | | | | |
| 17 | A3= | 1 | 0 | -4 | 3 | | | | | | |
| 18 | | 3 | 21 | 2 | -5 | | | | | | |
| 19 | | 4 | 3 | 3 | 0 | | | | | | |
| 20 | | | | | | | | | | | |
| 21 | | 0 | 1 | -13 | -5 | | | | | | |
| 22 | A4= | 1 | 0 | -2 | -4 | | | | | | |
| 23 | | 3 | 21 | 0 | 2 | | | | | | |
| 24 | | 4 | 3 | -5 | 3 | | | | | | |
| | | | | | | | | | | | |

Рис. 17.7



ПРИМЕР 17.3. Вычислить матрицу **C** по формуле: $C=A^2+2AB$, где

$$A = \begin{pmatrix} 3 & 9 & -2 \\ 2 & -13 & 3 \\ 11 & 2 & 4 \end{pmatrix}; \quad B = \begin{pmatrix} 1 & 4 & 11 \\ 4 & 5 & 5 \\ 11 & 3 & 7 \end{pmatrix}$$

Введем исходные данные на рабочий лист (рис. 17.8).

Для умножения матрицы **A** на матрицу **B**, выделим диапазон **B5:D7** и воспользуемся функцией **МУМНОЖ(B1:D3;G1:I3)**.

Результат вычисления $A^2=A*A$ поместим в ячейки **G5:I7**, воспользовавшись формулой **МУМНОЖ(B1:D3;B1:D3)**.

Умножение (деление) матрицы на число можно выполнить при помощи элементарных операций. В нашем случае необходимо умножить матрицу из диапазона **B5:D7** на число 2. Выделим ячейки **B9:D11** и введем формулу **=2*B5:D7**.

Сложение (вычитание) матриц выполняется аналогично. Например, выделим диапазон **G9:I11** и введем формулу **=B9:D11+ G5:I7**.

Для получения результата в обоих случаях необходимо нажать комбинацию клавиш **Ctrl+Shift+Enter**.

Кроме того, в строке формул рабочего листа, изображенного на рис. 3.8, показано как можно вычислить матрицу **C** одним выражением.

| | A | B | C | D | E | F | G | H | I |
|----|------|-----|-----|-----|----|------------------------|-----|-----|-----|
| 1 | | 3 | 9 | -2 | | | 1 | 4 | 11 |
| 2 | A= | 2 | -13 | 3 | | B= | 4 | 5 | 5 |
| 3 | | 11 | 2 | 4 | | | 11 | 3 | 7 |
| 4 | | | | | | | | | |
| 5 | | 17 | 51 | 64 | | | 5 | -94 | 13 |
| 6 | AB= | -17 | -48 | -22 | | A ² | 13 | 193 | -31 |
| 7 | | 63 | 66 | 159 | | | 81 | 81 | 0 |
| 8 | | | | | | | | | |
| 9 | | 34 | 102 | 128 | | | 39 | 8 | 141 |
| 10 | 2AB= | -34 | -96 | -44 | | C=A ² +2AB= | -21 | 97 | -75 |
| 11 | | 126 | 132 | 318 | | | 207 | 213 | 318 |
| 12 | | | | | | | | | |
| 13 | | | | | | | | | |
| 14 | | | | | C= | | 39 | 8 | 141 |
| 15 | | | | | | | -21 | 97 | -75 |
| 16 | | | | | | | 207 | 213 | 318 |
| 17 | | | | | | | | | |

Рис. 17.8



17.2 Решения системы линейных уравнений методом Гаусса

Система линейных алгебраических уравнений с n неизвестными имеет вид:

Нужно найти ее решение x_1, \dots, x_n .

В матричной форме система имеет вид $AX=B$.

A - матрица системы, B - матрица свободных членов, X - матрица неизвестных

Метод Гаусса - это наиболее известный метод решения системы.

Согласно методу, с помощью эквивалентных преобразований матрицу приводят к виду треугольной. Сначала выключают неизвестную X_1 из всех

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} B = \begin{pmatrix} b_1 \\ b_2 \\ \cdots \\ b_n \end{pmatrix} X = \begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_n \end{pmatrix}$$

уравнений системы, начиная из второго.

Эту процедуру выполняют еще $(n - 1)$ раз, прежде чем получить систему с треугольной матрицей. Этот этап преобразования носит название прямым ходом. Потом начинают второй этап, который носит название обратного хода. Из последнего уравнения находят X_n , подставляют его у уравнение, которое стоит перед ним, ищут X_{n-1} и т.д. к X_1 . Таким образом, находят решения системы.

Рассмотрим порядок выполнения задачи

В ячейки A1:D4 вводим матрицу A , то есть матрицу коэффициентов системы уравнений. В ячейки E1:E4 - матрицу B , то есть матрицу свободных элементов.

Первая строка матрицы A копируем в ячейки A6:E6, A11:E11, A16:E16. Выделяем ячейки A7:D7 и записываем формулу: $=\{A2:E2-\$A\$1:\$E\$1*(A2/\$A\$1)\}$

Потом нажимаем комбинацию клавиш Ctrl+Shift+Enter.

Копируем формулу из ячеек A7:E7 в ячейки A8:E9.

Значения из ячеек A7:E7 записываем в ячейки A12:E12, A17:E17. В ячейках A13:E13 записываем формулу: $=\{A8:E8-\$A\$7:\$E\$7*(B8/\$B\$7)\}$

Копируем формулу с A13:E13 в ячейки A14:E14. Из интервала A13:E13 значение копируем в ячейки A18:E18. В ячейки A19:E19 записываем формулу: $=\{A14:E14-\$A\$13:\$E\$13*(C14/\$C\$13)\}$

Потом начинаем обратный ход.

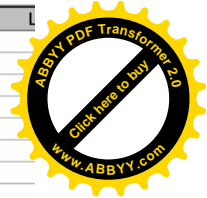
В ячейки G4:K4 записываем формулу: $=\{A19:E19/D19\}$

В ячейки G3:K3 записываем формулу: $=\{(A18:E18-G4:K4*D18)/C18\}$

В ячейки G2:K2 записываем: $=\{(A17:E17-G4:K4*D17-G3:K3*C17)/B17\}$



| | A | B | C | D | E | F | G | H | I | J | K | L |
|----|---|------|-------|-------|------|---|------|------|------|------|-------|---|
| 1 | 2 | 3 | 5 | 2 | 3 | | 1,00 | 0,00 | 0,00 | 0,00 | -0,48 | |
| 2 | 5 | 2 | 7 | 5 | 2 | | 0,00 | 1,00 | 0,00 | 0,00 | 0,94 | |
| 3 | 4 | 2 | 1 | 7 | 3 | | 0,00 | 0,00 | 1,00 | 0,00 | 0,06 | |
| 4 | 7 | 5 | 4 | 1 | 2 | | 0,00 | 0,00 | 0,00 | 1,00 | 0,43 | |
| 5 | | | | | | | | | | | | |
| 6 | 2 | 3 | 5 | 2 | 3 | | | | | | | |
| 7 | 0 | -5,5 | -5,5 | 0 | -5,5 | | | | | | | |
| 8 | 0 | -4 | -9 | 3 | -3 | | | | | | | |
| 9 | 0 | -5,5 | -13,5 | -6 | -8,5 | | | | | | | |
| 10 | | | | | | | | | | | | |
| 11 | 2 | 3 | 5 | 2 | 3 | | | | | | | |
| 12 | 0 | -5,5 | -5,5 | 0 | -5,5 | | | | | | | |
| 13 | 0 | 0 | -5 | 3 | 1 | | | | | | | |
| 14 | 0 | 0 | -8 | -6 | -3 | | | | | | | |
| 15 | | | | | | | | | | | | |
| 16 | 2 | 3 | 5 | 2 | 3 | | | | | | | |
| 17 | 0 | -5,5 | -5,5 | 0 | -5,5 | | | | | | | |
| 18 | 0 | 0 | -5 | 3 | 1 | | | | | | | |
| 19 | 0 | 0 | 0 | -10,8 | -4,6 | | | | | | | |
| 20 | | | | | | | | | | | | |



В ячейки G1:K1 записываем формулу: $=\{(A16:E16-G4:K4*D16-G3:K3*C16-G2:K2*B16)/A16\}$

Рис 17.9 Решение системы уравнений методом Гаусса

17.3 Решение системы линейных уравнений с помощью надстройки "Поиск решения"

Можно решить данную систему в Excel проще, используя не метод Гауса, а надстройку *Поиск решения* с меню *Сервис*. Сначала присвоим ячейкам A37, B37, C37 и D37 имена переменных x_1 , x_2 , x_3 и x_4 , а в ячейках A28 : D38 будем получать решение уравнений соответственно (для большей наглядности решения). Выше в ячейки D30: D33 записываем формулы, которые отвечают левым частям уравнений системы. Например, в ячейке D30 записываем формулу, которая отвечает первому уравнению $=2*A38 + 3*B38 + 5*C38 + 2*D38$ и т.п. Выбираем команду *Поиск решения* в меню *Сервис*.

В диалоговом окне *Поиск решения* выбираем ячейку D30 в качестве целевой и требуем установить ее равной 3 за изменение значений в ячейках A38 : D38. Переходим к ограничениям. Нажимаем кнопку *Добавить*, выбираем ячейку с следующим уравнением D31 и будем требовать, чтобы оно равнялось 2. Аналогично добавим еще 2 ограничения.

Нажимаем кнопку *Параметры*, появляется окно *Параметры Поиск решения*. Устанавливаем флажок *Линейная модель*. Нажимаем кнопку ОК в окне параметров, а потом кнопку *Выполнить* в окне поиска решений.

Excel сигнализирует, что решение найдено и предлагает его сохранить, что мы и делаем. Полученное решение совпадает с решением, полученным методом Гаусса.

Проверка правильности решения.

Подставим полученные значения (округленные до тысячных) в начальные уравнения

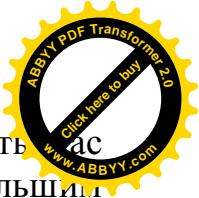
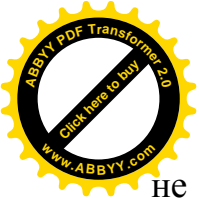
$$2 \cdot (-0.48) + 3 \cdot 0.94 + 5 \cdot 0.06 + 2 \cdot 0.43 = 3.02000;$$

$$5 \cdot (-0.48) + 2 \cdot 0.94 + 7 \cdot 0.06 + 5 \cdot 0.43 = 2.05000;$$

$$4 \cdot (-0.48) + 2 \cdot 0.94 + 1 \cdot 0.06 + 7 \cdot 0.43 = 3.03000;$$

$$7 \cdot (-0.48) + 5 \cdot 0.94 + 4 \cdot 0.06 + 1 \cdot 0.43 = 2.01000$$

Эта же проверка может быть выполнена с помощью формулы $=\text{МУМНОЖ}(A1:D4;K1:K4)$ (точная проверка), что надо записать в ячейки I8:I11.



Мы видим, что погрешность не превышает 0,05. Если такая точность не устраивает, то можем округлить найденные Excel значения с большим количеством знаков после запятой .

ЛЕКЦИЯ №18

18. НАХОЖДЕНИЯ КОРНЕЙ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

Общий вид алгебраического уравнения:

$$F(X) = A_0X^n + A_1X^{n-1} + A_2X^{n-2} \dots + A_n = 0, \quad (A_0 \neq 0), \quad \text{де}$$

$F(X)$ -многочлен степени n относительно x , A_i - коэффициенты при неизвестных, A_n - свободный член многочлена, X - неизвестная, n - степень уравнения.

Общие формулы, для поиска корней алгебраических уравнений существуют для уравнений

первой степени (линейные уравнения) $Ax = y$; $x = y/A$, якщо $A \neq 0$;

второй степени (квадратные уравнения) $Ax^2 + Bx + C = 0$;

$$x_{1,2} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

третьей степени (кубические уравнения)

$$X^3 + aX^2 + bX + c = 0 \quad (1)$$

В этом случае существует так называемый стандартный метод решения. Он состоит в следующем:

I. В уравнении (1) делается подстановка: $x = y - a/3$

И вместо (1) получаем неполное кубическое уравнение:

$$y^3 + py + q = 0; \quad \text{де} \quad P = -a^2/3 + b, \quad \text{а} \quad q = 2(a/3)^3 - ab/3 + c$$

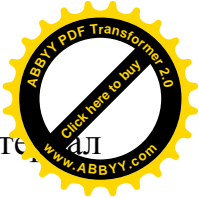
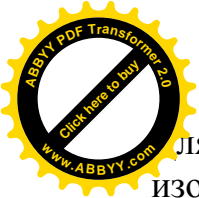
В дальнейшем вычисляют значения сменной $Q = (p/3)^3 + (q/2)^2$ и считают, что при $Q > 0$ уравнения имеет одних действительный и два комплексных корня,

$$y_1 = A + B \quad y_{2,3} = -\frac{A+B}{2} \pm i \frac{A-B}{2} \sqrt{3}$$

$$\text{де} \quad A = \sqrt[3]{-q/2 + \sqrt{Q}}, \quad B = \sqrt[3]{-q/2 - \sqrt{Q}}$$

при $Q = 0$ получаем три действительных корня, два с которых уровни, а при $Q < 0$ уравнения имеет три разных действительных корня.

Наиболее распространенными способами решения уравнений с помощью компьютера являются методы последовательных приближений. При этом выбирается начальное значение корня, которое потом последовательно уточняется.



Для поиска начального значения корня нужно найти так называемый интервал изоляции корня, такой, что сам корень будет находиться внутри него.

Поиск интервала изоляции выполняется на базе известной особенности непрерывных функций: если $f(x)$ непрерывна на интервале $[a, b]$ и на его концах имеет противоположные знаки: $f(a) > 0$, $f(b) < 0$, то внутри интервала может быть хотя бы один корень уравнения $f(x) = 0$.

Оценить корень, или найти интервал его изоляции можно графически. Если построить график функции $y = f(x)$, то абсцисса точки пересечения кривой с осью OX дает приблизительные значения действительных корней уравнения $f(x) = 0$ и разрешает указать для каждого корня интервал его изоляции.

Когда интервал изоляции выделен, корень уточняют каким-то из методов последовательных приближений, как это показано в дальнейшем на примере. Инструмент пакета Excel, что нужен для этого - *Подбор параметра*, который находится в меню *Сервис*.

Пример решения уравнения третьего степени с помощью Excel.

Предположим, что уравнения имеет вид:

$$X^3 - 0,01X^2 - 0,7044X + 0,139104 = 0, \text{ где } X \in [-1; 1], \text{ та } H_X = 0,2$$

Порядок поиска корней уравнения включает несколько шагов.

1. Согласно задаче с помощью Excel методом автозаполнения находим значение x , что выбираем из интервала $X \in [-1; 1]$ с шагом $H_X = 0,2$ (Исправления - Заполнить - Прогрессия). Значения x заносим в столбик A5:A17

2. Рассчитываем значение $F(x)$. Формулу в виде $=A5^3 - 0,01 * A5^2 - 0,7044 * A5 + 0,139104$ заносим в клеточку B5 и выполняем копирование ее в диапазон B6:B17.

3. По данным столбцов A5:A17 и B5:B17 строим точечный график.

4. Определяем интервалы, где полином изменяет свой знак.

$$[-1,0 ; -0,8]$$

$$[-0,2 ; 0,4]$$

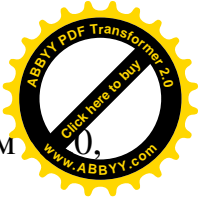
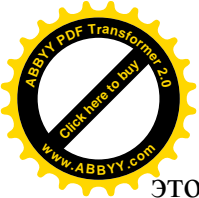
$$[-0,6 ; 0,8]. \text{ Это и есть интервалы локализации корня}$$

5. Исследуем уравнение на предмет количества изменений знака, который для этого уравнения равняется 2. Согласно теории это указывает на то, что график функции трижды пересечет ось X и даст два положительных и один отрицательный корня уравнения.

6. Превращаем уравнение к виду неполного кубического уравнения:

$$y^3 + py + q = 0; \quad \text{де } P = -a^2/3 + b, \quad \text{а } q = 2(a/3)^3 - ab/3 + c$$

и вычисляем значение переменной $Q = (p/3)^3 + (q/2)^2$



Вычисляем значение коэффициентов a, v, c, p, q (ячейки B20-B25). Имеем это означает, что уравнения имеет три разных действительных корня.

7. Дальнейший поиск корней уравнение выполняем с помощью инструмента Excel Подбор параметра. (Меню *Сервис* → *Подбор параметра*)

| | | | | |
|----------|---|-----|-----------------------------------|--------------|
| В ячейку | } | B29 | | [-1 ; - 0,8] |
| | | B30 | вносим любое значение из отрезков | [-0,2 ; 0,4] |
| | | B31 | | [-0,6 ; 0,8] |

Безусловно, этот шаг имеет важное значение. Если не установить значения из интервалов решения, можно найти не все корни уравнения, или получить два или трех одинаковые значения.

| | | | |
|----------|---|-----|--|
| В ячейку | } | C29 | |
| | | C30 | записуемо формулу = B29^3 – 0,01*B29^2 – 0,7044* |
| | | | B29+0,13904 |
| | | C31 | |

Вызываем режим *Сервис* - > *Подбор параметров*.

В окне *Подбор параметров* задаем адрес ячейки, где размещаем результат, значения которого необходимо достичь и интервал переменных ячеек, как это показано на рис.18.1.

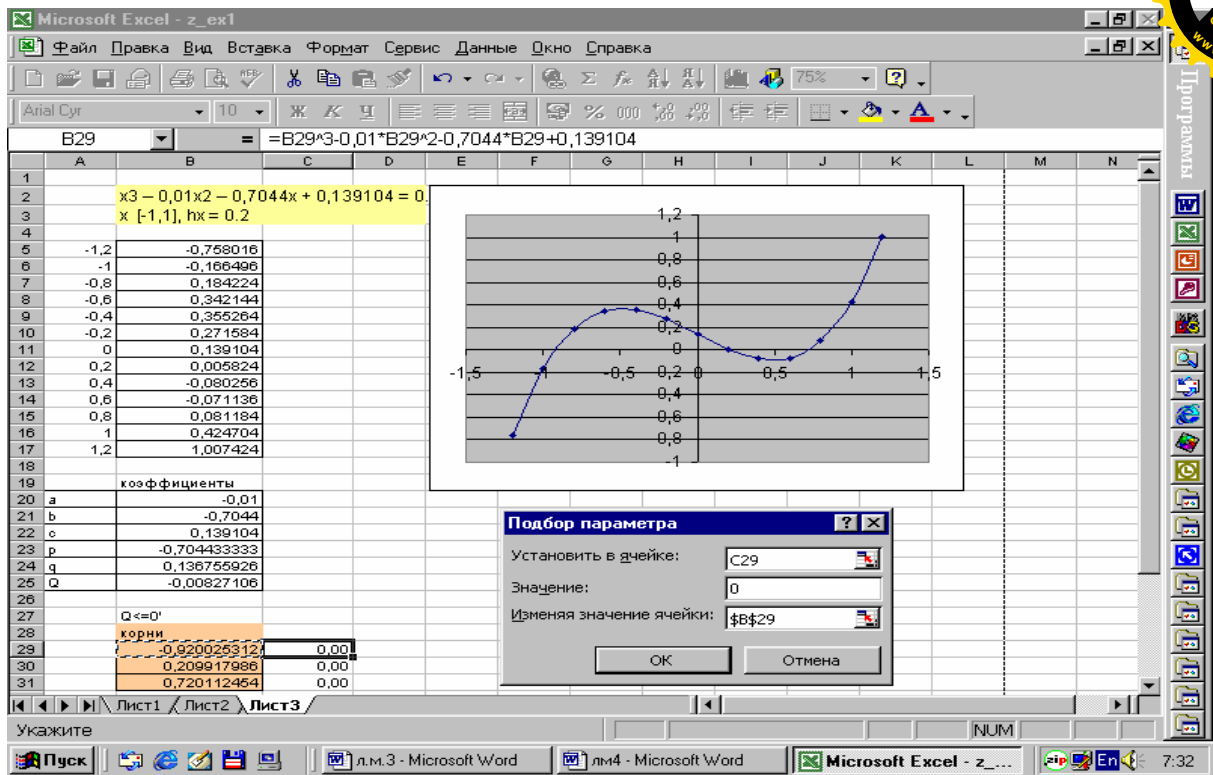
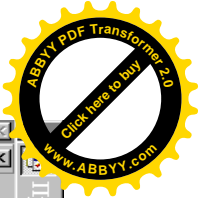


Рис. 18.1 Порядок решения нелинейных уравнений с помощью MS Excel

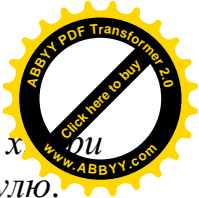
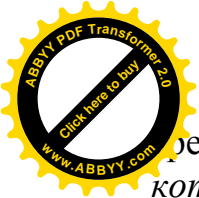
После нажатия кнопки ОК получаем окно *Результат подбора параметра*, проверяем, нашли ли решения и удовлетворяет ли оно нас. Снова нажимаем ОК и получаем результат.

В примере результат решения (корни уравнений) записанные в ячейки B29-B31, так что $x_1 = -0,92003$, $x_2 = 0,209918$, $x_3 = 0,720112$. Выполняем проверку решения и заканчиваем задачу.

18.1. Решение уравнения с одним неизвестным

Определение корней уравнения вида $f(x) = 0$ даже в случае алгебраических уравнений выше третьей степени представляет достаточно сложную задачу. Трансцендентные уравнения чаще всего вообще не имеют аналитического решения.

Трансцендентные – это уравнения, содержащие различные нелинейные функции $\sin(x)$, $\cos(x)$, $\ln(x)$, e^x и т. п. Например, $\sin(2x) - \ln(x^2) + e^x = 0$. В таких случаях единственным путем является получение приближенных решений. Обычно для нахождения приближенного решения используют *итерационные методы*. Все эти методы основаны на том, что вначале берут какое – то первое значение искомой переменной x подставляют его в правую часть уравнения и вычисляют ее. Если полученное значение существенно отличается от нуля, то берут следующее значение x (у каждого метода имеются свои правила определения этого значения) и опять вычисляется правая часть уравнения. Этих шагов будет сделано столько, сколько необходимо для того, чтобы правая часть после подстановки очередного значения x мало отличалась бы от нуля (в пределах



требуемой точности). Решением таких уравнений является значение x , в котором правая часть обращается в нуль или становится очень близкой к нулю.

В MS Excel для решения таких уравнений используется удобный и простой для понимания инструмент **Подбор параметра**. Он реализует алгоритм численного решения уравнений с одной неизвестной. Весь процесс решения состоит из двух этапов:

- задание на рабочем листе двух ячеек: ячейки, куда помещается значение искомой переменной (влияющей ячейки) и ячейки, куда помещается выражение правой части уравнения (зависящей ячейки);
- ввод адресов влияющей и зависящей ячеек в диалоговом окне **Подбор параметра** с дальнейшим получением ответа.

К сожалению, с помощью процедуры **Подбор параметра** можно решить не все типы уравнений.

Прянок решения таких уравнений удобно рассмотреть на примере. Найти решение уравнения $\ln(x) = 0$.

Первый этап.

- командами *Вставка – Лист открыть новый рабочий лист*,
- в выбранную ячейку, например *A1*, занести ориентировочное значение x , например, *3*,
- в другую выбранную ячейку, например, *B1* занести правую часть уравнения: $\ln(A1)$. Здесь вместо x записан адрес ячейки *A1*, в которой содержится значение переменной x ,
- щелкнуть по кнопке *ОК*. В ячейке *B1* появляется число *1,098612*.

Второй этап.

- выполнить команды **Сервис – Подбор параметра**,
- в раскрывшемся окне **Подбор параметра**, в поле **Установить в ячейке**, щелчком мыши по ячейке *B1* отобразить адрес этой ячейки, в поле **Значение установить с клавиатуры** значение правой части уравнения *0*, а в поле **Изменяя значение ячейки** мышью указать адрес ячейки *A1*,
- щелкнуть по кнопке **ОК**.

После этих действий в ячейке *A1* отобразится число *0,999872*, что является приближенным решением уравнения $\ln(x) = 0$. При этом в ячейке *B1* будет представлено число: $-0,00013$, что составляет погрешность вычисления. При $x = 0,999872$ значение $\ln(x)$ составляет $-0,00013$, что близко к нулю. Известно, что точным решением является $x = 1$.

Нахождение корней уравнения путем подбора параметров

$$X^3 - 0,01x^2 - 0,7044x - 0,139 = 0$$

Так как данный полином третьей степени, то имеется не более трех корней.

Для нахождения корней их необходимо локализовать, то есть найти интервалы, на которых эти корни существуют. Такими интервалами могут служить промежутки, на концах которых функция имеет противоположный знак.

С целью нахождения интервала необходимо построить график данного уравнения. После построения графика выбрать интервалы, где график пересекает ось абсцисс. Эти интервалы будут приближенными. Затем, используя подбор параметров, найти точное значение корней полинома.



ЛЕКЦИЯ №19

19. ЭЛЕКТРОННЫЕ ТАБЛИЦЫ MS EXCEL. ФОРМАТИРОВАНИЕ ДАННЫХ. ОПЕРАЦИИ С БОЛЬШИМИ ТАБЛИЦАМИ И ЛИСТАМИ

19.1. Форматирование данных

Форматирование – это представление информации в удобной форме. Программа MS Excel самостоятельно определяет тип данных и форматирует их в соответствии с правилами стандарта, установленного для каждого типа данных. *Следует помнить*, что выбранные приемы форматирования закрепляются *не за значением, а за ячейкой*. Это следует понимать как то, что в ячейке могут помещаться различные значения, а форма их представления остается неизменной. И с другой стороны, одно и то же значение, записанное в различных ячейках, с различными форматами, будет представляться по-разному. Например, пусть ячейка отформатирована таким образом, что вещественные числа отображаются с одним знаком после запятой. Если в эту ячейку попытаться поместить число 0,25, то это не удастся. Вместо него будет размещено число 0,3. То есть число 0,25 будет округлено до 0,3. Несмотря на это в расчетах будет использоваться *введенное* число 0,25, а не *отображенное* число 0,3.

Обычно данные вводятся в *общем формате*, который используется для отображения как *текстовых*, так и *числовых* данных. Если такой формат не удовлетворяет пользователя, то его *можно изменить на более удобный*. Это означает, что для каждой ячейки таблицы можно устанавливать *собственный формат данных*.

Для установления необходимого формата данных следует выполнить следующие операции:

- выделить ячейку или группу ячеек, для которых предполагается установить новый формат;
- открыть окно **Формат ячеек** одной из следующих операций:
 - нажать комбинацию клавиш Ctrl + 1;
 - вызвать контекстное меню (щелчком правой клавиши мыши), где выбрать пункт **Формат ячеек**;
 - использовать главное меню **Формат – Ячейки – Формат ячеек**;
- в раскрывшемся диалоговом окне сделать необходимые установки.

19.1.1. Форматирование текста

Ранее отмечалось, что размер текстового значения (непрерывная последовательность букв, цифр, специальных знаков), размещаемого в ячейке, не должен превышать 255 символов. Если же в ячейку требуется ввести текст достаточно большого размера, то следует использовать *режим переноса текста по словам*. Для этого необходимо

- в диалоговом окне **Формат ячеек** раскрыть вкладку **Выравнивание**;



– *установить флажок перед строкой* **Переносить по словам.**

После выполнения этих операций текст будет размещаться в нескольких строках внутри одной ячейки.

В диалоговом окне **Выравнивание** можно установить и другие параметры по аналогии с форматированием абзацев в текстовом редакторе MS Word: *выравнивание по горизонтали, выравнивание по вертикали, размер отступов.* По умолчанию текст выравнивается по *левому* краю. На этой же вкладке можно устанавливать различную ориентацию текста по предлагаемому образцу, то есть указать наклон в градусах.

На вкладках **Граница** и **Вид** окна **Формат ячеек** находятся элементы управления, с помощью которых для произвольной группы ячеек можно выбрать желаемый вариант оформления и заливки.

19.1.2. Форматирование чисел

Числа в ячейках представляются в *общем формате* и отображаются так, как они вводятся. При необходимости этот формат можно изменить. Программа MS Excel располагает всеми необходимыми для этого инструментами. Пусть в ячейку введено число 2378,2. В различных форматах это число будет представлено различным образом, что видно из таблицы 20.1.

Таблица 19.1 Форматы представления чисел

| <i>Общий</i> | <i>Числовой</i> | <i>Денежный</i> | <i>Процентный</i> | <i>Экспоненциальный</i> |
|--------------|-----------------|-----------------|-------------------|-------------------------|
| 2378,2 | 2378,20 | 2378,20 р. | 237820,00% | 2,38E+03 |

Здесь в числовом формате, рассматриваемого в примере числа, используется две позиции для представления дробной части. В принципе размер дробной части числа может быть установлен и другой.

Для установления необходимого формата числа следует в диалоговом окне **Формат ячеек** раскрыть вкладку **Число** и из списка **Числовые форматы**

- *общий*
- *числовой*
- *денежный*
- *финансовый*
- *дата*
- *время*
- *процентный*
- *дробный*
- *экспоненциальный*
- *дополнительный*
- *(все форматы)*



Выбрать необходимый формат. Кроме этого способа формат ячеек можно устанавливать с помощью кнопок

- денежный формат,
- процентный формат,
- формат с разделителями,
- увеличить разрядность,
- уменьшить разрядность,

расположенных на панели инструментов.

Если число должно представляться в *процентном* формате, то оно *автоматически умножается на 100*.

Если число должно представляться в *текстовом* формате, например, индекс почтового отделения, то перед ним следует ставить *апостроф* ('83245) или же выбрать текстовый формат на вкладке **Число** диалогового окна **Формат ячеек**.

19.1.3. Форматирование символов

Программа MS Excel предусматривает возможность форматирования символов. Этот процесс аналогичен тому, как это делается в текстовом редакторе MS Word. Для этой цели необходимо в диалоговом окне **Формат ячеек** раскрыть вкладку **Шрифт** и произвести необходимые установки по выбору *размера, начертания, толщины* шрифта или же воспользоваться кнопками на панели инструментов.

19.1.4. Автоформатирование

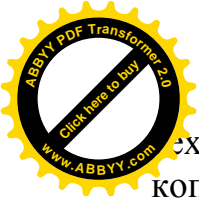
Наряду с рассмотренными приемами форматирования программа MS Excel предлагает несколько образцов (шаблонов) оформления документов. После выбора подходящего образца программа автоматически будет оформлять данные по этому образцу. Для этого следует:

- ✓ выделить произвольную группу смежных ячеек или таблицу целиком;
- ✓ выполнить последовательность операций **Формат – Автоформат**;
- ✓ в раскрывшемся диалоговом окне выбрать один из вариантов оформления;
- ✓ щелкнуть мышью по кнопке **Параметры** и появившейся дополнительной вкладке **Изменить**: (формат чисел, рамки, шрифт, узоры, выравнивание, ширину и высоту) установить или убрать необходимые флажки.

Автоформатирование можно отменить, используя соответствующую команду из меню **Правка**. Если же после форматирования осуществлялись какие-либо действия над ячейками, то после этого следует вновь воспользоваться командами **Формат – Автоформат**.

19.1.5. Копирование форматирования

Все, установленные для какой-либо ячейки, средства форматирования можно перенести на другую ячейку, *скопировав* их. Это избавляет от выполнения одних и



тех же операций по выбору средств форматирования для других ячеек. Во время копирования ячейки стандартными приемами копируется все содержимое ячейки и ее формат.

Если необходимо скопировать не все содержимое ячейки, а только ее отдельные элементы, то для этого нужно:

- выполнить команды меню **Правка – Специальная вставка**;
- в раскрывшемся диалоговом окне на вкладке **Вставить** включить переключатель, соответствующий тому элементу формата ячейки, который необходимо копировать. Например, для копирования только элементов форматирования необходимо включить переключатель **Форматы**.

К сожалению за одно обращение к окну Специальная вставка можно скопировать только один элемент. Поэтому для копирования нескольких элементов приходится несколько раз выполнять команду

Правка – Специальная вставка.

Для копирования всех параметров форматирования, которые уже установлены в ячейке, можно использовать еще один прием:

- выделить ячейку с нужными параметрами форматирования,
- щелкнуть по кнопке **Формат по образцу** на панели инструментов. После этого курсор мыши изменится на крестик и рядом с ним появится изображение кисточки. Одновременно вокруг выделенной ячейки дополнительно появится пунктирный бегущий контур;
- щелкнуть по ячейке или выделить интервал, куда нужно копировать параметры форматирования.

При форматировании по образцу копируются все параметры ячейки, в том числе все параметры форматирования чисел, текстовых фрагментов, цвет, обрамление, заливка.

19.2.Операции с большими таблицами

Довольно часто приходится работать с таблицами (создавать, редактировать, печатать), размер которых превышает размер экрана. Если необходимо попеременно работать с данными таблицы, которые не попадают одновременно на экран, то приходится по очереди выводить на экран то один, то другой участок таблицы, что вызывает неудобства. С целью их исключения программой MS Excel предусмотрены специальные средства.

Скрытие столбцов и строк

Иногда при работе с таблицей большого размера необходимо одновременно иметь в поле зрения столбцы, которые находятся на таком удалении друг от друга, что одновременно не попадают на экран. В этом случае можно воспользоваться приемом «скрытия» тех столбцов, которые разделяют интересующие столбцы. После этого разделяющие столбцы исчезают с экрана, позволяя сомкнуться интересующимся столбцам и оказаться одновременно в поле зрения.

Операция скрытия столбцов осуществляется следующим образом:

- выделяются столбцы, подлежащие временному скрытию,



– выполнить последовательность команд меню

Формат – Столбец – Скрыть.

После этого выделенные столбцы на экране скроются, сомкнув таблицу. После выполнения необходимых операций следует восстановить таблицу и показать скрытые столбцы. Это осуществляется последовательностью команд меню:

Формат – Столбец – Отобразить.

Аналогично производится операция скрытия строк. В этом случае вместо второй команды **Столбец** следует выбирать команду **Строка**.

Деление окна

В тех случаях, когда необходимо иметь в поле зрения не удаленные столбцы или удаленные строки, а удаленные *фрагменты таблицы* (столбцы и строки), приемов скрытия столбцов и строк становится недостаточно. В таких случаях используется способ *деления окна*. Используя этот способ можно разделить таблицу на две или четыре части, как это показано на рис.18.1. Каждая из таких частей снабжается собственными полосами прокрутки, что позволяет выводить на экран интересующие фрагменты таблицы из каждой разделенной части.

Для выполнения операции деления необходимо выполнить следующие операции:

- выделить ячейку на границе деления,
- выполнить операции меню **Окно – Разделить**.

| | A | B | C | D | E | F | G | H | I | K | L |
|----|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | | | | |
| 2 | | | | | | | | | | | |
| 3 | | | | | | | | | | | |
| 4 | | | | | | | | | | | |
| 5 | | | | | | | | | | | |
| 6 | | | | | | | | | | | |
| 7 | | | | | | | | | | | |
| 8 | | | | | | | | | | | |
| 9 | | | | | | | | | | | |
| 10 | | | | | | | | | | | |
| 11 | | | | | | | | | | | |

Рис.19.1 Схема деления окна на четыре части.

На рис.19.1 выделена ячейка F5, которая делит окно на четыре области. Выделенная ячейка размещена в верхнем левом углу правой нижней области.

Еще проще делить окно с помощью специальных *делителей* или *вешек разбивки* (в виде маленьких прямоугольников), которые размещены рядом с кнопками полос прокрутки. *Вертикальный делитель* располагается справа от горизонтальной полосы прокрутки, а *горизонтальный* – вверху вертикальной полосы прокрутки. Перемещая эти делители путем *перетаскивания* в нужные



еста, можно таким образом делить окно на две или четыре части. позиционирования делителя необходимо:

- *совместить с ним указатель мыши таким образом, чтобы делитель преобразовался в двунаправленную стрелку,*
- *нажать левую клавишу мыши и, не отпуская ее, перемещать делитель до предполагаемой границы деления,*
- *отпустить клавишу мыши.*

Для снятия деления необходимо выполнить операции с пунктами меню

Окно – Снять разделение

или с помощью мыши переместить делители на исходные позиции (в полосы прокрутки).

Закрепление областей

При работе с большими таблицами бывает удобно, чтобы во время прокрутки основной части таблицы заголовки столбцов, строк или других ее областей не прокручивались, а оставались в поле видимости. Для таких случаев программой MS Excel предусмотрена операция *закрепления областей*. Она позволяет при просмотре содержимого ячеек, находящихся в отдаленных областях таблицы, не терять из виду их обозначения, которые при прокрутке в обычном режиме исчезают из поля зрения.

Чтобы закрепить нужные области таблицы необходимо:

- *выделить ячейку на границе закрепляемых областей, как и в случае деления окна,*
- *выполнить последовательность команд **Окно – Закрепить области**.*

После выполнения этих действий при производстве прокрутки основной части таблицы *верхняя и (или) левая область окна не прокручивается.*

Для снятия режима закрепления областей необходимо выполнить команды:

Окно – Снять закрепление областей.

19.3.Операции с рабочими листами

Программой MS Excel предусматривается ряд операций с листами. К ним относятся *переименование* листов, *выделение группы* листов, *копирование, перемещение и удаление* листов, *изменение порядка расположения.*

Переименование листов

По умолчанию в новой создаваемой книге из трех листов устанавливаются их названия *Лист1, Лист2, Лист3*. Названия эти нейтральные и не вызывают никаких ассоциаций о содержании таких листов. При желании эти названия можно заменять на более содержательные. Для этого следует:

- *дважды щелкнуть по ярлыку переименовываемого листа* или
- *выбрать команду **Переименовать** из контекстного меню* или
- *выполнить последовательность команд главного меню*

Формат – Лист – Переименовать;

Выполнение одной из этих операций приводит к тому, что ярлык выделенного листа покрывается темным фоном.



- установить указатель мыши в поле ярлыка, при этом указатель мыши превращается из стрелки в текстовый курсор;
- обычным приемом редактирования текста изменить название,
- нажать клавишу **Enter**.

Выделение группы листов

Для выделения группы смежных листов необходимо:

- щелкнуть мышью по первому листу выделяемой группы,
- нажать клавишу **Shift**,
- не отпуская клавиши **Shift**, щелкнуть по ярлыку последнего листа группы.

После этого серый фон ярлыков выделенной группы листов заменится на белый.

Для выделения группы несмежных листов необходимо при нажатой клавише **Ctrl** щелкать по ярлыкам выделяемых листов.

Для отмены выделения необходимо щелкнуть по ярлыку любого, не входящего в выделяемую группу листа, а если выделены все листы книги, то достаточно щелкнуть по любому ярлыку уже выделенной группы.

Копирование листов

Для копирования листа или группы листов следует:

- выделить лист (группу листов),
- выполнить команды основного меню **Правка – Копировать** или из контекстного меню выбрать команду **Копировать** или щелкнуть по кнопке **Копировать** на панели инструментов,
- из раскрывшегося окна выбрать книгу, в которую следует помещать копию листа (группы листов) и ее место в книге..

Перемещение листов

Для перемещения листа или группы листов следует:

- выделить лист (группу листов),
- выполнить команды основного меню **Правка – Переместить** или из контекстного меню выбрать команду **Переместить**,
- из раскрывшегося окна выбрать книгу, в которую следует переместить лист (группу листов) и его (их) место в книге.

Удаление листов

Для удаления листа или группы листов следует:

- выделить лист (группу листов),
- выполнить команды основного меню **Правка – Удалить лист**.

Добавление листов

Для добавления нового листа следует:

- выполнить команды основного меню **Правка – Лист**.

Изменение порядка расположения листов

При необходимости изменения порядка расположения листов в книге необходимо:



- установить указатель мыши на ярлык листа, который след переместить на другое место так, чтобы указатель мыши принял форму стрелки,
- нажать левую клавишу мыши. При этом стрелка примет форму стрелки с листом бумаги и закрашенным треугольником вверху справа;
- не отпуская левую клавишу мыши, перемещать мышшь так, чтобы треугольник расположился на новом месте,
- отпустить клавишу мыши.

19.4. Печать

Созданный в среде MS Excel документ можно распечатать. Напечатать можно

- книгу,
- отдельный рабочий лист,
- выделенный фрагмент рабочего листа,
- область печати, представляющий собой зафиксированный специальным образом фрагмент рабочего листа.

В отличие от фрагмента, область печати сохраняется при сохранении документа. Для создания области печати листа необходимо выполнить следующие действия:

- выделить фрагмент рабочего листа любым известным способом,
- выполнить последовательность команд главного меню

Файл – Область печати – Задать.

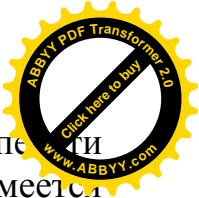
После выполнения этих команд вокруг выделенного фрагмента появится пунктирный контур. Это означает, что область печати создана.

Для снятия режима области печати необходимо выполнить команды:

Файл – Область печати – Убрать.

Перед печатанием необходимо выполнить подготовку к печати. Она, как и в редакторе MS Word, сводится к установке необходимых параметров (формат листа, на котором предполагается печать, ориентация листа, величина полей и др.). Все это выполняется на вкладке **Параметры страницы** окна **Файл**.

- На вкладке **Страница** с помощью переключателей следует выбрать ориентацию листа (*книжная, альбомная*), в раскрывающемся списке **Размер бумаги** выбирается один из нескольких стандартных форматов листа, в поле ввода **Номер первой страницы** указывается номер первой страницы, с помощью элементов управления можно установить масштаб печатаемого документа, качество печати.
- На вкладке **Поля** выбираются размеры полей (*верхнее, нижнее, правое, левое*), *верхнего* и *нижнего* колонтитула. На этой же вкладке имеются переключатели, с помощью которых можно центрировать на странице (*горизонтально, вертикально*) печатаемый фрагмент. В центре вкладки располагается макет страницы, иллюстрирующий текущие установки.
- На вкладке **Колонтитулы** размещаются элементы управления, обеспечивающие создание колонтитулов.
- На вкладке **Лист** находится группа флажков **Печать**, определяющие режим печати сетки (*сетка, черновая*) и заголовков таблицы (*заголовки строк и*



столбцов). Список **Примечания** определяет необходимость и место печати примечаний к ячейкам таблицы. В нижней части окна вкладки имеется группа переключателей **Последовательность вывода страниц**, которыми следует пользоваться в случае, если печатаемый документ занимает несколько страниц. В таком случае необходимо задавать порядок печати: **вниз затем вправо** или **вправо затем вниз**.

После завершения всех установок следует перейти в режим предварительного просмотра, чтобы оценить окончательный вид подготовленного к печати документа. Если его формат удовлетворяет, то следует переходить к завершающему этапу – печати документа. Этот процесс можно выполнить несколькими способами:

- щелкнуть мышью по кнопке **Печать** на панели инструментов,
- нажать комбинацию клавиш **Ctrl + P**,
- выполнить команды **Файл – Печать**.

В последнем случае откроется окно, в котором помощью группы переключателей **Вывести на печать** (*выделенный диапазон, выделенные листы, всю книгу*) обеспечивается вывод на печать соответствующего объекта.



ЛЕКЦИЯ № 20

20. ПОСТРОЕНИЕ ГРАФИКОВ В MICROSOFT EXCEL

Пример 1.

$$y = \sqrt[3]{x^2(x+3)}$$

Определить функцию у.

Для этого в диапазон A1A21 необходимо ввести значение аргумента при помощи автозаполнения, в данном случае с шагом 0,5. В ячейку B1 заносим формулу

$$= (A1^2*(A1+3))*(1/3)/$$

выделим диапазон A1A21, воспользуемся мастером диаграмм и выберем точечную диаграмму со значениями, соединяющимися сплошными линиями без маркеров. Чтобы график получился выразительным, необходимо подписать название графика, оси, по возможности увеличить толщину линий.

Пример 2.

Построить график функции.

$$y = \frac{4x^2 + 5}{4x + 8}$$

Для данного графика определить область определения.

$$4x+8 \neq 0$$

$$x \neq -2$$

При $x=-2$ функция неопределенна. Значение аргумента задаем в два этапа. Например, в диапазоне [-3; 3] сначала 0, 2 точка (-2) будет выколота.

В ячейке B1 формула: ...

Пример 1.

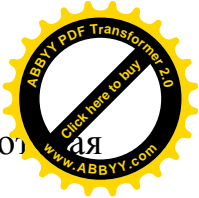
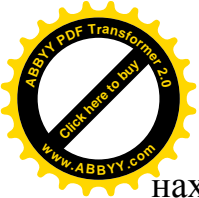
Построить график функции

$$y = \begin{cases} 1+x, & x < 0 \\ e^x, & x \in (0;1) \\ x^2, & x \geq 1 \end{cases}$$

$$y=1+x^2$$

x принадлежит [-3; 3]

| | A | B | C |
|---|----------------|---------|---|
| 1 | x | x | |
| 2 | -3 | =1+A2^2 | |
| 3 | F2+0.5- 2.5 | | |
| 4 | | | |



При построении данного графика воспользуемся функцией «если», которая находится в категории «логические».

ЕСЛИ (A2<0; 1+A2; (ЕСЛИ (A2>=1, A2^2, EXP (A2)))).

Пример 2.

Построить линию заданную неявно функцией.

$$4y^2 + 5x^2 - 20 = 0$$

$$f(x; y) \approx 0$$

$$\frac{y^2}{5} + \frac{x^2}{4} = 1$$

Так как данная функция заданна неявно, разрешим данное уравнение относительно у.

$$4y^2 = 20 - 5x^2$$

$$y^2 = \frac{1}{4}(20 - 5x^2)$$

$$y = \pm \frac{1}{2} \sqrt{(20 - 5x^2)}$$

$$f(x1) = \frac{1}{2} \sqrt{(20 - 5x^2)}$$

$$f(x2) = -\frac{1}{2} \sqrt{(20 - 5x^2)}$$

$$20 - 5x^2 \geq 0$$

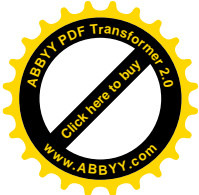
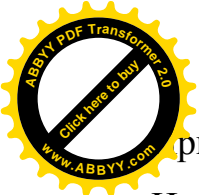
$$-5x^2 \geq -20$$

$$x^2 \leq 4$$

$$x \leq 2$$

$$[-2; 2]$$

| | A | B | C | D | E | F |
|---|------|--------------------|---------------------|---|---|---|
| 1 | X | f(x1) | f(x2) | | | |
| 2 | -2 | =0,5*((20-5*A2^2)) | =-0,5*((20-5*A2^2)) | | | |
| 3 | -1,8 | | | | | |
| 4 | ... | | | | | |
| 5 | ... | | | | | |
| 6 | ... | | | | | |
| n | 2 | | | | | |



пример 3.

Изобразите линию, заданную

$$\frac{x^2}{4} - \frac{y^2}{9} = 1$$

$$-\frac{y^2}{9} = 1 - \frac{x^2}{4}$$

$$y^2 = 9\left(\frac{x^2}{4} - 1\right)$$

$$y = \pm 3\sqrt{\frac{x^2}{4} - 1}$$

Для нахождения интервала находим ОДЗ корня.

$$\frac{x^2}{4} - 1 \geq 0$$

$$x^2 - 4 \geq 0$$

$$x^2 \geq 4$$

$$x \geq \pm 2$$

$$(-\infty; -2] \cup [2; \infty)$$

Построение поверхности

Построим верхнюю часть эллипсоида.

$$\frac{x^2}{9} + \frac{y^2}{16} + \frac{z^2}{4} = 1$$

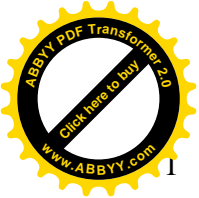
Разделим данное уравнение относительно переменной z.

$$\frac{z^2}{4} = 1 - \frac{x^2}{9} - \frac{y^2}{16} \quad (1)$$

$$z^2 = 4\left(1 - \frac{x^2}{9} - \frac{y^2}{16}\right) \quad (2)$$

$$z = \pm 2\sqrt{1 - \frac{x^2}{9} - \frac{y^2}{16}} \quad (3)$$

Так как по условию нужно найти верхнюю часть эллипсоида рассмотрим ОДЗ положительной части уравнения.



$$1 - \frac{x^2}{9} - \frac{y^2}{16} \geq 0$$

$$-\frac{x^2}{9} \geq \frac{y^2}{16} - 1$$

$$x^2 \leq 9 \left(1 - \frac{y^2}{16} \right)$$

$$x = \pm 3 \sqrt{1 - \frac{y^2}{16}}$$

$$-\frac{y^2}{16} \geq \frac{x^2}{9} - 1$$

$$y^2 \leq 16 \left(1 - \frac{x^2}{9} \right)$$

$$y \leq \pm 4 \sqrt{1 - \frac{x^2}{9}}$$

$$x \leq \pm \frac{3}{4} \sqrt{16 - y^2}$$

$$y \in [-4; 4]$$

$$y \leq \pm \frac{4}{3} \sqrt{9 - x^2}$$

$$x \in [-3; 3]$$

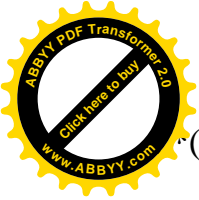
Приступим к построению поверхности.

| | A | B | C | ... | ... | ... | ... | I |
|-----|----|--------------|----|-----|-----|-----|-----|---|
| 1 | | -4 | -3 | | | | | 4 |
| 2 | -3 | формула 3 | | | | | | |
| 3 | | | | | | | | |
| ... | | | | | | | | |
| ... | | | | | | | | |
| ... | | | | | | | | |
| ... | | | | | | | | |
| ... | | | | | | | | |
| 14 | 3 | | | | | | | |

В диапазон В1:J1 введем значение переменной у, а в диапазон А2:А14 – значение переменной х.

Шаг по х и по у может быть любой.

В ячейку В2 введем формулу 3.



$$\sqrt[0,5]{(1-(A2^2)/9-(B1^2)/16)}$$

Знак \$, стоящий перед буквой в имени ячейки дает абсолютную ссылку на столбец с данным именем, а стоящий перед цифрой – на строку с именем, поэтому перед копированием формулы из ячейки B2 в ячейки диапазона B2:U14, в них будет найдено значение Z при соответствующих значениях x и y.

После вычисления в ячейках строим заданную поверхность, используя мастер диаграмм категории «Поверхность».

ЛЕКЦИЯ №21

21. БАЗЫ ДАННЫХ. ОСНОВНЫЕ ПОНЯТИЯ И ОПРЕДЕЛЕНИЯ

Настоящее время – это время информационного всплеска, время развития информационных технологий. Информация всегда была и остается важнейшим фактором для организации деятельности отдельного человека и общества в целом. Тот, кто владеет более полной информацией, тот имеет большие предпосылки для успешной деятельности. Для того чтобы оперативно пользоваться информацией необходимо ее *хранить, извлекать, пополнять и изменять*. Эффективное выполнение последних трех требований возможно в случае *рациональной организации* хранимой информации, то есть ее *систематизации*.

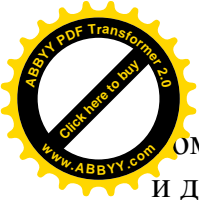
Систематизация – это группирование информации по удобным для пользователя признакам.

Совокупность систематизированной информации называют **базой данных**.

Обычно базы данных создают для большого объема информации, поскольку малый объем информации не вызывает больших затруднений при работе с нею. Каждый человек, занимающийся каким – либо видом деятельности, накапливает информацию в области его профессиональных интересов. Сведения, относящиеся к этой области, называют *предметной областью*. Как правило, каждая база данных имеет свою предметную область, то есть она ориентирована на свой круг вопросов и содержит информацию по определенному виду объектов. Например, вся информация об электродвигателях различного типа.

По принятой терминологии группы объектов, о которых содержится информация в базе данных, называют *сущностями*.

До появления вычислительной техники информация хранилась на бумажных носителях в виде книг, отчетов и т.п. Такого вида информация также систематизировалась по определенным признакам (математика, физика, электротехника и др.) и размещалась в определенных местах (комнаты, стеллажи, полки и др.). Но даже с систематизированной таким образом информацией работать было весьма затруднительно, так как это требовало больших затрат времени на поиск и пополнение информации, больших средств на строительство



помещений, на приобретение оборудования, поддержания климатических усл
и др.

С появлением компьютеров и современных носителей информации большинство из отмеченных трудностей было исключено, но появились другие проблемы, связанные с поиском информации, внесением в нее изменений, в условиях ее значительно возросшего объема. Таким образом, возникла задача создания средств, которые обеспечивали бы *быстрый поиск* информации по заданным признакам, *быстрый доступ* к ней, возможность ее *изменять, пополнять, удалять*. Решением этой задачи стала разработка *инструментальных пакетов программ*, которые получили название **систем управления базами данных (СУБД)**. Не следует смешивать понятия *базы данных* и *системы управления базой данных*. База данных – это, как уже было отмечено, *совокупность систематизированной информации*, а система управления базой данных – это *совокупность программ*, обеспечивающих работу с базой данных (поиск, изменение, дополнение, удаление информации).

Любая рассматриваемая *сущность* (объект, явление) характеризуется своими определенными свойствами. Эти свойства называют *атрибутами*. В общем, любая сущность может иметь *не один, а несколько* атрибутов. Например, электродвигатель имеет такие атрибуты как *мощность, напряжение, ток, момент и др.* В электроприводе эти атрибуты называются *параметрами*, не так как принято в базах данных, но суть от этого не меняется.

В базах данных различают два вида объектов (сущностей):

- *однородные,*
- *неоднородные.*

Однородными называют объекты, которые характеризуются *одним и тем же перечнем атрибутов*. Например, электродвигатели характеризуются такими атрибутами как мощность, напряжение, ток, частота вращения и др. Значение этих атрибутов для различных электродвигателей могут быть различными, но они являются свойством каждого электродвигателя.

Неоднородными называют объекты, которые имеют *различные наборы атрибутов*. Например, электродвигатель и трансформатор. И у электродвигателя и у трансформатора есть такие атрибуты как мощность, напряжение, ток. В то же время у электродвигателя есть движущиеся элементы (якорь, ротор) и поэтому он имеет еще и такой атрибут как частота вращения, а у трансформатора нет движущихся частей, это статический объект, и поэтому у него нет такого атрибута. Поскольку у этих двух объектов различный набор атрибутов, то их и следует относить к неоднородным объектам.

Не следует смешивать понятия объект и сущность. Сущность – это *обобщенный представитель группы однородных объектов* в базе данных (группы объектов). Это *элемент информационной модели*, определяющей структуру базы данных, и является «информационным» термином. Объект – *это конкретное изделие* (электродвигатель, трансформатор и др.) *или явление* (ветер, приливы, отливы и др.).



21.1. Форма хранения информации

Существуют различные схемы, определяющие способ хранения информации. Наибольшее распространение получила система размещения информации в *табличном виде*. При таком способе хранения вся информация делится на несколько групп однородных объектов, каждая из которых размещается в отдельной таблице. Это таблицы связаны между собою.

Системы управления базами данных, использующие способ хранения информации в виде *связанных между собою таблиц*, называют **реляционными**.

Вообще—то табличная форма представления и хранения информации является весьма удобной и поэтому получила широкое распространение. Примером тому могут быть текстовый редактор MS Word, электронные таблицы MS Excel. Однако различия между текстовым редактором и электронными таблицами с одной стороны и системами управления базами данных с другой стороны существенны.

Таблицы текстового редактора MS Word предназначены для наглядного представления текстовой или числовой информации, а автоматизация вычислений и поиска данных у них развита слабо.

Таблицы программы MS Excel служат в основном для автоматизации вычислений.

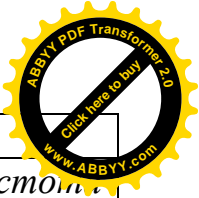
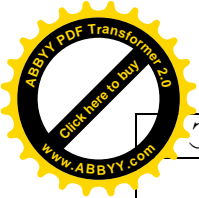
Основное же назначение таблиц базы данных – это *обеспечение эффективного и автоматизированного поиска необходимых данных* среди большого их объема и многообразия.

12.2. Реляционная модель базы данных

Реляционная модель базы данных представляет собой *совокупность связанных между собой таблиц*. Данные организованы в этих таблицах таким образом, чтобы *обеспечить объединение* разнородной информации, *исключить ее дублирование*, а также *предоставить оперативный доступ* к ней и *эффективное сопровождение* базы данных в целом.

21.2.1. Структура таблиц базы данных

Каждая таблица базы данных содержит информацию о *группе однородных объектов*. Пример такой таблицы представлен рис.21.1.



Электродвигатели

| <i>Код</i> | <i>Тип</i> | <i>Мощность, кВт</i> | <i>Напряжение, В</i> | <i>Ток, А</i> | <i>Частота вращения, Об/мин</i> |
|-------------------------|------------|----------------------|----------------------|---------------|---------------------------------|
| 1 | П31 | 27 | 400 | 67,5 | 2700 |
| 2 | П42 | 39 | 400 | 97,5 | 1800 |
| 3 | П58 | 45 | 400 | 112,5 | 1500 |
| <i>Строка состояния</i> | | | | | |

Рис.21.1 Общий вид таблицы базы данных

Как видно из таблицы, в ней содержатся однородные объекты – электродвигатели. Все они имеют один и тот же перечень параметров (атрибутов).

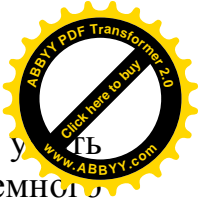
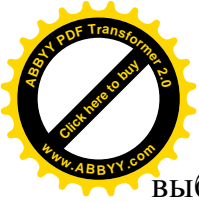
Табличную форму размещения и хранения данных предложил в 1970 г. Э.Ф. Кодд. Он показал, что набор двумерных таблиц, при соблюдении определенных ограничений, позволяет хранить информацию об объектах и моделировать связи между ними. В терминологии Кодда связанные между собой таблицы называются *отношениями* (англ. *relation*). Отсюда и название модели – *реляционная*.

Среди многообразия разработанных СУБД наибольшее распространение получила система управления базами данных, корпорации Microsoft – MS Access.

В терминологии принятой в системах управления базами данных строка таблицы имеет название *запись*, а столбец – *поле*. Каждая таблица СУБД имеет свою специализацию. Она хранит информацию об однородной группе объектов. Например, одна таблица хранит информацию о типах электродвигателей, заводах – изготовителях, стоимости, другая таблица содержит информацию об их электрических параметрах (мощность, напряжение, ток и др.), третья таблица – о конструктивных особенностях (высота, ширина, длина, масса и др.) и т. д. Все эти таблицы связаны между собой, поскольку они только в совокупности дают полную информацию об объектах.

Запись представляет собой *набор значений всех атрибутов информационной модели рассматриваемой сущности*. Проще говоря, запись – это совокупность сведений об объекте. В одной таблице все записи состоят из одних и тех же полей, поскольку таблица формируется из однородных объектов. Каждый объект имеет свой набор значений в этих полях. В некоторых полях значения могут отсутствовать или как принято считать – иметь значение *Null*. Название каждого поля в таблице *уникальное*. То есть одна таблица *не может иметь в своем составе два или более полей с одинаковым названием*.

Типы данных, содержащихся в отдельных полях, могут быть различными. В одних полях информация может представляться в текстовом виде, например, тип двигателя, в других полях – в числовом виде, например, мощность, напряжение, ток. *Размер поля* определяется количеством байт, необходимым для размещения значения атрибута. Иногда вместо термина *размер поля* употребляют термины *ширина* или *длина поля*.



Для получения из базы данных тех или иных сведений необходимо уметь выбирать записи о тех или иных объектах. Если таблица содержит немного записей, да и сами записи имеют небольшое количество полей, то это сделать несложно, рассматривая таблицу на экране. Но если записей тысячи и полей десятки или сотни, то выбирать необходимые сведения представляет огромную трудность и связано с большими затратами времени. В таких случаях *только автоматизация* позволит сделать этот процесс эффективным. Именно такой способ предусматривает СУБД. Для реализации такого принципа поиска информации необходимо указать поле или группу полей, значения которых позволят однозначно выбрать из многообразия записей одну необходимую.

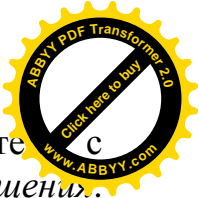
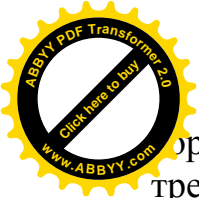
Поле, с помощью которого можно идентифицировать запись, называют *первичным ключом* или просто *ключом таблицы*. Значение *первичного ключа* должно быть *уникальным*, то есть нигде не повторяться. При этом поле ключа не должно содержать значение *Null*.

Если ключ состоит из одного поля, то он считается *простым*, если он состоит из нескольких полей, то его называют *составным*. В тех случаях, когда невозможно выбрать первичный ключ, когда значения всех полей двух или более записей одинаковые, используется *дополнительное поле*, куда может помещаться уникальный код, например, порядковый номер. В таком поле двух одинаковых порядковых номеров не может быть. Поля, содержащие порядковые номера строк, относят к типу *счетчик*.

21.2.2. Индексы базы данных

Одним из приемов ускорения поиска информации в системах управления базами данных является расположение информации в *упорядоченном* виде. Например, расположение электродвигателей в таблице по возрастанию мощности или другого параметра, размещение фамилий сотрудников в алфавитном порядке и т. п. Такое упорядочение называют *сортировкой*. Для выполнения действий по упорядочению информации требуется дополнительное время и память компьютера. Если информация в базе данных постоянно изменяется (что-то удаляется, что-то добавляется), то это требует частой ее сортировки и дополнительной потери времени. В таких случаях предпочитают использовать несортированную таблицу данных. Но тогда возрастает время поиска информации.

Для достижения компромисса в этой ситуации вводят *дополнительную таблицу*, называемую *индексами базы данных*. В эту таблицу помещаются *только первичные ключи* из основной таблицы *и поле, по которому совершается сортировка*. Такая таблица с упорядоченными ключами и полем атрибута с упорядоченными значениями представляет собой *индекс* для основной таблицы базы данных. Для одной и той же таблицы данных можно создать несколько индексов. Например, для таблицы, содержащей информацию о параметрах электродвигателей, можно создать один индекс по мощности, другой индекс по напряжению, а третий по частоте вращения и др. Это удобно. Если необходимо найти электродвигатель с необходимой мощностью, то в этом случае удобно использовать *индекс по мощности*, так как поле мощности в этом индексе



сортировано, что обеспечит быстрый поиск. При выборе электродвигателя требуемой частотой вращения удобно использовать *индекс по частоте вращения*. Индексные таблицы значительно меньше основных таблиц и на их сортировку требуется существенно меньше времени. Этим и определяется полезность их применения.

Каждому созданному индексу следует устанавливать имя, чтобы можно было легко выбрать объект по интересующему атрибуту.

В общем случае создание индексов может осуществляться по значениям нескольких полей. Такой индекс называют *составным*. Это имеет смысл, если в отдельных полях размещаются, например, фамилия, имя и отчество сотрудника. При создании индекса по полю, содержащему фамилии, может возникнуть ситуация, когда при наличии нескольких одинаковых фамилий последующие поля (имя и отчество) будут подсоединены к выбранной фамилии в произвольном порядке. При наличии составного индекса по фамилии, имени и отчеству такой вариант будет исключен.

21.2.3. Связи базы данных

Иногда при работе с базой данных возникает потребность получить не полную, а частичную информацию об интересующем объекте. Например, в базе данных «Университет» имеется полная информация о преподавателях, студентах, дисциплинах, успеваемости, материальной базе и др. Безусловно, вся эта информация связана между собой, образуя своеобразную цепь. Если потащить за одно звено, то можно вытащить всю цепь. То же можно сказать и об информации по электродвигателям.

Полная информация содержит сведения об электрических параметрах, о конструктивных параметрах, о заводах–изготовителях, о стоимости и др. Если все эти сведения размещать в одной *сплошной* таблице, то в случае потребности получения информации только об электрических параметрах интересующего электродвигателя, будет выдана полная информация о нем. Это не всегда нужно. Чтобы исключить такие случаи, всю информацию разделяют на отдельные логически обоснованные части и для каждой из этих частей создают отдельную таблицу. Реально база данных должна хранить информацию о нескольких взаимосвязанных сущностях, образующих предметную область, то есть быть представленной *набором связанных таблиц*. Эта связь должна быть синхронной. Это означает, что если удаляется, например, тип двигателя в одной таблице, то удаляется вся связанная с ним информация во всех таблицах и наоборот, если добавляется в базу данных какой–то двигатель, то автоматически во все таблицы добавляется соответствующая информация о нем.

При проектировании таблиц и определении связей между ними используют способ *нормализации*. Этот способ позволяет разделить исходную, сплошную таблицу на ряд элементарных таблиц, между которыми устанавливаются связи. Такая совокупность связанных таблиц создает единую информационную цепь



объекта. Вместе с тем, из этой цепи можно выбирать частичную информацию об объекте, хранимую в отдельной таблице. В то же время связи позволяют осуществить и обратный процесс «сборки» информации по информации одной из таблиц.

Нормализация – это пошаговый процесс замены одной таблицы (или набора таблиц) другими, имеющими более простую структуру. Способ нормализации опирается на теорию множеств и реляционной алгебры. На каждом этапе нормализации таблицы приводятся к виду, который называется *нормальной формой*.

Существует четыре вида связей (отношений между таблицами):

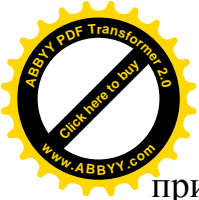
- *один к одному,*
- *один ко многим,*
- *многие к одному,*
- *многие ко многим.*

Связь **один к одному** означает, что *каждой записи первой таблицы соответствует только одна, связанная с ней запись второй таблицы и наоборот*. Такой тип отношений используется крайне редко, так как фактически все данные могут быть помещены в одной таблице. Он полезен тогда, когда одну громоздкую таблицу целесообразно разделить на две с логически сгруппированными полями. Например, в одной таблице разместить электрические параметры электродвигателя, а в другой – его конструктивные параметры. Такое деление используют тогда, когда в одну таблицу выносятся секретные или очень важные данные с ограниченным доступом.

Связь **один ко многим** характерна тем, что *запись одной таблицы связана с несколькими записями другой таблицы*. В то же время запись второй таблицы не может быть связана более чем с одной записью первой таблицы. Например, один покупатель приобретает в магазине несколько различных товаров, но один конкретный экземпляр товара не может быть продан одновременно нескольким покупателям (купит кто–то один).

Связь **многие ко многим** или *непрямая табличная связь* определяет *связь одной записи первой таблицы с несколькими записями второй таблицы, в то же время как одна запись второй таблицы может быть связана с несколькими записями первой таблицы*. На практике такой сложный вид связи между двумя таблицами реализуется через *промежуточную (связующую) таблицу*. Она позволяет заменить одну связь вида *многие ко многим* на две последовательные связи: *многие к одному* и *один ко многим*, которые реализуются проще.

Из двух связанных таблиц одна выступает в роли *главной (первичной)*, а другая – в роли *подчиненной (вторичной)*. Например, таблица **Дисциплины** содержит перечень дисциплин, которые преподаются различными преподавателями. Сведения о преподавателях содержатся в другой таблице **Преподаватели**. Здесь таблица **Дисциплины** играет роль главной, а таблица **Преподаватели** – подчиненной.



На рис.21.2 приведены две таблицы **Дисциплины** и **Преподаватели**. В примере первичным ключом таблицы **Дисциплины** является **Код дисциплины**, а первичным ключом таблицы **Преподаватели** – **Код преподавателя**. Для установления связи между этими таблицами в подчиненную таблицу **Преподаватели** добавлено поле *внешнего ключа*, роль которого играет поле **Код дисциплины** таблицы **Дисциплины**.

Из этого рисунка видно, что дисциплину **Информатика** преподают два преподавателя Петров И. С. и Серов Г. Л., так как их коды соответственно 2 и 3 связаны с кодом 2 таблицы **Дисциплины**. Таким образом, реализована связь *один ко многим*, где одна дисциплина **Информатика** таблицы **Дисциплины** связана с двумя преподавателями Петровым И. С. и Серовым Г. Л., таблицы **Преподаватели**.

Конкретный набор всех таблиц, необходимых для хранения и поиска информации, образует *структуру базы данных*. Разработка и создание структуры базы данных представляет собой самый важный и самый сложный этап. Эту работу, как правило, поручают высококвалифицированным специалистам в этой области.



Главная таблица

| Дисциплины: таблица | | | | |
|---------------------|-------------|---------|-------------------------|----------------------------|
| Код дисциплины | Дисциплина | Семестр | Количество лекций, час. | Количество лаб. раб., час. |
| 1 | Математика | 1 | 64 | – |
| 2 | Информатика | 1 | 32 | 32 |
| (счетчик) | | | | |

Поле первичного ключа таблицы **Дисциплины**.

Связь

Подчиненная таблица

| Преподаватели: таблица | | | | | |
|------------------------|-------------------|-------------|-----------|----------------|------|
| Код дисциплины | Код преподавателя | Фамилия | Должность | Ученая степень | Стаж |
| 1 | 1 | Иванов А.В. | Профессор | д.ф-м.н. | 25 |
| 2 | 2 | Петров И.С. | Доцент | к.т.н. | 19 |
| 2 | 3 | Серов Г. Л. | Доцент | к.т.н. | 15 |
| | (счетчик) | | | | |

Внешний ключ

Первичный ключ таблицы **Преподаватели**

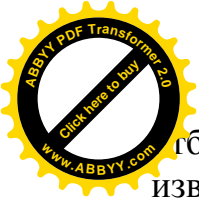
Рис. 21.2 Пример связи типа *один – ко – многим*

21.3. Основные виды работы с базами данных

После разработки структуры базы данных, где определяется количество таблиц и их содержание, а также связи между ними, приступают к заполнению этих таблиц информацией. Вводить данные можно непосредственно в таблицы базы данных. Однако значительно удобнее это делать, используя так называемые **формы**. Они представляют собой *диалоговые окна*, содержащие *поля ввода*, которые служат для отображения и изменения данных в полях записей. Дополнительно в этих окнах содержатся различные переключатели, флажки, кнопки. С помощью них можно управлять процессом работы с окнами. Отличительной особенностью работы с формами является то, что в них отображаются *поля только одной текущей записи*, в то время как при работе с таблицами в них отображаются *одновременно поля нескольких записей*. Работа с формой существенно упрощает процесс ввода данных. С помощью форм можно не только производить ввод первичной информации, но впоследствии и изменять ее.

Для того, чтобы получить необходимую информацию из базы данных, необходимо организовать *запрос*.

Запросом в базу данных называют предложение, содержащее требование на отбор данных, хранящихся в таблицах, или требование на выполнение определенных действий над данными. Обычно такое требование предполагает



Выбор информации по определенному критерию. По запросу данные могут извлекаться как с одной, так и с нескольких различных таблиц. Результатом выполнения запроса является *сводная таблица*, в которой представляются все интересующие данные. Такая таблица может содержать не только значения атрибутов, но и вычисляемые поля.

Информация, получаемая из базы данных, может оформляться в виде *документа готового к печати*. Такая форма представления информации называется *отчетом*. В отчете вся информация представляется в сгруппированном, отсортированном и отформатированном определенным образом виде. Поскольку результаты запросов и отчеты в большинстве своем оформляются в виде таблиц, то, в отличие от них, таблицы, содержащие первичную информацию и составляющие базу данных, называют еще *основными таблицами*.

Эксплуатация базы данных состоит не только в выборке информации из нее, но и в поддержании базы данных в рабочем состоянии. Эти функции выполняет *администратор базы данных*. Функции администратора обычно выполняет высококвалифицированный в этой области специалист.

ЛЕКЦИЯ 22

22. СИСТЕМА УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ MS ACCESS

22.1 . Назначение и возможности СУБД MS Access

Программа MS Access – это одна из распространенных систем управления базой данных, входящая в состав пакета программ Microsoft Office.

СУБД MS Access предназначена для создания и поддержания в рабочем состоянии баз данных, для получения из них информации в требуемом виде, для создания приложений, автоматизирующих работу с базами данных.

База данных по версии MS Access представляет собой совокупность взаимосвязанных ее компонентов, называемых *объектами базы*. Объектами базы данных считаются:

- *таблицы данных,*
- *формы,*
- *запросы,*
- *отчеты,*
- *страницы,*
- *макросы,*
- *модули.*

Таблица – это форма представления информации. Ее устройство рассматривалось ранее.

Форма – это диалоговое окно, с помощью которого можно вводить информацию в таблицы, редактировать и удалять ее, а также ограничивать доступ к данным и отображать их только в режиме просмотра.

Запрос – инструмент управления данными, позволяющий извлекать из таблицы базы данных необходимые сведения, отвечающие заданному критерию. С



С помощью запросов можно автоматизировать процесс обновления и удаления записей, а также выполнять вычисления, используя значения данных, размещенных в таблицах.

Отчет – это форма представления информации подготовленной к печати в виде отформатированных данных, результатов итоговых расчетов, графиков, диаграмм, проще говоря, в форме удобной для восприятия и анализа.

Страница – это документ, содержащий сведения из базы данных, и предназначенный для размещения в Internet (веб–страница). Такой документ играет роль своеобразного интерфейса к данным. Обращаясь к таким страницам через Internet, можно просматривать и вводить информацию в таблицы, а также создавать отчеты.

Макрос и модуль – инструмент автоматизации работы с базами данных и создания приложений, необходимых для работы с базами данных.

Программа MS Access позволяет создавать в базе данных до 32768 различных объектов. Все объекты базы данных хранятся в *одном общем файле базы данных*, который имеет расширение *.mdb*. Эта программа достаточно ресурсоемкая. Так версия MS Access 2002 требует 128 Мгбайт оперативной памяти.

22.2. Запуск и завершение работы программы MS Access

В стандартном виде *запуск программы* может осуществляться одним из следующих способов:

- *щелчком мыши по ярлыку MS Access на рабочем столе,*
- *с помощью последовательности команд рабочего стола*

Все программы – Microsoft Access

Завершение работы с программой MS Access производится одним из способов:

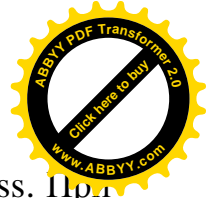
- *нажатием комбинации клавиш **Alt + F4**,*
- *командами **Файл – Выход**,*
- *щелчком по кнопке окна*

Интерфейс программы MS Access имеет весьма схожий вид, что и интерфейс текстового редактора MS Word и электронных таблиц MS Excel.

23.3. Типы данных

В СУБД MS Access поддерживается 10 основных типов данных:

- *текстовый,*
- *МЕМО,*
- *числовой,*
- *денежный,*
- *дата/время,*
- *счетчик,*
- *логический,*
- *объект OLE,*
- *гиперссылка,*



– мастер подстановок.

Текстовый тип – наиболее часто используемый тип в СУБД MS Access. При создании новых полей он принимается *по умолчанию*. Информация текстового типа представляется совокупностью любых символов алфавита, включая числа. Если в текстовом поле помещается число, то оно воспринимается как текст и поэтому не может участвовать в каких-либо вычислительных операциях. Обычно в такие поля заносятся имена, фамилии, названия объектов, адреса, номера телефонов, почтовые индексы и т.п.

При выборе поля текстового типа необходимо указывать его размер. Так как значения текстовых полей могут представляться различным количеством символов, то при указании размера поля следует представлять его максимально возможный вариант.

Максимальный размер текстового поля составляет 255 символов. Если размер поля не указывается, то он принимается *по умолчанию равным 50 символам*.

В тех случаях, когда размеры текстового поля колеблются в широких пределах, удобнее использовать другой тип текстового поля – **МЕМО**.

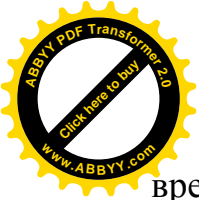
Поле **МЕМО** – это текстовое поле большого размера (до 65536 знаков). В таких полях обычно помещаются комментарии, пояснения, примечания, а также другая информация такого рода. Для данных в полях МЕМО выделяется столько места для записи значения, сколько потребуется. К примеру, для одной записи 2000 позиций, а для другой – 30.

Числовой тип представляется *числами*, которые могут быть использованы в различных вычислительных операциях. Числовой тип имеет несколько подтипов (диапазонов и форм представления чисел), приведенных в табл. 22.1.

Таблица 22.1 Диапазоны чисел.

| Название подтипов | Десятичные Знаки | Диапазон значений | Размер памяти, байт | |
|----------------------------|------------------|--|---------------------|---|
| <i>Байт</i> | Нет | 0...255 (2^8) | 1 | |
| <i>Целое</i> | Нет | -32768...32767 ($-2^{16} \dots 2^{16}-1$) | 2 | |
| <i>Длинное целое</i> | Нет | $-2^{32} \dots 2^{32}-1$ | 4 | |
| Одинарное плавающей точкой | с | 7 | - 3.4E38...3.4E38 | 4 |
| Двойное плавающей точкой | с | 15 | - 1.8E308...1.8E308 | 8 |
| <i>Действительное</i> | 28 | - 1.0E-28... 1.0E28 - 1 | 14 | |
| <i>Денежный</i> | 4 | - 922337203685477.5808 ... + 922337203685477.5807 | 8 | |
| <i>Код репликации</i> | Нет | Не применяется | 16 | |

Денежный тип – это разновидность данных числового типа. Он используется для отображения цен, сумм, счетов и т.п. Этот тип отличается от числового типа тем, что исключает округление чисел во время вычислений.



Дата/время – это тип, предназначенный для представления значений даты, времени или обе эти величины одновременно. В MS Access предусмотрено несколько форматов для таких полей. Он занимает 8 байт памяти.

Счетчик – это уникальный идентификатор, предназначенный для нумерации записей. Последовательность номеров записи может быть как с *шагом единица*, так и *случайными числами*. Тип *счетчик* может быть только у одного поля таблицы, и оно обычно используется в качестве ключевого. Занимает этот тип 4 байта памяти.

Логический тип. Поля такого типа могут содержать информацию в виде значений переменной логического типа: *да*, *нет* или *1*, *0*. Значение *Null* не допускается.

Поле объекта OLE (Object Linking and Embedding – технология связывания и внедрения) предназначено для размещения рисунков, электронных таблиц, диаграмм, аудио – и видео – фрагментов, а также других данных в двоичном формате из других приложений, поддерживающих технологию *OLE*. Занимает объем памяти до 1 Гбайт.

Гиперссылка обеспечивает связь с Web – страницей, расположенной в Internet или Intranet (внутренняя, локальная сеть) или на локальном компьютере. Позволяет переходить из текущего поля к информации в другом файле. При выборе такого поля программа MS Access автоматически запускает Web – страницу и отображает указанную страницу.

Мастер подстановок – это *не тип* данных, а *свойство* поля. Используется в MS Access для более эффективного и корректного ввода данных. При выборе этого типа запускается мастер подстановок, который создает поле, позволяющее выбрать значение из другой таблицы или списка. Как правило, в ходе операции подстановки выполняется запрос к другой таблице для получения подстановочных данных. Занимает память 4 байта.

22.4. Разработка и создание базы данных

Разработка базы данных начинается с разработки ее структуры. Это весьма сложный процесс, требующий хорошего знания предмета и значительного опыта.

22.4.1. Разработка структуры базы данных

Процесс разработки структуры базы данных состоит из нескольких этапов.

На **первом этапе** определяются все необходимые атрибуты объекта, которые составили бы полную информацию о нем. Затем создается исходная (*сплошная*) таблица *в универсальной форме*. Это означает, что каждое поле такой таблицы должно иметь *только одно (атомарное) значение*.

На **втором этапе** выполняется процесс *нормализации* таблицы. Он заключается в разбиении исходной таблицы на несколько меньших по объему связанных таблиц. Существует несколько методов нормализации. Если число полей исходной таблицы не превышает 20, то чаще всего используют метод нормализации Бойса – Кодда. Он позволяет однозначно установить минимальное количество таблиц с минимальным набором полей (атрибутов). Изучение этого и



Других методов нормализации требует значительного количества времени. В настоящем изложении дисциплины не рассматривается. Основной акцент будет сделан на приемы работы с уже созданной структурой базы данных.

После определения количества, назначения и состава таблиц следует *определить для каждого поля таблицы тип данных*, которые будут в нем храниться, то есть определить размер текстовых полей и подтип числовых данных.

Далее следует *определить связи* между таблицами.

При выполнении этих этапов следует придерживаться следующих рекомендаций:

- стремиться включить в базу данных всю необходимую информацию, относящуюся к предметной области;
- создавать простые и не громоздкие таблицы;
- полезно использовать для составления таблиц уже имеющиеся и успешно используемые в работе формы на бумаге (бланки, анкеты и др.);
- избегать использования имен таблиц и полей, совпадающих с зарезервированными в MS Access словами;
- не следует создавать в таблицах поля, значения которых определяются как результат каких-либо вычислений;
- выбирать имена полей, вызывающие правильные ассоциации об их назначении;
- хорошим стилем считается использование составных названий таблиц и полей. При этом первое слово должно определять сущность информационного объекта, а второе слово или несколько слов – его уточнение. Например, *Мощность_электродвигателя*.

22.4.2. Имена полей и объектов

Имена полей и объектов образуются из совокупности различных символов за исключением символов *точка* (.), *восклицательный знак* (!), *апостроф* (') и *квадратная скобка* ([] ()).

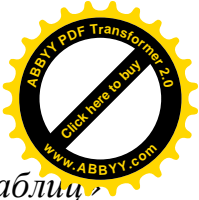
Предельная длина имени составляет 64 символа.

Имя **не должно начинаться с пробела** и **включать управляющие знаки** с кодами ASCII от 0 до 31.

22.5. Создание новой базы данных средствами СУБД MS Access

После разработки базы данных на бумаге приступают к созданию ее электронного варианта, то есть файла базы данных с расширением *.mdb*. Созданный «пустой» файл базы данных уже занимает около 96 Кбайт дискового пространства. Это объясняется тем, что он уже содержит «заготовки» в виде скрытых системных таблиц, в которые в дальнейшем будет заноситься информация о различных объектах создаваемой базы данных. По мере создания новых объектов – таблиц, форм, отчетов и заполнения самой базы данных объем этого файла будет расти.

Базу данных в СУБД MS Access можно создавать двумя способами:



- самостоятельное создание и заполнение таблиц,
- создание с использованием образцов, предлагаемых «мастером таблиц» СУБД MS Access.

–

22.5.1. Создание таблиц базы данных с помощью мастера таблиц

Под созданием базы данных подразумевается создание системы связанных таблиц с именами и заголовками полей, со значениями атрибутов в полях.

После запуска программы MS Access следует *выполнить одну из операций:*

- щелкнуть по кнопке **Создать** на панели инструментов;
- нажать комбинации клавиш **Ctrl + N**;
- выполнить команды главного меню **Файл – Создать**;
- в появившемся окне **Создание** раскрыть вкладку **Общие**, выбрать ярлык **База данных** и щелкнуть по нему дважды;
- в раскрывшемся окне **Файл новой базы данных**, в строке **Имя файла**, расположенной в нижней части окна, указать место размещения файла (путь) и его имя;
- щелкнуть по кнопке **Создать**, расположенной справа от этой строки;
- в левой части раскрывшегося окна **База данных** с именем создаваемой базы данных представляется список объектов (**Таблицы, Запросы, Формы, Отчеты, Страницы, Макросы, Модули, Группы избранное**), из которых по умолчанию выделяется объект **Таблицы**, а в правой части окна представляются способы создания таблиц (**Создание таблицы в режиме конструктора, Создание таблицы с помощью мастера, Создание таблицы путем ввода данных**). Из списка предлагаемых способов создания таблиц следует выбрать желаемый способ.

Для самостоятельного создания базы данных необходимо дважды щелкнуть по ярлыку против строки **Создание таблицы путем ввода данных**. Вследствие этого на экране появится окно в виде пустой таблицы с заголовками: *Поле1, Поле2, ..., Поле10*. После этого следует в верхние ячейки этих полей поместить заголовки, а в последующие ячейки – значения. Таким способом создаются все таблицы базы данных. Однако этот способ требует много времени и напряжения.

Процесс создания базы данных значительно облегчается при использовании *мастера таблиц*. Работа с мастером таблиц предусматривает *создание таблиц* за несколько шагов.

На **первом шаге** создается таблица по одному из образцов, предложенных *мастером таблиц*. При этом мастер таблиц предлагает 25 вариантов таблиц делового применения (**Контакты, Сотрудники, Задачи, Студенты** и др.) и 20 вариантов таблиц личного пользования (**Адреса, Рецепты, Альбомы** и др.).

Первый шаг.

- В раскрывшемся окне **База данных** необходимо дважды щелкнуть по строке **Создание таблицы с помощью мастера** или по ярлыку перед этой строкой;



– в раскрывшемся окне **Создание таблиц** изображаются три дополнительных окна с названиями: **Образцы таблиц**, **Образцы полей** и **Поля новой таблицы**. Из списка в дополнительном окне **Образцы таблиц** следует выделить строку с подходящим названием таблицы, суть которой ближе всего отображает содержание создаваемой таблицы, и щелкнуть по ней;

– в появившемся списке в окне **Образцы полей** необходимо выделить строку с подходящим названием поля и щелкнуть по кнопке **>**. После этого выбранное название поля отобразится в окне **Поля новой таблицы**. Таким способом можно расположить в этом окне и другие названия выбранных полей.

Если какое-либо название поля необходимо *исключить* из окна **Поля новой таблицы**, то его следует выделить, а затем щелкнуть по кнопке **<**. Для отображения всего списка названий полей окна **Образцы полей** в окне **Поля новой таблицы** следует щелкнуть по кнопке **>>**, а для исключения всех полей из списка **Поля новой таблицы** следует щелкнуть по кнопке **<<**.

При необходимости *переименования* названия поля в списке **Поля новой таблицы** следует выделить это название и щелкнуть по кнопке **Переименовать поле...** и в открывшейся строке ввести новое название поля.

После завершения формирования названий полей следует щелкнуть по кнопке **Далее** в окне **Создание таблиц**.

Этой операцией завершается первый шаг работы мастера таблиц и осуществляется переход ко второму шагу.

В окне **второго шага** задается имя для создаваемой таблицы и выбирается способ задания первичного ключа.

Второй шаг.

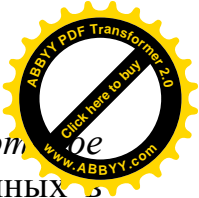
В раскрывшемся окне второго шага следует:

- в имеющейся строке с названием **Задать имя для новой рассылки** установить имя создаваемой таблицы;
- выбрать способ установки первичного ключа (*автоматически, самим пользователем*).

Способ установки первичного ключа в режиме *автоматически* выбирается тогда, когда пользователь затрудняется самостоятельно сделать этот выбор. В таком случае *мастер таблиц* делает это сам. Если пользователь определился с полем первичного ключа еще на стадии проектирования базы данных, то ему следует выбрать вариант установки первичного ключа *самим пользователем*.

- После выбора варианта следует щелкнуть по кнопке **Далее** этого окна.

При выборе второго варианта задания первичного ключа открывается диалоговое окно с названием **Создание таблиц**. В верхней части этого окна изображена строка с названием **Выберите поле с уникальными для каждой записи данными**. Эта строка содержит раскрывающийся список названий полей создаваемой таблицы, выбранных на предыдущем этапе.



- Из раскрывающегося списка необходимо выбрать *название поля, которое будет выполнять функцию первичного ключа* из трех, представленных в этом окне вариантов:
 - *последовательные числа, автоматически присваиваемые каждой новой записи,*
 - *числа, вводимые пользователем при добавлении новой записи,*
 - *сочетание чисел и букв, вводимое пользователем при добавлении новых записей.*
- Щелкнуть по кнопке **Далее** этого окна.

Третий шаг.

Он выполняется, *если в базе данных уже существуют ранее созданные другие таблицы.*

Для этого раскрывается очередное окно, в котором мастер задает вопрос:

Связана ли новая таблица с другими таблицами базы?.

Одновременно формируется список с названиями ранее созданных таблиц. Для каждой из них будет показано *отсутствие связи с новой таблицей.* Если одна из уже созданных таблиц должна находиться с вновь создаваемой таблицей в связи *один ко многим,* то ее название нужно выделить и щелкнуть по кнопке **Связи.**

В новом раскрывшемся окне путем выбора нужного переключателя уточнить характер связи и щелкнуть по кнопке **ОК** этого окна.

Четвертый шаг.

В раскрывшемся окне с названием **Создание таблиц** мастер предлагает три варианта дальнейших действий:

- *изменить структуру,*
- *ввести данные непосредственно в таблицу (устанавливается по умолчанию),*
- *ввести данные в таблицу с помощью формы, создаваемой мастером.*

Далее следует выбрать нужный вариант и щелкнуть по кнопке **Готово** этого окна. При этом на экране появится новая (пустая) таблица, в которую можно вводить данные.

В случае выбора первого варианта происходит переключение в режим *Конструктор,* в котором в произвольной форме можно изменить структуру созданной таблицы.

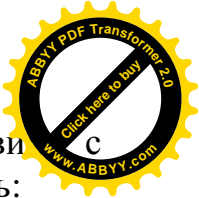
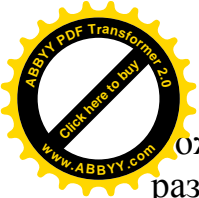
При выборе второго варианта происходит переключение в режим *Таблица.* Это позволяет вводить необходимые данные непосредственно в ячейки ее полей.

В третьем случае мастер автоматически создаст форму удобную для ввода данных в таблицу.

Очевидно, что при создании *первой* таблицы, третий шаг мастера исключается.

22.5.2. Создание таблиц в режиме конструктора

Конструктор – это специальное приложение программы MS Access, которое позволяет создать совершенно новую таблицу базы данных либо добавить, удалить или изменить свойства полей уже существующей таблицы. В режиме конструктора



можно создать структуру будущей таблицы в полном соответствии с разработанным ранее проектом. В этом режиме предоставляется возможность:

- *определить список полей для таблицы,*
- *выбрать тип данных,*
- *настроить свойства для каждого поля,*
- *ввести описание полей.*

Для работы с *конструктором* необходимо в окне **База данных** дважды щелкнуть по строке **Создание таблицы в режиме конструктора** или по ярлыку перед ней. Выход в окно **База данных** осуществляется уже ранее описанным способом. После выполненных действий раскрывается окно с таблицей полей в верхней части вида:

| Таблица 1: таблица | | | |
|--------------------|-------------------|------------|----------|
| | Имя поля | Тип данных | Описание |
| | Код преподавателя | Счетчик | |
| | Фамилия И.О. | Текстовый | |
| | | | |
| | | | |
| | | | |
| | | | |

Эта таблица служит для *определения названий и типов полей* создаваемой таблицы. При переходе в ячейку *Тип данных* по умолчанию в ней появится значение *Текстовый*, а справа в этой ячейке – кнопка с указателем, с помощью которой можно раскрывать список типов этой ячейки. Щелкнув по этой кнопке, можно из раскрывшегося списка типов выбрать необходимый тип данных для этой ячейки, а не набирать его с клавиатуры. После выбора типа данных следует перейти в соседнюю ячейку столбца *Описание*. В этой ячейке требуется поместить краткое и ясное описание поля. Текст описания будет выводиться в строке состояния при вводе данных в поле и может быть удобной подсказкой о допустимых значениях этого поля. Эта операция не является обязательной, но целесообразной.

Таким способом помещаются названия всех полей создаваемой таблицы в ячейках поля *Имя поля*, а в ячейках поля *Тип данных* – соответствующие им типы.

Для каждого поля можно установить не только тип данных, но и другие характеристики. Все они представляются в области **Свойства поля**, располагаемой в нижней части окна.

Эти характеристики используются для задания и изменения *свойств полей* таблицы. Они представляют собой набор параметров, обеспечивающих дополнительные возможности управления хранением данных, их вводом и отображением в поле. Количество доступных свойств зависит от типа данных. В



качестве примера перечень этих параметров вкладки **Общие** представлен ниже с указанием параметров для поля *Фамилия И.О.*

| | |
|-------------------------|-----------------------------|
| Размер поля | 50 |
| Формат поля | |
| Маска ввода | |
| Подпись | |
| Значение по умолчанию | |
| Условие на значение | |
| Сообщение об ошибках | |
| Обязательное поле | Да |
| Пустые строки | Нет |
| Индексированное поле | Да (Допускаются совпадения) |
| Сжатие Юникод | Да |
| Режим IME | Нет контроля |
| Режим предположений IME | Нет |

После завершения формирования *структуры* таблицы следует назначить *первичный ключ*. Для этого следует:

- выделить одну или несколько строк (случай составного ключа) таблицы, соответствующих полям *первичного ключа*;
- щелкнуть по кнопке **Ключевое поле** (с изображением ключа) на панели инструментов MS Access. Слева от заголовков выделенных строк появятся значки с изображением ключа, что символизирует установление этим полям статуса *первичного ключа*.

Если *первичный ключ* выбран неудачно, то процедуру его назначения следует повторить с самого начала.

Завершающим этапом создания структуры таблицы является ее сохранение. Для этого необходимо:

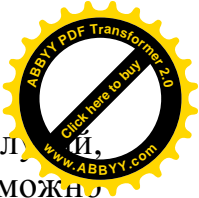
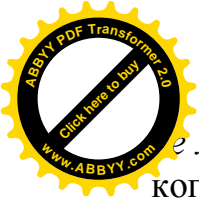
- щелкнуть по кнопке **Сохранить** на панели инструментов окна MS Access;
- в открывшемся диалоговом окне **Сохранение** ввести имя таблицы;
- щелкнуть по кнопке **ОК** окна **Сохранение**.

Для ввода данных в созданную таблицу необходимо в окне **База данных** щелкнуть по имени таблицы и в открывшемся окне с пустой таблицей начать вводить в соответствующие ячейки данные.

22.5.3. Создание связей

Для того, чтобы совокупность таблиц, содержащих информацию о сущностях, могли образовать единую базу данных, необходимо их связать. При создании таблиц с помощью *мастера* таблиц это делается на *последнем шаге* работы. При создании таблиц с помощью *конструктора* таблиц такая операция *не предусматривается*. Следовательно, созданные таблицы необходимо связать самостоятельно.

При создании связей между таблицами связываемые поля, содержащие *первичный* и *внешний* ключи, могут иметь различные имена. Однако они никогда



е могут иметь различные типы данных. Исключением может быть только сл, когда поле первичного ключа является полем типа счетчик. Это поле можно связывать с любым числовым полем, если значение свойства **Размер поля** обеих полей совпадают.

Создавать или изменять связи между открытыми таблицами нельзя. Поэтому необходимо предварительно закрыть все открытые таблицы, если таковые имеются.

Для создания связей между таблицами следует придерживаться следующего порядка:

- открыть окно **Базы данных**;
- щелкнуть по кнопке **Схема данных** на панели инструментов программы MS Access или выполнить команды **Сервис – Схема данных** главного меню программы. В результате этих операций откроется окно **Схема данных**, а в случае, если связи между таблицами еще не определены, то откроется и еще одно дополнительное окно **Добавление таблицы**. В нем будет представлен список названий таблиц, которые можно добавить в окно **Схема данных**. Если это окно не открывается, и в окне **Схема данных** отсутствуют таблицы, для которых необходимо установить связь, то следует выполнить последовательность команд

Связи – Добавить таблицу

или щелкнуть по кнопке **Отобразить таблицу**.

- в открытом окне **Добавление таблицы** следует сделать двойные щелчки по названиям таблиц, между которыми необходимо установить связи;
- щелкнуть по кнопке **«Закрыть»** окна **Добавление таблицы**. После этого в окне **Схема данных** появятся подчиненные окна, содержащие названия и списки полей, выбранных на предыдущем этапе таблиц. Причем ключевые поля этих таблиц будут выделены полужирным шрифтом;
- выделить ключевое поле главной таблицы и перетащить его к полю внешнего ключа подчиненной таблицы, расположив его над ним. После этих операций откроется окно **Изменение связей**.
- Установить в окне **Изменение связей** флажок **Обеспечение целостности данных**;
- щелкнуть по кнопке **Создать** окна **Изменение связей** и этим закрыть его. После закрытия окна **Изменение связей** в окне **Схема данных** отобразится связь в виде линии, связывающей ключевые поля таблиц. При этом, на линии связи возле главной таблицы, появится *цифра 1*, а возле подчиненной таблицы – *значок бесконечности*. Так отображается связь вида *один ко многим*. В соответствии с этой процедурой устанавливаются все связи между таблицами.
- Выполнить операции с пунктами главного меню программы

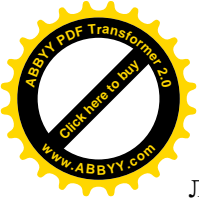
Файл – Сохранить;

- закрыть окно **Схема данных**.

22.5.4. Редактирование, удаление связей

Для изменения или удаления ранее установленной связи необходимо:

- открыть окно **Схема данных**;



– в контекстном меню связи (курсор мыши необходимо точно совместить с линией связи) выбрать команду **Изменить связь** или **Удалить**.

ЛЕКЦИЯ № 23

23. ПРОГРАММА MS ACCESS. РАБОТА С ОБЪЕКТАМИ

23.1. Формы. Работа с формами

Форма – это объект СУБД MS Access, который представляет собой *диалоговое окно*. Это окно используется для *отображения данных*, содержащихся в базе, а также для их *ввода и редактирования*. Форма предназначена для работы *только с одной записью* таблицы. Это избавляет от прокрутки таблицы при поиске необходимой записи.

Окно формы имеет различные элементы управления:

- поля,
- кнопки,
- переключатели,
- списки,
- флажки и др.,

которые служат для выполнения, отмеченных выше операций. В полях отображаются, вводятся и редактируются данные из таблиц базы.

Обычно форма присоединяется к одной или к нескольким таблицам базы, которые заполняются данными из полей формы или же из которых данные поступают в поля формы для отображения или редактирования. Такие таблицы считаются *источником данных* для формы.

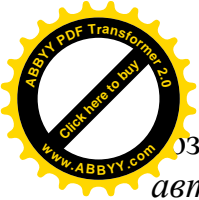
В форму не обязательно включаются все поля из каждой таблицы, на основе которой она создается. Кроме всего прочего, поля формы могут использоваться для вычисления значений выражений, например, для вычисления некоего итогового значения (суммы, среднего арифметического и т.п.). Такие поля являются аналогом формул в ячейках электронных таблиц. Флажки, переключатели и списки могут использоваться для формирования значений в полях записей. С помощью кнопок, расположенных в формах, можно открывать другие формы или отчеты базы данных. Форма может содержать различные пояснения, заголовки, рисунки, разделительные линии. Все эти элементы управления, текстовые и графические изображения в совокупности составляют *макет формы*.

В MS Access форму можно создавать с помощью:

- *автоформ*,
- *мастера форм*,
- *конструктора форм*.

23.1.1. Создание формы в режиме автоформ

При использовании автоформ следует указывать только ту (одну) таблицу, к которой необходимо привязать форму. При этом все поля таблицы размещаются на



создаваемой форме, а размеры и оформление формы подбираются автоматически.

Для создания формы в режиме автоформы следует:

- *выйти в окно **База данных**,*
- *щелкнуть по объекту **Формы**,*
- *щелкнуть по кнопке **Создать** главного меню окна **База данных**. На экране появится диалоговое окно **Новая форма**, в котором будут представлены варианты категорий формы в виде:*

Автоформа: *в столбец,*

Автоформа: *ленточная,*

Автоформа: *табличная*

- *выбрать вид формы и щелкнуть по нему,*
- *в раскрывшемся в нижней части окна списке названий таблиц щелкнуть по названию таблицы, с которой следует связать форму,*
- *щелкнуть по кнопке **ОК**.*

Результатом таких действий является появление на экране выбранной формы.

Вариант формы *в столбец* предусматривает вывод всех названий полей на экран как один столбец, а справа от названий – их значения. Если столбец, образованный из названий полей не вмещается на экран, то он разбивается на два столбца и т. д.

Ленточная форма позволяет выводить на экран одну запись в виде строки.

Табличная форма предполагает расположение записей в формате таблицы.

23.1.2. Создание формы в режиме мастера форм

В отличие от автоформы, мастер создания форм позволяет учесть пожелания пользователя при выборе полей, макета и стиля отображения создаваемой формы.

Для создания формы в режиме мастера форм следует:

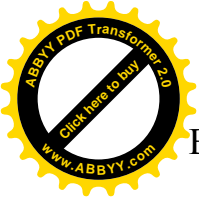
- *выйти в окно **База данных**,*
- *щелкнуть по объекту **Формы**,*
- *дважды щелкнуть по ярлыку **Создание формы с помощью мастера**.*

На экране появляется *первое* окно **Создание форм** мастера создания форм. *Это первый шаг.*

- *Из раскрываемого списка строки **Таблицы и запросы** выбрать название таблицы, которая содержит поля будущей формы. После этого окно **Доступные поля** заполнится названиями полей этой таблицы.*

- *С помощью кнопки **>**, расположенной справа от этого окна, поместить названия нужных полей в рядом расположенное окно **Выбранные поля**. Эту же операцию можно также выполнить двойным щелчком по имени поля. Если требуется включить в будущую форму поля других таблиц, то следует возвратиться к строке **Таблицы и запросы** и установить название другой таблицы. Выбор ее полей выполнить уже описанным способом.*

- *После завершения формирования списка полей необходимо щелкнуть по кнопке **Далее** окна **Создание форм** и перейти ко **второму** шагу мастера.*



В новом окне следует выбрать один из вариантов внешнего вида формы:

- *в один столбец,*
- *ленточный,*
- *табличный,*
- *выровненный,*
- *сводная таблица,*
- *сводная диаграмма.*

Чаще используется вид *в один столбец.*

– *щелкнуть по кнопке **Далее** после завершения выбора внешнего вида формы.*
После этого происходит переход к **третьему** шагу мастера форм.

В окне третьего шага мастер предлагает 10 вариантов стиля для формы:

- *диффузный,*
- *камень,*
- *международный,*
- *наждачная бумага,*
- *официальный и др.*

– *щелчком по кнопке **Далее** завершается работа третьего и происходит переход к четвертому шагу.*

В последнем окне мастера

- *в поле **Задайте имя формы** следует заменить предлагаемое имя формы на желаемое,*
- *включить переключатель **Открыть форму** для просмотра и ввода данных или **Изменить макет формы,***
- *щелкнуть по кнопке **Готово.***

При выборе варианта **Изменить макет формы** произойдет переключение в режим конструктора, в котором можно выполнить любую модификацию формы, созданной мастером.

23.1.3 Ввод и редактирование записей с помощью формы

После того как форма создана, ее можно использовать для ввода или редактирования записей в тех таблицах и в тех полях, с которыми она связана. Для реализации этих действий необходимо выполнить следующие операции:

- *щелкнуть в окне **База данных по объекту Форма,***
- *в появившемся списке имен созданных форм выделить нужное имя и выполнить по нему двойной щелчок,*



- в раскрывшемся окне выбранной формы выбрать нужную запись таблицы,
- щелкнуть по нужному полю и ввести требуемое значение. После ввода необходимых значений во все поля записи осуществить переход к следующей записи. Это делается щелчком по кнопке **Новая запись** на панели переходов.

Для удаления текущей записи следует выполнить команды главного меню **Правка – Удалить запись**.

23.1.4. Простейшие модификации формы с помощью конструктора

Создание новой формы с помощью конструктора является достаточно сложной задачей, и поэтому в дальнейшем рассматриваются приемы редактирования с помощью конструктора ранее созданной формы.

Для перехода в режим конструктора следует:

- в окне **База данных** щелкнуть по объекту **Форма**,
- в появившемся списке с именами созданных форм выделить имя нужной формы и щелкнуть по кнопке **Конструктор** окна **База данных**,
- в раскрывшемся списке выбрать строку **Конструктор формы**. В случае, когда форма уже открыта, следует вызвать контекстное меню и в нем выбрать команду **Конструктор формы**.

В раскрывшемся окне конструктора отобразится выбранная форма в виде доступном для редактирования. Используя это окно и другие подчиненные окна, можно вносить различные изменения в макет формы (изменения значений полей, выбор цветового фона, цвета заливки, фоновый рисунок). Для внесения изменений в макет формы следует использовать команду **Свойства** контекстного меню.

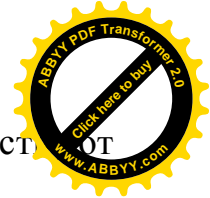
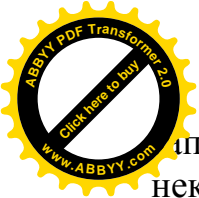
23.2. Поиск и запросы

Поиск нужных сведений путем обыкновенного просмотра таблиц среди большого объема и многообразия информации, содержащейся в базе данных, является весьма трудоемкой операцией, занимающей достаточно много времени.

СУБД MS Access располагает средствами, упрощающими и автоматизирующими операции поиска необходимой информации. Самыми мощными инструментами поиска нужной информации являются *запросы*. Вместе с тем программа MS Access предлагает и более простые средства поиска информации.

23.2.1. Поиск и фильтрация данных

Для нахождения определенной записи в таблице можно обратиться к упрощенному средству поиска – команде **Найти**. С ее помощью можно отыскать



писи, значение полей которых полностью или частично соответствует некоторому условию поиска.

Для поиска определенной записи необходимо выполнить следующую последовательность действий:

- *открыть таблицу, в которой требуется совершить поиск.* Для этого нужно открыть окно **База данных**, щелкнуть по объекту **Таблицы**, в раскрывшемся списке таблиц дважды щелкнуть по имени интересующей таблицы.
- *Выделить нужное поле таблицы, в пределах которого будет совершаться поиск щелчком по заголовку или любой ячейке поля.*
- *Выполнить команду **Найти** любым из трех способов:*
 - *с помощью команд главного меню программы MS Access*

Правка – Найти,
 - *щелчком по кнопке **Найти** на панели инструментов,*
 - *комбинацией клавиш **Ctrl + F**.*

В открывшемся диалоговом окне **Поиск и Замена** на вкладке **Поиск**, в раскрывшемся списке **Поиск в**, по умолчанию будет установлено имя выбранного поля.

– *В текстовое окно **Образец** окна **Поиск и Замена** ввести фрагмент данных, которые необходимо найти.* Это можно сделать:

- *с помощью кнопки просмотра окна **Образец**, раскрыв список и, выбрав искомое значение.*

Если точное значение неизвестно, то можно воспользоваться маской из символов *, # и ?.

Символ * соответствует произвольной строке символов: a^* — $a1$, $azbyka$, $a2x4$.

Символ ? соответствует одному символу: $EAPU07?$ — $EAPU06a$, $EAPU06b$, $EAPU06v$.

Символ # соответствует одной цифре: $\text{№}100$ — $\text{№}###$.

- *Если предполагается вести поиск не только по текущему полю, а по всем полям таблицы, то следует раскрыть список в окне **Поиск в** и щелкнуть по имени таблицы или формы;*
- *раскрыть список строки **Совпадение** и выбрать одну из строк*
 - *с любой частью поля,*
 - *поля целиком,*
 - *с начала поля,*

определяющих участок просматриваемого поля.

а) Если выбрана строка «**С любой частью поля**», то при установлении образца **АЛЕКСАНДР** из всех значений поля будут выбраны значения:

АЛЕКСАДРА, АЛЕКСАНДРИЯ, АННА АЛЕКСАНДРОВНА.

б) Если выбрана строка «**Поля целиком**», то при установлении образца **ОМСК** поле, содержащее значение **ТОМСК** будет пропущено. Здесь требуется полное совпадение.

в) Если выбрана строка «**С начала поля**», то при установлении образца **ЛУГ** из всех значений поля будут выбраны значения:

ЛУГАНСК, ЛУГОВОЕ, но не КАЛУГА.



- При необходимости сужения области поиска по регистру текст данных необходимо установить флажок опции «С учетом регистра». Тогда при установленном образце *ВАСИЛЬЕВА* значения *васильева* и *Васильева* будут проигнорированы.
- Для поиска тех полей, формат значений которых совпадает с форматом образца необходимо установить флажок опции «С учетом формата полей». Это означает, что при образце *30.04.07* значение *30 – апр – 2007* будет проигнорировано.
- После установления всех параметров поиска, чтобы запустить процесс поиска, следует щелкнуть по кнопке **Найти далее** окна **Поиск и Замена**.
- Если поиск выполнен успешно, то указатель в таблице устанавливается на найденной записи. Для нового поиска следует щелкнуть по кнопке **Найти далее**.

Иногда требуется найти не только нужные записи, но и выполнить в них замену образца поиска на другое значение. Эта операция выполняется так же, как и в текстовом редакторе MS Word с помощью команд

Правка – Замена

После этого следует задать образцы поиска и замены, установить параметры поиска и щелкнуть по нужной кнопке **Найти далее**, **Заменить** или **Заменить все**.

23.2.2. Применение фильтра

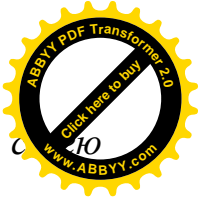
Чаще всего запросы в базу данных создаются с использованием фильтров.

Фильтр в MS Access представляет собой набор условий накладываемых на различные поля одной и той же таблицы. Из этого следует, что фильтр может связываться только с одной таблицей. Проще говоря, позволяет из всей, имеющейся в таблице, информации показывать ту, которая будет оговорена в запросе. Например, в таблице **Студенты** содержится информация о студентах потока ЭАПУ07 (фамилии, дисциплины, экзаменационные оценки и др.). Если необходимо выбрать информацию о том, какие студенты, с какой группы получили по «Информатике» экзаменационные оценки 4 и 5 за первый семестр, то следует сформулировать запрос, в котором необходимо указать какие следует отобразить поля и какие значения полей. Для рассматриваемого примера, очевидно, что необходимо отображать поля

- *Фамилия, Имя, Отчество* студента,
- *группа,*
- *дисциплина,*
- *семестр,*
- *оценка >= 4.*

Следствием этих действий будет сформирован запрос, в котором будут представлены фамилии студентов и их группы, получивших по «Информатике» в первом семестре оценки 4 и 5. То есть из всех записей будут отфильтрованы только те поля, которые содержат интересующую информацию, *остальная информация будет проигнорирована*.

Для установки фильтра необходимо выполнить следующие действия:



- известными приемами *открыть* нужную таблицу или связанную с ней форму,
- щелкнуть по кнопке **Изменить фильтр** на инструментальной панели или выполнить команды главного меню **Фильтр – Изменить фильтр**.

В результате этих действий откроется пустая таблица, в полях которой следует установить необходимые значения. Пример такой таблицы приведен на рис. 23.1.

| Студенты: фильтр | | | | | | | |
|------------------|--------|--------|-------------|---------|--------|---------------------|--|
| Код преподават. | Ф.И.О. | Группа | Дисциплина | Семестр | Оценка | Получение стипендии | |
| | | | Информатика | 1 | >= 4 | Да | |
| | | | | | | | |

Рис.23.1. Форма таблицы для установки фильтра.

Фильтр показывает, по какому признаку следует осуществлять отбор информации. Щелчок по полю активизирует его и выводит какое-либо значение этого поля. Если оно не устраивает, то можно с помощью кнопки, изображаемой в правой части активизированной ячейки, выбрать необходимое значение поля из числа возможных значений. В полях **Оценка** и **Получение стипендии** установить критерий отбора.

После установки всех параметров фильтра необходимо:

- щелкнуть по кнопке **Применение фильтра** на панели инструментов или выполнить команды **Фильтр – Применить фильтр**.

В результате этих действий на экране отобразится таблица записей в соответствии с условиями фильтра.

Чтобы снять фильтр, необходимо:

- щелкнуть вторично по кнопке установки фильтра, которая к этому моменту будет уже иметь название **Удалить фильтр** или выполнить команды **Фильтр – Удалить фильтр**.

Установленный фильтр можно сохранить для дальнейшего использования. Это делается стандартными способами.

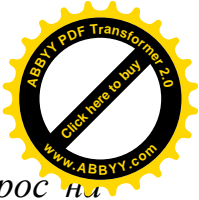
23.3. Типы запросов

В отличие от фильтров запросы могут выполнять не только информационную функцию, отображая какие-либо данные из таблиц, но и выполнять анализ данных.

Результат работы запроса – это группа записей, которые удовлетворяют заданному критерию запроса.

В MS Access существует несколько видов запросов:

- на выборку,
- с параметрами,
- на изменение,
- перекрестный
- SQL – запрос.



Запрос на выборку

Простейшим и наиболее часто употребляемым запросом является *запрос на выборку*. Он устанавливается *по умолчанию*. В отличие от фильтра, в нем формируются требования на отбор информации *из одной или нескольких таблиц*. Результатом выполнения такого запроса является *набор записей, отображаемых в виде таблицы*. Кроме того, запросы на выборку позволяют располагать информацию в определенном порядке, осуществлять расчет сумм, средних значений, количества записей и других итоговых значений.

Запрос с параметрами

Это интерактивный (диалоговый) тип запроса. Прежде, чем запрос будет выполнен, на экран выводится диалоговое окно с приглашением ввести один или ряд параметров, например, условие отбора записей по определенному полю. В этих запросах фигурируют величины, которые в разных вариантах запроса могут принимать различные значения. Например, успеваемость студентов по «Информатике» за первый или второй семестр, то есть временной интервал, из которого берется информация. Здесь условия отбора информации одни и те же, но в разные периоды времени. Такие параметрами, например, время, по которым производят отбор информации, называют *параметрами запроса*.

Запрос на изменение

В запросах на изменение содержатся требования *на изменение* значений в тех или иных полях, на удаление или добавление новых записей, а также создать на основе результирующего набора новую таблицу. Для конкретизации изменений предусмотрены четыре подтипа запросов:

- *запрос на обновление,*
- *запрос на добавление,*
- *запрос на удаление,*
- *запрос на создание таблицы.*

Перекрестный запрос

Результаты такого вида запроса представляются в специальном формате, напоминающем электронную таблицу. Они группируются по двум наборам данных. Первый набор выводится в виде столбца слева (аналог оси ординат), а второй – в виде строки вверху (аналог оси абсцисс), в ячейках на пересечении столбцов и строк помещается интересующая информация. Например, столбец из фамилий студентов, а строка название дисциплин, на пересечении – экзаменационная оценка.

SQL – запрос

SQL – запросы относятся к наиболее обобщающим запросам. Они могут представляться как запросами на выборку, запросами с параметрами или запросами на изменение. Отличительной чертой SQL – запросов является их запись (синтаксис) в соответствии с правилами международного языка запросов, которые применяются как стандартные средства составления запросов в большинстве СУБД.



23.3.1. Создание запроса на выборку

Создать запрос на выборку можно выполнить двумя путями:

- с помощью мастера запросов,
- с помощью конструктора.

23.3.1.1. Создание запроса с помощью мастера

Для создания запроса с помощью мастера необходимо выполнить следующую последовательность действий:

- щелкнуть в окне **База данных по объекту Запросы**,
- щелкнуть по кнопке **Создать инструментальной панели окна База данных**,
- в окне **Новые запросы** выбрать строку **Простой запрос**.

Эти действия приводят к *первому шагу* мастера.

В раскрывшемся окне *первого шага* с помощью списков Таблицы и запросы и Доступные поля, а также кнопок >, >>, <, << сформировать список Выбранные поля,

- щелкнуть по кнопке **Далее** окна Таблицы и запросы.

На этом завершается первый шаг мастера и происходит переход ко второму его шагу.

Второй шаг мастера запросов.

- из раскрывшегося окна второго шага *выбрать один из вариантов* запроса:

- *подробный*,
- *итоговый*;

Выбор варианта **Подробный** приводит к включению *всех отобранных полей всех записей*. Он устанавливается по умолчанию.

Выбор варианта **Итоговый** приводит к разблокированию (отображение контрастным шрифтом) кнопки **Итоги**. Щелчок по этой кнопке вызывает открытие окна, содержащего флажки **Sum** (суммирование), **Avg** (среднее), **Min** (наименьшее), **Max** (наибольшее) для каждого числового поля запроса и общий флажок **Подсчет числа записей**. С помощью указанных флажков можно включить режим группировки по строкам таблицы с созданием в запросе дополнительных столбцов, соответствующих включенным флажкам. После установки необходимых флажков следует:

- щелкнуть по кнопке **ОК**, что приведет к переходу к предыдущему окну,
- щелкнуть по кнопке **Далее**, что приведет к переходу в окно *третьего шага* мастера.

В последнем окне (*третьего шага*) предлагается ввести название созданного запроса и открыть его для просмотра или модификации.

Чтобы завершить работу с мастером, необходимо

- щелкнуть по кнопке **Готово** последнего окна мастера запросов.

Этой последней операцией завершается работа мастера. При этом программа MS Access автоматически сохраняет и выполняет созданный запрос, а также отображает результат запроса в отдельном окне.

23.3.1.2. Создание запроса с помощью конструктора

В отличие от мастера запросов конструктор предлагает более гибкие возможности. Для перехода в режим конструктора запросов необходимо выполнить следующие действия:

База данных – Запросы – Создать – Конструктор

или же вместо двух последних операций *дважды щелкнуть* по строке **Создание запроса в режиме конструктора**.

В результате этих действий откроется окно с пустым *бланком запроса* и окно **Добавление таблицы** такое же, как и при создании связей между таблицами.

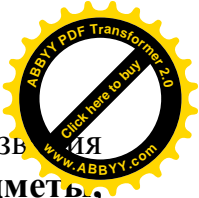
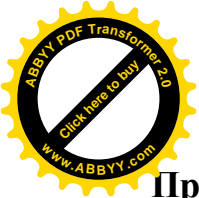
В окне **Добавление таблицы** следует выполнить *двойные щелчки* по названиям всех таблиц, из которых должны отбираться данные в запрос, а затем *заккрыть* его.

После этого появляется окно бланка запроса (рис.24.2).

| Запрос1: запрос на выборку | | | | | × | | | | | | | | |
|----------------------------|----------------|---|-------------|-------------|---|--|--|--|----------|-------------|--|--|--|
| | | <table border="1"> <tr><td>Преподават.</td></tr> <tr><td>Код препод.</td></tr> <tr><td> </td></tr> <tr><td> </td></tr> </table> | Преподават. | Код препод. | | | | <table border="1"> <tr><td>Предметы</td></tr> <tr><td>Код препод.</td></tr> <tr><td> </td></tr> <tr><td> </td></tr> </table> | Предметы | Код препод. | | | |
| Преподават. | | | | | | | | | | | | | |
| Код препод. | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| Предметы | | | | | | | | | | | | | |
| Код препод. | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| Поле: | Фамилия | Должность | Предметы | Семестр | | | | | | | | | |
| Имя таблицы | Преподават. | Преподават. | Предметы | Предметы | | | | | | | | | |
| Сортировка | По возрастанию | | По убыванию | | | | | | | | | | |
| Вывод на экран | ▼ | ▼ | ▼ | ▼ | | | | | | | | | |
| | | | | < 6 | | | | | | | | | |
| | | | | 8 | | | | | | | | | |
| Условие отбора | | | | | | | | | | | | | |
| Или: | | | | | | | | | | | | | |

Рис. 23.2. Общий вид окна бланка запроса

В верхней части бланка отображаются подчиненные окна ранее выбранных таблиц со списками полей и с изображением связей между таблицами. В нижней части окна располагается пустая таблица с элементами управления (слева от нее), предназначенными для создания запроса (*поле;* *имя таблицы;* *сортировка;* *вывод на экран;* *условие отбора;* *или:*). В эту таблицу следует внести имена интересующих полей, имена таблиц, которым эти поля принадлежат. Проще всего эту операцию можно выполнять путем *перетаскивания* нужных имен из таблиц, расположенных в верхней части окна.



Таким образом были перемещены с таблицы **Преподаватели** назв **Преподаватели**, **Должность**, а с таблицы **Предметы** – названия **Предметы**, **Семестр**.

В строке *Сортировка* любого выбранного столбца бланка можно установить способ сортировки (*по возрастанию или убыванию*). Для этого необходимо щелкнуть по ячейке, в которой необходимо указать вид сортировки, и с помощью появившейся кнопки раскрытия списка установить способ сортировки.

Если в запросе необходимо выполнить групповые операции (*сумма, среднее, минимальное, максимальное и др.*), то необходимо:

- *вызвать контекстное меню щелчком правой клавиши мыши по любой ячейке бланка,*
- *выбрать команду **Групповые операции**. Вследствие этого, в бланке запроса, выше строки *Сортировка*, появится строка *Групповая операция*,*
- *щелкнуть по ячейке, которая должна участвовать в групповой операции,*
- *щелкнуть по появившейся в ячейке кнопке раскрытия списка и выбрать из списка необходимую операцию,*
- *ответить на поля Вывод на экран, Условия отбора,*
- *сохранить созданный запрос в файле базы данных с помощью известной процедуры:*

Файл – Сохранить – указать имя – ОК.

23.3.2. Выполнение запроса

Для выполнения только – что, созданного в режиме конструктора, запроса следует:

- *щелкнуть по кнопке **! Запуск** основного меню базы данных или выполнить команды меню **Запрос – Запуск**.*

Для выполнения ранее созданного и сохраненного в базе данных запроса следует:

- *щелкнуть по пункту **Запросы** меню базы данных,*
- *щелкнуть по кнопке **Открыть** панели инструментов базы данных*

23.3.3. Модификация запроса

Для изменения ранее созданного запроса следует:

- *щелкнуть по объекту **Запросы** окна **База данных**,*



- щелкнуть по строке с названием запроса этого окна,
- щелкнуть по кнопке **Конструктор** панели инструментов окна,
- в открывшемся окне внести необходимые изменения в бланк запроса.

Если в запрос необходимо добавить поля из таблицы, которая отсутствует в бланке запроса, то необходимо:

- щелкнуть по кнопке **Отобразить таблицу** или
- выполнить команды **Запрос – Отобразить таблицу**,
- дважды щелкнуть по названию недостающей таблицы в раскрывшемся окне

Добавление таблицы,

- щелкнуть по кнопке **Заккрыть**.

Удаление ненужного поля производится с помощью клавиши **Del**.

ЛЕКЦИЯ №24

24. СОЗДАНИЕ ОТЧЕТОВ В MS ACCESS. ПРЕЗЕНТАЦИЯ

24.1. Создание отчетов

В MS Access предусмотрено создание такого объекта как *отчет*.

Отчет – это наглядная форма представления информации в печатном виде. Проще говоря, это документ для печати. Структура отчета, созданного средствами MS Access, предусматривает:

- *заголовок*,
- *основную (информационную) часть*,
- *верхний и нижний колонтитулы*.

В отчетах предусматривается возможность сортировки записей и их группировки по различным признакам. Кроме этого разработчику предоставляются необходимые средства для predания отчету нужного вида.

Отчеты могут быть связаны с одной или несколькими таблицами, запросами. В отчеты можно поместить графики и рисунки, включить нумерацию страниц и выводить значения вычисляемых полей. Элементы оформления отчета (*заголовки, даты, номера страниц, поясняющие надписи, разделительные линии*) сохраняются в макете отчета.

Предусмотрены три способа создания отчетов:

- *в режиме автоотчета*,
- *с помощью мастера отчетов*,
- *в режиме конструктора*.



24.1.1. Режим автоотчета

Автоотчет – это самый простой, но мощный инструмент, с помощью которого можно начать знакомство с отчетами, как одним из объектов базы данных.

Создание отчета в режиме автоотчета происходит на основе *только одной таблицы* или *одного запроса*. Возможность выбора полей *не предусмотрена*. В автоотчет помещаются все поля исходной таблицы или запроса. Хотя автоотчет работает только с одной таблицей, но на основании хорошо продуманного многотабличного запроса, он позволяет быстро перенести на бумагу данные из нескольких таблиц.

Для создания отчета необходимо:

- *щелкнуть в окне **База данных по объекту Отчет**,*
- *щелкнуть по кнопке **Создать панели инструментов окна База данных**,*
- *в окне **Новый отчет** выбрать строку **Автоотчет в столбец** или **ленточный**,*
- *в окне **Новый отчет** выбрать таблицу или запрос, данные из которых должны помещаться в отчете,*
- *щелкнуть по кнопке **ОК** этого окна.*

После этого будет представлен отчет в режиме *предварительного просмотра*.

Если результаты отчета устраивают, то его следует сохранить с помощью команд **Файл – Сохранить**.

Если же возникнет потребность внести в отчет некоторые изменения, то следует *щелкнуть по кнопке **Заккрыть***. Это приведет к переключению в режим конструктора для модификации, созданного отчета.

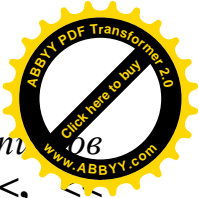
24.1.2. Создание отчета с помощью мастера

Мастер работает с полями нескольких таблиц или запросов и создает отчеты с более сложной структурой, в которой допускается группировка записей по некоторым логическим категориям. Мастер отчетов позволяет быстро создавать структуру отчета, которую затем можно доработать в режиме конструктора.

Чтобы создать отчет под руководством мастера, необходимо выполнить несколько шагов. Для начала необходимо:

- *щелкнуть по объекту **Отчеты** в окне **База данных**,*
- *в раскрывшемся окне дважды щелкнуть по ярлыку **Создание отчета с помощью мастера**.*

Первый шаг.



- В раскрывшемся окне **Создание отчетов первого шага**, с помощью сп... **Таблицы и запросы**, **Доступные поля**, а также кнопок **>**, **>>**, **<**, **<<** этого окна аналогично тому, как это делается при создании форм и запросов, формируется список **Выбранные поля**, значение которых необходимо отобразить в отчете. Эти операции прodelываются многократно с каждой таблицей.
- щелкнуть по кнопке **Далее** этого окна после завершения формирования списка **Выбранные поля**. Это приводит к переходу ко второму шагу мастера отчетов.

Второй шаг.

В раскрывшемся окне **Создание отчетов второго шага** (Рис.24.1) отображаются две панели. В левой панели окна представлены выбранные на первом шаге поля. Значения этих полей используются для создания заголовков групп. То есть создаются уровни группировки записей в отчете, их последовательность по обобщающему признаку. Например,

| | |
|----------------------|---------------------|
| <i>Специальность</i> | (Первый уровень) |
| <i>Группы</i> | (Второй уровень) |
| <i>Дисциплина</i> | (Третий уровень) |
| <i>Оценка</i> | (Четвертый уровень) |
| <i>Ф.И.О.</i> | (Пятый уровень). |

Процесс формирования структуры отчета осуществляется с помощью кнопок **>**, **>>**, **<**, **<<** и похож на создание списка полей в запросах.

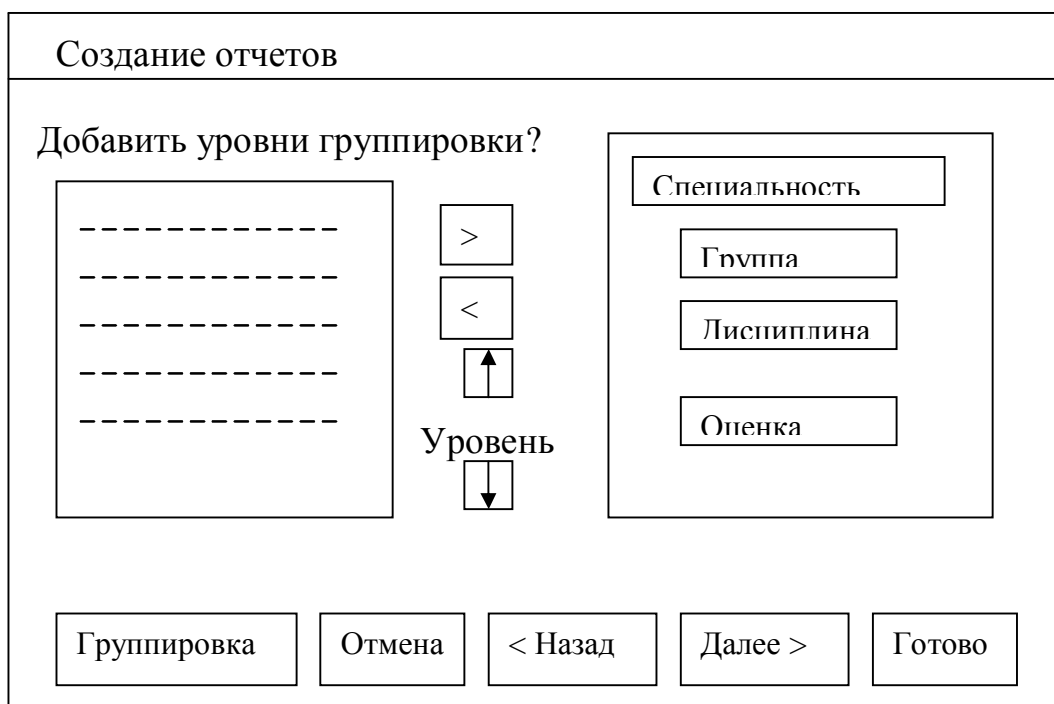


Рис. 24.1 Окно второго шага мастера отчетов

Сначала выбирается верхний уровень, затем второй и т. д. В правой панели окна отображается образец структуры отчета. Вверху представлен самый высокий уровень, далее следуют более низкие уровни. В отчете можно определять всего *четыре уровня* группировки. Если возникает необходимость реорганизовать структуру отчета и создать другие логические подразделы, то следует воспользоваться кнопками **Уровень**. Кнопка со стрелкой, направленной вверх, повышает уровень выделенного поля, а со стрелкой, направленной вниз, понижает уровень. После завершения формирования структуры отчета необходимо:

- *щелкнуть по кнопке **Далее***. Этим действием производится переход к *третьему шагу* мастера отчетов.

Третий шаг.

В окне третьего шага предоставляется возможность формирования итоговых записей и определения их следования в группах. Мастер позволяет выбрать до четырех полей сортировки.

Чтобы задать поле, по которому следует выполнить сортировку, следует выбрать имя нужного поля из раскрывающегося списка под номером 1. Справа от этого списка расположена кнопка, на котором указан порядок сортировки (*по возрастанию, по убыванию*). По умолчанию принимается сортировка *по возрастанию*. Эта кнопка действует как переключатель. Щелчок по ней переводит с одного вида сортировки на другой и наоборот.

После завершения сортировки *следует щелкнуть по кнопке **Далее*** и перейти к *четвертому шагу* мастера.



Четвертый шаг.

В окне четвертого шага мастера предлагается один из шести стандартных вариантов оформления отчета:

- *ступенчатый,*
- *блок,*
- *структура и др.,*

На этом этапе следует выбрать *ориентацию* отчета (*альбомная, книжная*), режим автоматического выбора ширины полей.

Щелчком по кнопке **Далее** завершается *четвертый шаг* мастера и происходит переход к *пятому* шагу.

Пятый шаг.

На пятом шаге предлагается *стиль оформления* отчета:

- *деловой,*
- *обычный,*
- *полужирный,*
- *сжатый и др.*

После выбора стиля *следует щелкнуть по кнопке Далее* этого окна. Этим действием *завершается работа на пятом шаге* и *следует переход к шестому шагу* мастера.

Шестой шаг.

На этом шаге устанавливается *имя отчета*, способ его открытия (*для просмотра, для изменения*).

Завершается процесс создания отчета щелчком по кнопке **Готово**.

24.1.3. Модификация отчета в режиме конструктора

Отчет, созданный с помощью автоотчета или мастера, как правило, нуждается в доработке. Этот вид работы удобно выполнять в режиме конструктора.

Для внесения изменений в ранее созданный отчет необходимо:

- *щелкнуть по объекту **Отчеты** в окне **База данных**,*
- *щелкнуть по строке с названием отчета,*
- *щелкнуть по кнопке **Конструктор** панели инструментов окна. В раскрывшемся окне отобразится макет отчета.*
- *С помощью команд меню и панели инструментов внести необходимые изменения в макет (в его содержание или оформление),*



– *сохранить отчет стандартными операциями.*



24.2. Печать отчета

Прежде, чем печатать созданный отчет, его следует просмотреть. Это следует делать для того, чтобы окончательно убедиться, что отчет отвечает всем предъявляемым требованиям. Для этого необходимо:

- *щелкнуть по объекту **Отчеты** в окне **База данных**,*
- *щелкнуть по кнопке **Просмотр** панели инструментов.*

В режиме предварительного просмотра можно выбрать масштаб изображения отчета. Если необходимо внести некоторые изменения в макет отчета, то следует выполнить команды

Файл – Параметры страницы

и установить необходимые параметры, как это делается в текстовом редакторе MS Word. После завершения всех подготовительных операций к печати отчета следует переходить непосредственно к его печати. Лучше всего это делать следующим образом:

Файл – Печать – Параметры печати

В окне **Параметры печати** можно установить *диапазон печати* (все или отдельные страницы), *количество экземпляров*, *ориентацию страниц* (книжная, альбомная) и др. После выполнения всех необходимых установок следует

- *щелкнуть по кнопке **ОК**.*

Сводная таблица

Для вызова сводной таблицы необходима команда **Данные → Сводная таблица**. При вызове появляется диалоговое окно «Мастер сводных таблиц». В данном окне необходимо установить: Откуда берутся данные – из таблиц, либо из листа Excel, либо из внешних данных.

После заполнения первого диалогового окна, заполняем второе диалоговое окно, в котором выделяем диапазон созданной базы данных.

В третьем окне необходимо установить, где будет расположена сводная таблица (по типу диаграммы при построении графиков).

После заполнения всех окон необходимо создать макет таблицы. Поля базы данных, на основе которых строится сводная таблица, представлены в окне создания макета в виде кнопок этих полей. Перетаскивание их в соответствующее место и создает соответствующую структуру таблицы.

В окне таблицы имеются области:

- Строка (для использования данных полей в качестве заголовка строки)
- Столбец (для использования данных полей в качестве заголовка столбцов)



- Данные (для суммирования значений полей)
- Страница

Мастер сводных таблиц - макет

| | | | | |
|--|--|--|--|--|
| | | | | |
| | | | | |
| | | | | |
| | | | | |

| | | | | |
|----------|--------------|---------------|---------|--------------|
| страница | | город | столбец | наименование |
| | наименование | Сумма по полю | | |
| | строка | данные | | дата |
| | | | | поставщик |
| | | | | город |
| | | | | прибыль |

Презентация 7 – 10 слайдов.

ЛЕКЦИЯ №25

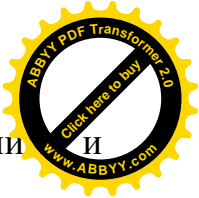
25. РАБОТА В СРЕДЕ MATHCAD

Среда **MathCAD** содержит такие же панели инструментов, как Excel и Word. Исключение составляют три главные панели **MathCAD**:

- Стандартная
- Форматирования
- Математическая

Панель **Math** вызывает на экран 8 дополнительных панелей:

1. калькулятор
2. булевы операции (логические)
3. панель построения графиков
4. для вставки матриц и матричных операций



5. для вставки операторов дифференцирования, суммирования и интегрирования
6. для программирования с использованием **MathCAD**
7. для вставки греческих символов
8. для вставки символов операторов.

Ввод и редактирование формул

Интерфейс редактирования.

1. указатель мыши;
2. курсор (находится внутри документа в одном из трех видов: курсор ввода – крестик красного цвета, который отмечает пустое место в документе, в котором можно вводить текст или формулу; линия ввода – горизонтальная и вертикальная линия синего цвета, выделяющая в тексте или формуле определенную часть);
3. вертикальная линия, определяющая ввод текста;
4. местозаполнители.



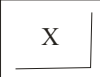
- заполнители символа.

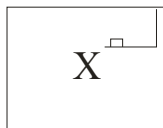


- заполнители оператора.

Порядок ввода формул

(на примере x^{5+x})

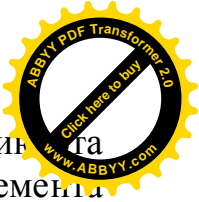
1. Щелкнуть мышью, обозначить место ввода
2. Нажмем клавишу x . в этом месте появится регион с формулой , содержащий символ x (выделенный линиями ввода).
3. Введем оператор, возведем в степень двумя способами (1 – с помощью \wedge ; 2 – с помощью калькулятора).



Вставка оператора в разные части функции

Задачи линейной алгебры в MathCAD. Векторы и матрицы.

Для работы с данными удобно использовать массив. Переменную, представляющую собой список данных, называют одномерным массивом или вектором. Для доступа к данным, находящимся в определенном месте массива, необходимо указать его порядковый номер или индекс. Данные, представленные в виде таблиц в формате строк и столбцов, хранятся в двухмерных массивах (матрицах). Для доступа к нему указывают его имя и два индекса. Нижняя граница



Индексации в **MathCAD** определяется переменной «ORIGIN». По умолчанию переменная равна 0. индексы могут быть только целыми числами. Индекс элемента массива определяется при помощи [после имени переменной.

Пример ввода элементов вектора (матрицы) при ORIGIN=1.

X1:=5; x2:=8; x3:=10

$$X = \begin{pmatrix} 5 \\ 8 \end{pmatrix}$$

$A_{1,1} = 0; A_{1,2} = -2.5; A_{2,1} = -1; A_{2,2} = 5.2$

$$A = \begin{pmatrix} 0.1 & -2.5 \\ -1 & 5.2 \end{pmatrix}$$

ORIGIN=0

$X_0 := 5; X_1 := 8; X_2 := 10$

$A_{0,1} :=$

$A_{1,0} :=$

$A_{1,1} :=$

Работа с матрицами и векторами

Для работы с матрицами и векторами необходимо обратиться к панели №4, которая содержит следующие кнопки:

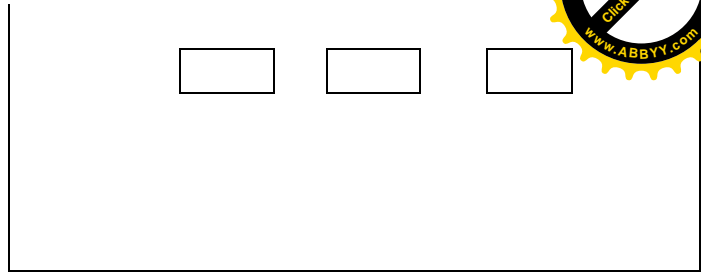
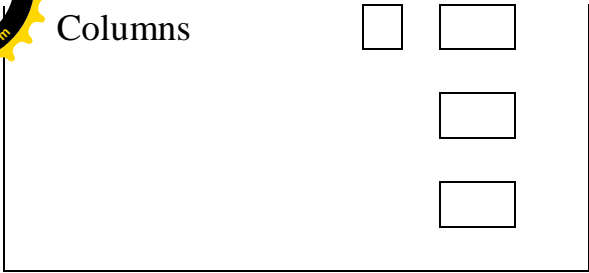
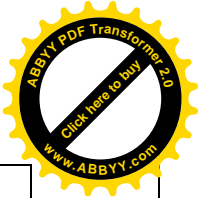
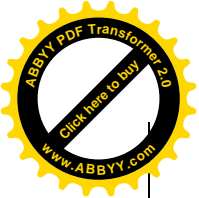
1. определение размеров матрицы
2. вычисление определителя матрицы
3. оператор векторизации (поэлементные операции с векторами и матрицами)
4. скалярное программирование векторов
5. векторное программирование векторов
6. вычисление суммы, компонентов вектора
7. определение столбца матрицы
8. транспонирование матрицы
9. визуализация цифровой информации, сохраняющейся в матрице (обработка рисунков, которые представлены матрицей)

Действия необходимые для ввода матрицы

1. ввести имя матрицы и присвоить A:=
2. на кнопке «панели матриц» выбираем первую кнопку [...]

| | |
|---------------|--|
| insert Matrix | X |
| Rows | <input type="text"/> <input type="button" value="OK"/> |

| | | | |
|-----|----------------------|----------------------|----------------------|
| | <input type="text"/> | <input type="text"/> | <input type="text"/> |
| A:= | <input type="text"/> | <input type="text"/> | <input type="text"/> |



Пример операции над матрицами.

$$A := \begin{pmatrix} 1 & 2 & 3 & 2 \\ 0 & 3 & 2 & 1 \\ 2 & 1 & 0 & 3 \\ 4 & 3 & 0 & 1 \end{pmatrix} - \text{определение матрицы } A \ 4 \times 4.$$

$$A_{2,2} = 3$$

$A^{-1} = (\dots\dots\dots)$ – обратная матрица.

$|A| = 64$ - определитель

$$A^T = \begin{pmatrix} 1 & 0 & 2 & 4 \\ 2 & 3 & 1 & 3 \\ 3 & 2 & 0 & 0 \\ 2 & 1 & 3 & 1 \end{pmatrix} - \text{транспонированная матрица } A.$$

$L := 1/2 * A$ $L = (\quad)$ – умножение матрицы на число.

Умножение матриц.

$$L * A = (\quad)$$

Сложение матриц.

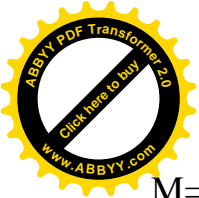
$$A + L = (\quad).$$

Примеры векторных операций

ORIGIN:=1

$$A_i = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 0 & 3 \\ 3 & 1 & 2 \end{pmatrix}$$

$$M_i = H^{<2>}$$



$$M = \begin{pmatrix} 2 \\ 0 \\ 1 \end{pmatrix} - \text{выделение 2го столбца}$$

$$K_i = 2 * M$$

$$K = \begin{pmatrix} 4 \\ 0 \\ 2 \end{pmatrix}$$

$$D_i = M + K - \frac{H^{(3)}}{5}$$

$$D = \begin{pmatrix} 6 \\ -0,6 \\ 2,6 \end{pmatrix}$$

$$U := D^T$$

$$U = (6 \quad -0,6 \quad 2,6)$$

$$d_i = |D|$$

$$d = 6,567$$

$$S_i = \sum D$$

$$S = 8$$

$$M * D = 14 * 6$$

$$M \times D = \begin{pmatrix} 0,6 \\ 0,8 \\ -1,2 \end{pmatrix}$$



Функции определения матриц и операции с блоками матриц

1. `matrix (m, n, f)` – создает и заполняет матрицу A размерностью m – строк на n – столбцов, как элемент равен значению функции f и j

ORIGIN:=1

$$F(x, y) := |y^2 - x^2|$$

$A := \text{matrix}(4, 3, f)$

$$A = \begin{pmatrix} 0 & 3 & 4 \\ 1 & 0 & 3 \\ 4 & 3 & 0 \\ 9 & 8 & 5 \end{pmatrix}$$

$$a_{1,1} = (1^2 - 1^2) = 0$$

$$a_{1,2} = (2^2 - 1^2) = 3$$

2. `diag (v)`

Создает диагональную матрицу, элементы которой хранятся в векторе v

3. `identity (n)`

создает единичную матрицу порядка n

ORIGIN:=1

$E_i = \text{identity}(3)$

$$E = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

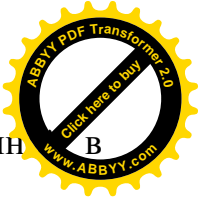
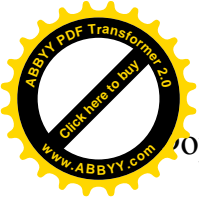
4. `augment (A, B)`

Из матрицы A и B формируют третью матрицу. Первые столбцы которой содержат матрицу A , а последние – матрицу B . (данные матриц должны иметь одинаковое количество строк).

5. `stack (A, B)`

Из матрицы A и B формируют третью матрицу, первые строки которой включают в себя первые строки A , а последние B (матрицы должны иметь одинаковое число столбцов)

6. `submatrix (A, l, k, p, r)`



формируется матрица, которая является блоком матрицы A, расположенных в строках с 1 по k и столбцах с p по r.

Пример.

$$B := \begin{pmatrix} 1 & 2 & 3 \\ 3 & 0 & 5 \\ 1 & 8 & 1 \end{pmatrix}$$

F := submatrix (B, 1.2, 2.3)

$$F = \begin{pmatrix} 2 & 3 \\ 0 & 5 \end{pmatrix}$$

Функции вычисления различных числовых характеристик матриц

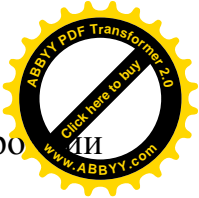
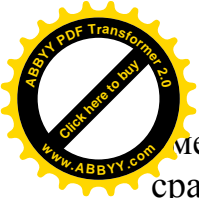
1. last (v) – номер последней компоненты V
2. length (v) – количество компонентов V
3. rows (A) – число строк в матрице A
4. cols (A) - число столбцов в матрице A
5. max (A) - наибольший элемент матрицы A
6. min (A) - наибольшее значение матрицы A
7. mean (A) - среднее значение матрицы A
8. median (A) – медиана матрицы A
9. ranc (A) – ранг матрицы A
- 10.sort (v) – сортировка элементов V в порядке возрастания
- 11.reverse (v) – переставляет элементы V в обратном порядке
- 12.csort (A, n) – переставляет местами строки , чтобы отсортировать столбец N
- 13.rsort (A, n) – переставляет столбцы A, чтобы отсортировать строки N.

Построение графиков в MathCAD

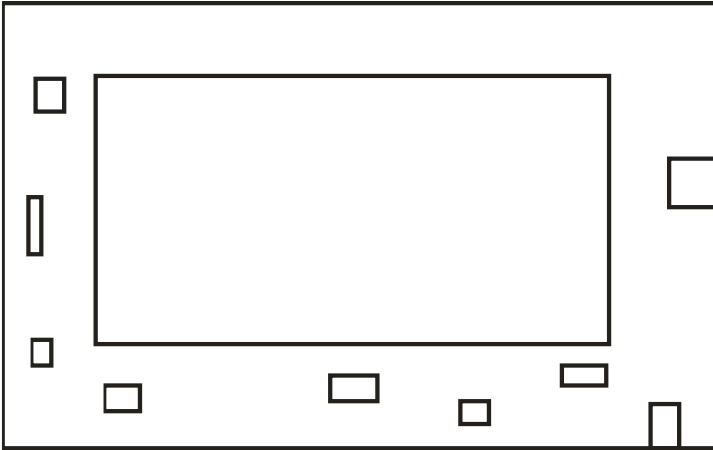
Для построения графиков в **MathCAD** используется панель №3. команды представлены в меню Graf соответственно кнопкам на панели инструментов и позволяют строить следующие графики:

1. X-X Plot (№1)
2. Polar Plot (№2)
3. Surface Plot (5)
4. Contour Plot (6)
5. 3D Scatter Plot (8)
6. Vector Field Plot (9)

Шаблон для построения графика появляется в любом месте, где стоит красный крестик. График можно перетащить с помощью мыши. Графики можно



уменьшить, увеличить, растянуть в условно обозначенных местах. При построении сразу появляется шаблон.



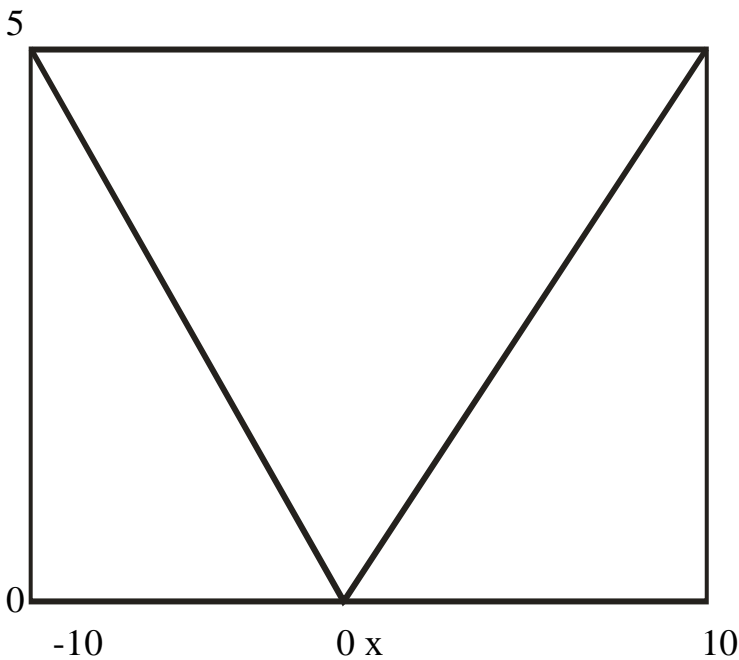
В MathCAD пределы значений аргумента по умолчанию изменяются от [-10; 10].

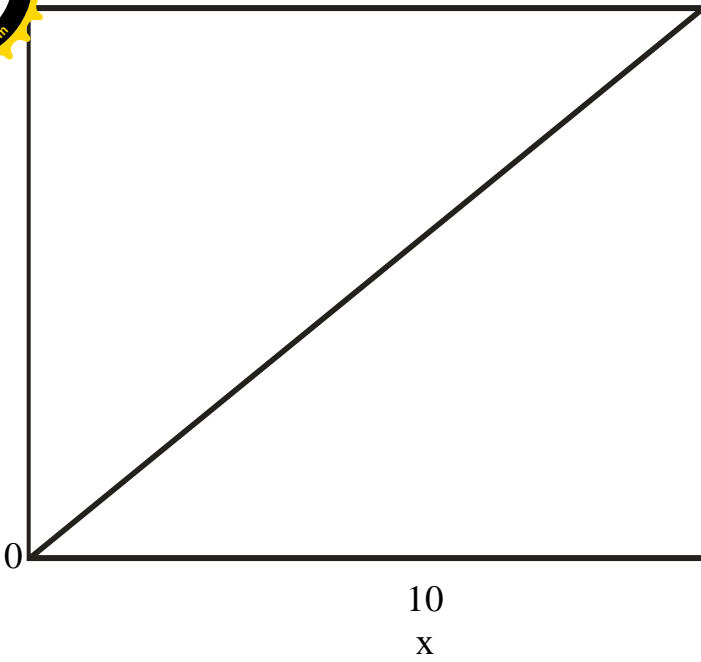
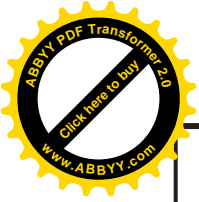
Пример.

$$y = |x| / 2$$

Порядок построения:

1. вставить шаблон для построения графиков (№1) или с помощью меню Graf.
2. внести имя аргумента в поле лоси ординат
3. щелчком мыши на чистом поле кликнуть и будет график





Пример 2.

Построение графиков с ранжированием аргумента

$$y = \cos \frac{x}{2} + \cos \frac{5x}{5}$$

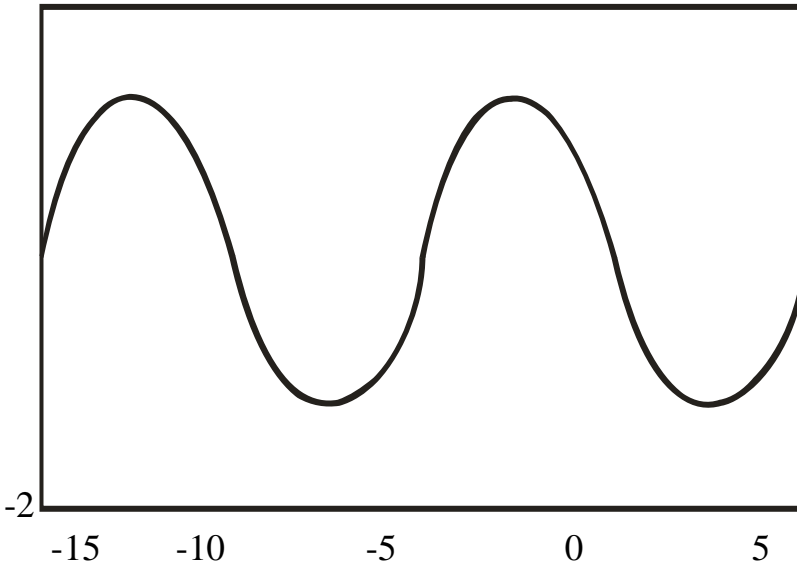
$[-15;15]$

1. $x:=-15; 15$

$$y(x) := \cos \frac{x}{2} + \cos \frac{5x}{5}$$

$y(x)$, шаг=1

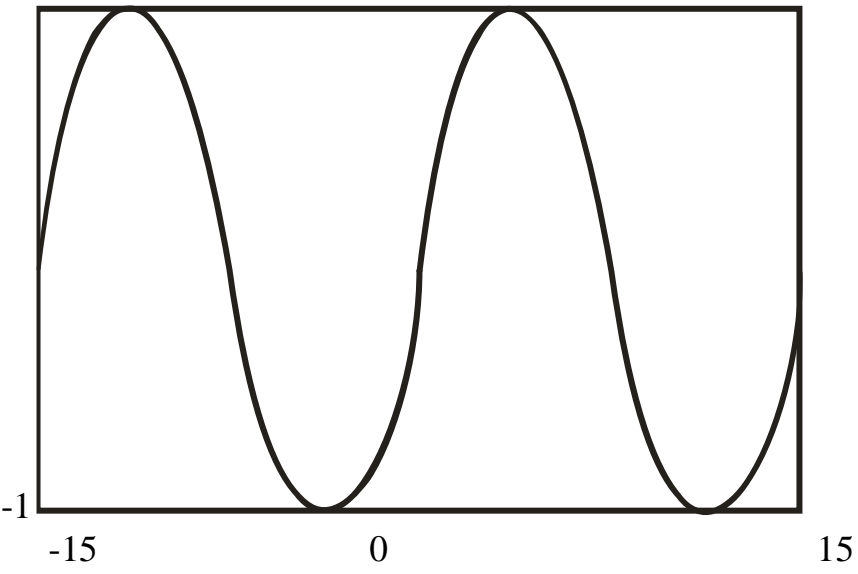
2





1

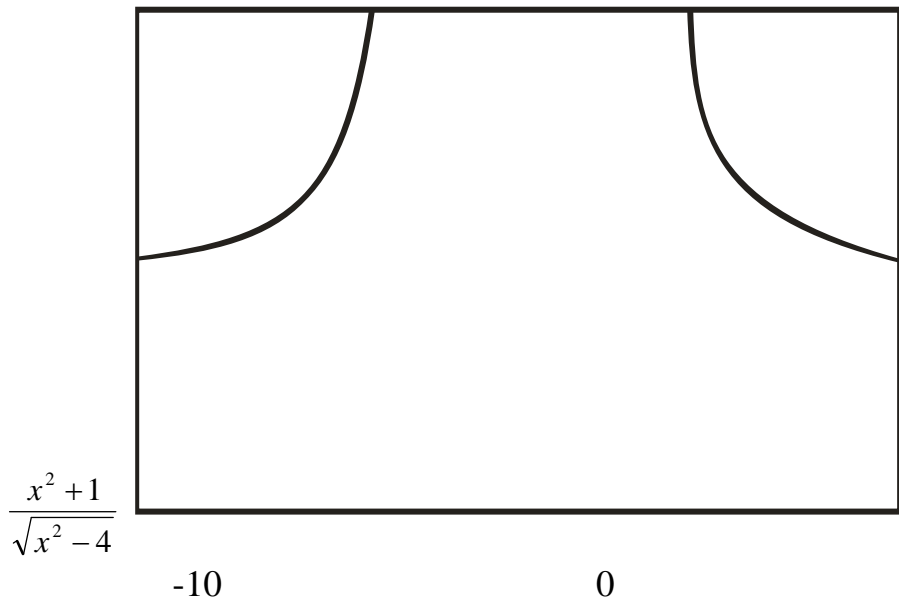
2.



sin (x)

Пример 3.

$$y = \frac{x^2 + 1}{\sqrt{x^2 - 4}}$$

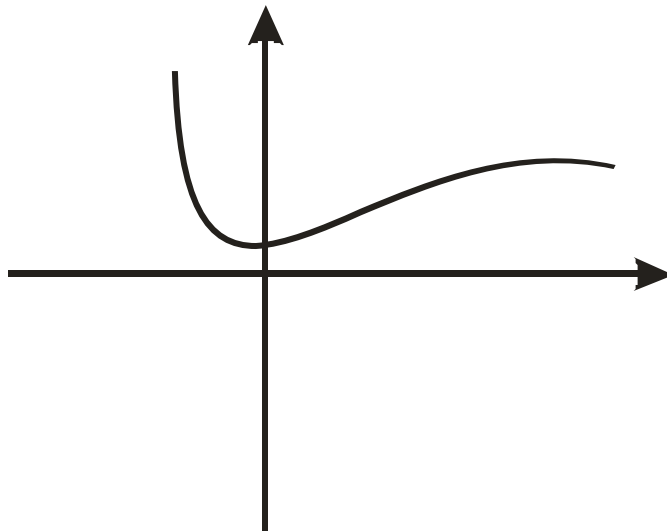


$$\frac{x^2 + 1}{\sqrt{x^2 - 4}}$$

Пример 4.

$$y = \begin{cases} \frac{|x|}{2 + x^2}, & x < 0 \\ \sqrt{1 + x}, & x \geq 0 \end{cases}$$

x:=-4, -3,99 4
 y (x):=if (x<0, (1), (2))



Y (x)

Пример 3.

$$\frac{x^2}{3} + \frac{y^2}{5} - 1 = 0$$

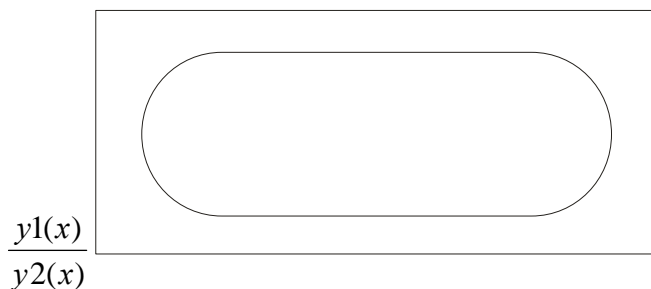
Выражение y через x.

$$\left[\begin{array}{l} \frac{1}{3}(-15x^2 + 45)\frac{1}{2} \\ -\frac{1}{3}(-15x^2 + 45)\frac{1}{2} \end{array} \right]$$

x:=-2, -1,992

$$y \ 1(x) := \frac{1}{3}(-15x^2 + 45)\frac{1}{2}$$

$$y \ 2(x) := -\frac{1}{3}(-15x^2 + 45)\frac{1}{2}$$

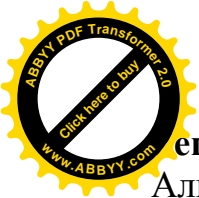


Построение графиков поверхности

Пример 1.

Построить график функции, заданной в виде таблицы

1. определить матрицу (панель №4)
2. вызвать шаблон для создания трехмерного графика командой Incert – Graf – Cervet Plot или нажать кнопки №5 панели №3.
3. записать имя матрицы в поле ввода шаблона
4. щелкнуть мышью за пределами построенного графика



Решение нелинейных уравнений систем в MathCad

Алгебраические уравнения решаются с помощью двух функций:

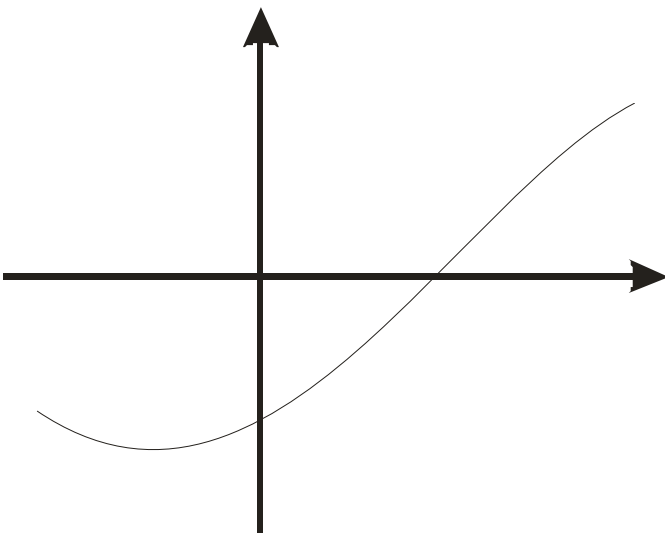
1. Root (f(x), x)
2. PolyRoots (v)

1. возвращает с заданной точностью значение переменной x, при которой выражение f(x)=0
2. возвращает вектор всех корней полинома n степени, коэффициент которого хранится в массиве v, длиной n+1

$$x^3 + 0.4x^2 + 0.6x - 1 = 0$$

$$v := \begin{pmatrix} -1 \\ 6 \\ -4 \\ 1 \end{pmatrix}$$

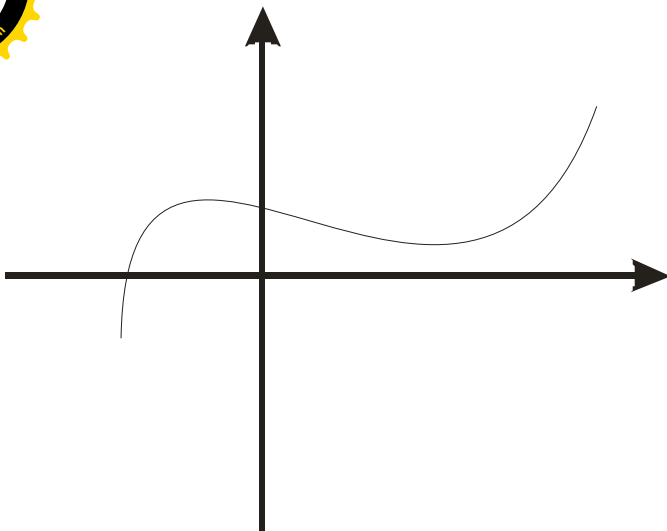
$$PolyRoots(v) = \begin{pmatrix} -0.558 + 1.043i \\ \dots\dots\dots \\ \dots\dots\dots \\ 0.715 \end{pmatrix}$$



Пример.

$$Y(x) = 0$$

$$Y(x) = x^5 - x^3 + 1$$



$x := -z$

Root (y (x), x) = -1.237

Для нахождения корней полинома можно использовать операцию Coeefs, Которая находится на панели №8.

Пример.

$$y(x) := x^4 - 18x^2 + 6$$

$$A := y(x) \text{coeefs}, x \rightarrow \begin{pmatrix} 6 \\ 0 \\ -18 \\ 0 \end{pmatrix} \text{PolyRoots}$$

$$\begin{pmatrix} -4 \\ -0.53 \\ 0.53 \\ 4 \end{pmatrix}$$



ЛЕКЦИЯ № 26

26. ИНТЕРНЕТ, ОБЩИЕ ПРЕДСТАВЛЕНИЯ. ОСНОВЫ РАБОТЫ В ИНТЕРНЕТЕ

Интернет (*Internet*) – это самая крупная (глобальная) компьютерная сеть в мире. Это сеть сетей в мировом масштабе.

Интернет рассматривается как единое целое образование и не принадлежит какому – либо государству, организации или частному лицу. Он представляет собой добровольное объединение множества локальных и территориальных сетей, принадлежащих коммерческим компаниям, научным и университетским центрам, правительственным учреждениям.

Координирующим органом сети является организация ISOC (Internet Society), состоящая из специалистов и заинтересованных лиц различных стран мира. Эта организация выбирает совет IAB (Internet Architecture Board – совет по архитектуре Интернета), который принимает решения, связанные с функционированием сети. Технические рекомендации совету готовит инженерная группа IETF (Internet Engineering Task Force – рабочая группа по инженерным проблемам Интернета).

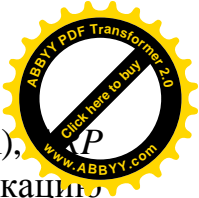
26.1. Основные понятия и определения

Компьютеры, подключенные к сети Интернет, обычно имеют различную архитектуру и различное программное обеспечение. Для достижения их совместимости при работе в сети, используются специальные *правила передачи информации по сети*. Эти правила называются *протоколами*.

26.2. Протоколы Интернет

Работа Интернет основана на использовании протокола TCP/IP (Transmission Control Protocol / Internet Protocol). Он объединяет целую группу других протоколов, каждый из которых решает свой набор задач. К этой группе относятся следующие протоколы:

- *транспортный протокол TCP*, который управляет процессом передачи данных между компьютерами;
- *протоколы маршрутизации IP, ICMP* (Internet Control Message Protocol), *RIP* (Routing Information Protocol), которые обрабатывают адресацию данных, обеспечивают их физическую передачу и отвечают за выбор наилучшего маршрута до адресата;



- *протокол поддержки сетевого адреса DNS (Domain Name System), (Address Resolution Protocol), обеспечивающий идентификацию компьютера по его уникальному адресу;*
- *иллюзовые протоколы EGP (Exterior Gateway Protocol), GCP (Gateway – togeteway Protocol), IGP (Interior Gateway Protocol) отвечают за передачу информации о маршрутизации данных и состоянии сети, а также обрабатывают данные для взаимодействия с локальными сетями;*
- *протоколы прикладных сервисов FTP (File Transmission Protocol), Telnet и другие сетевые программы, обеспечивающие доступ к различным услугам и службам Сети, например передачу файлов между компьютерами;*
- *другие важные протоколы, например, протокол SMTP (Simple Mail Transfer Protocol), отвечающий за передачу сообщений электронной почтой.*
-

26.2.1. IP – адреса и классы сетей

Каждый компьютер, подключенный к сети, должен иметь свой уникальный или как его еще называют *IP – адрес*. Этот адрес может быть *постоянным* или *динамическим (временным)*, назначаемым сервером при соединении с сетью.

IP – адрес образуется из четырех целых чисел, взятых из диапазона 0...255, и занимает объем памяти равный 4 байтам. Числа между собой отделяются точками. Например: **192.94.31.238**. При таком способе образования адресов можно создать более четырех миллиардов комбинаций IP – адресов.

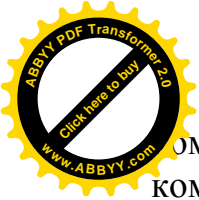
Постоянные IP – адреса устанавливаются для компьютеров, которые работают в сети *постоянно*.

Динамические IP – адреса устанавливаются компьютерам, которые подключаются к сети *эпизодически* и только *на время работы компьютера в сети Интернет*.

Компьютеры, получившие адреса, называются *головными* или *ведущими* (host – компьютерами). Host – ведущий. Каждый головной компьютер соединен с другими головными компьютерами сети.

К головным компьютерам могут подсоединяться другие компьютеры и терминалы, не имеющие своих IP – адресов. Эти компьютеры не являются полноправными членами сети Интернет, но и они могут получать доступ к Сети. Это возможно через *посредника*, в роли которого выступает *свой головной компьютер*. Такая схема подключения увеличивает время доступа, так как информация поступает не напрямую в сеть (или из сети), а с запоминанием на головном компьютере.

Головные компьютеры, находящиеся в распоряжении отдельных пользователей, в свою очередь, подсоединяются к более мощным головным



компьютерам локальной сети организации или учреждения, а те – к головным компьютерам территориальных или национальных сетей. Последние же связаны по всему миру.

26.2.2. Доменные адреса

IP – адреса используются сетевой аппаратурой для адресации и передачи информации между головными компьютерами Сети. Этот способ адресации удобен для сетевого оборудования, но не очень удобен для людей, работающих в Сети. Он труден в запоминании и не создает необходимых ассоциаций.

Поэтому параллельно с IP – адресами в Интернете введена другая, *доменная адресация*.

В этой системе адресации вся совокупность компьютеров сети Интернет разделена на большие группы, которые называются *доменами верхнего или первого уровня*. В эти группы включаются головные компьютеры сети по территориальному, национальному, коммерческому или по какому – либо другому признаку. Каждый домен верхнего уровня имеет собственное имя. Эти имена регистрируются в координирующем органе Сети (ISOC) и закрепляются на постоянной основе. Например:

au – Австралия,

be – Бельгия,

ru – Россия,

ua – Украина,

com – коммерческие организации (в основном США).

Каждый домен верхнего уровня, в свою очередь, может содержать произвольное количество доменов следующего (более низкого) – *второго уровня*. Обычно такие домены образуются из крупных региональных, территориальных или городских сетей. Например, в домене *ru* (Россия) имеются домены *altai* (Алтай), *vologda* (Вологда).

Домены второго уровня могут содержать домены еще более низкого *третьего уровня*. Таким образом, доменный адрес представляет собой *иерархию адресов*. Например, доменный адрес сервера ДонНТУ имеет вид:

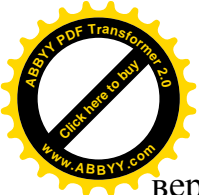
donntu . edu . ua

Он включает в себя следующие части:

donntu – домен третьего уровня, что означает компьютерную сеть ДонНТУ,

edu – домен второго уровня, что означает образовательную сеть,

ua – домен первого (верхнего) уровня, что означает территориальную сеть Украины.



В доменных адресах иерархия образуется слева направо, от нижнего уровня к верхнему. Следовательно, *каждый компьютер*, входящий в сеть Интернет, имеет *IP – адрес и однозначно связанный с ним доменный адрес*.

Имеется специальная таблица соответствия адресов, которая находится в специально выделенном компьютере, называемом *DNS – сервером* (Domain Name Server). Функцию *DNS – сервера* могут выполнять несколько компьютеров локальной или территориальной сети. Конечным пользователям достаточно знать только доменные адреса, так как они более понятны и запоминаемы.

26.3. Узел Интернета

Узел Интернета – это *организация*, которая подсоединяет отдельных пользователей к сети Интернет и имеет право выделять постоянные или динамические *IP – адреса*. Такую организацию еще называют *поставщиком услуг* или на английский манер *провайдером*.

В состав узла Интернета входит *сервер имен доменов* и *компьютер с таблицами маршрутов*, по которым для каждого компьютера определяется следующий ближайший узел Сети. Компьютеры с таблицами маршрутов называют *маршрутизаторами* или *роутерами (router)*. В принципе один и тот же компьютер может совмещать функции маршрутизатора и сервера имен доменов.

26.3.1. Получение доступа к Интернету

Существует два способа подключения компьютера к Интернету:

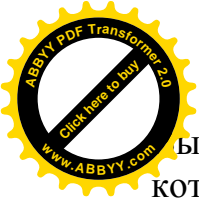
- *постоянный*,
- *временный*.

Постоянный способ отличается тем, что компьютер подсоединен к какой – либо локальной сети, например предприятия или университета, с помощью линии связи. При этом локальная сеть имеет постоянное соединение с Интернет. В этом случае, для получения доступа к сети Интернет необходимо обратиться к *администратору локальной сети* и получить от него:

- *входное имя*,
- *пароль*,
- *правила входа и работы в Интернет*.

Такие пользователи обычно работают не на головном компьютере Интернета, а на, подсоединенных к головному компьютеру, терминалах или персональных компьютерах.

Для подсоединения локальной сети к Интернет необходимо выбрать провайдера, то есть организацию, которая оказывает платные Интернет – услуги.



Выбор провайдера для конкретной сети производится, исходя из условий, в которых будет работать сеть, из условий оплаты услуг и скорости передачи данных. После заключения соглашения между провайдером и администратором сети производится подключение локальной сети к Интернет.

Для постоянного подключения компьютера его необходимо оснастить *сетевой платой* и *подсоединить с помощью сетевого кабеля* к локальной сети организации. Впрочем, это не означает, что пользователи всех компьютеров такой сети могут пользоваться Интернет постоянно. В целях разгрузки локальной сети некоторые компьютеры могут отключаться.

Временный способ состоит в подключении персонального компьютера к Сети с помощью *модема* и *телефонной линии связи*. Работа модема управляется специальным *PPP* – протоколом. Этим способом пользуется большинство владельцев домашних персональных компьютеров, которые имеют телефонные номера.

Для такого способа подключения владелец компьютера должен:

- *подобрать подходящего провайдера и заключить с ним соглашение об условиях работы,*
- *получить от провайдера:*
 - *имя пользователя (User),*
 - *пароль (password),*
 - *номер удаленного телефона, по которому он будет подключать свой компьютер к Интернет через собственную телефонную консоль.*

Кроме этого, пользователь может получить полную инструкцию по настройке своего компьютера для работы в Интернет или попросить сделать это специалистов провайдера.

После решения всех вопросов подключения необходимо настроить компьютер на связь с сервером провайдера и установить на компьютере необходимое программное обеспечение, например, Web – браузер, почтовый клиент, FTP – клиент, в зависимости от характера работ. После этих действий компьютер готов к работе с сети Интернет.



26.3.2. Вход в Интернет

Для входа в Интернет владелец компьютера должен настроить программу подключения к сети провайдера. Это удобно сделать с помощью *Мастера подключения к Интернету*, который является составной частью операционной системы Windows XP.

Для этого необходимо выполнить команды:

Пуск – Панель управления – Сетевые подключения – Файл – Новое подключение.

После этого будет запущен *Мастер новых подключений*.

В *первом* раскрывшемся окне установить переключатель **Подключить к Интернету** и *щелкнуть по кнопке Далее* этого окна.

Во *втором* раскрывшемся окне из трех предложенных методов подключения:

- с получением новой записи для работы в Интернете,
- вручную с использованием учетной записи, предоставленной провайдером,
- с помощью компакт – диска, предоставленного провайдером *Интернета*

установить переключатель **Подключение вручную** и *щелкнуть по кнопке Далее* этого окна.

В *третьем* раскрывшемся окне установить переключатель **через обычный модем** и *щелкнуть по кнопке Далее* этого окна.

В *четвертом* окне следует установить имя создаваемого соединения, указав его в строке **Имя подключения**, а затем *щелкнуть по кнопке Далее* этого окна.

В *пятом* окне необходимо *указать номер телефона провайдера* и *щелкнуть по кнопке Далее* этого окна.

В *шестом* окне следует указать *имя пользователя (login)* и его *пароль (password)* и *щелкнуть по кнопке Готово* этого окна.

После этого можно установить связь с провайдером. Для Windows XP следует выполнить команды:

– **Пуск – Подключение,**

– в раскрывшемся окне *выбрать имя нужного соединения* и *щелкнуть по кнопке Вызов* этого окна.

В результате этих действий компьютер начинает набор номера телефона для связи с сетью провайдера, затем идет процесс соединения, который сопровождается характерным «пением» модема. Все стадии соединения комментируются в окне подключения. Если данные указаны верно, то в течение минуты производится соединение компьютера с сервером провайдера и вход в



еть. С этого момента компьютер находится в Интернет, а сервер провайдера начинает отсчет времени (в секундах) нахождения компьютера в сети.

По окончании работы следует отключиться от сети Интернет. Для этого необходимо вызвать все то же окно подключения, которое теперь отсчитывает секунды и показывает количество принятой и переданной информации и щелкнуть по кнопке отключения от Сети.

26.4. Информационные ресурсы Интернет

Самим интересным, удобным и эффективным ресурсом Интернета, имеющим огромную популярность, является *Всемирная паутина* – WWW (World Wide Web).

WWW представляет собой *гипертекстовую информационную систему*, содержащую *связанные между собой ссылками документы*, которые могут создаваться в различных программных средах и находится в любом из компьютеров сети Интернет.

Не следует считать, что Интернет и WWW одно и то же.

Интернет – это линии связи, сетевая аппаратура, компьютеры, данные, программное обеспечение, коллективы специалистов, обслуживающих все это.

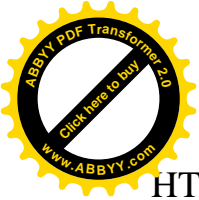
WWW – это только одна из возможностей Интернета, представляющая собой *совокупность связанных данных*, организованных в виде гипертекста и расположенных на серверах сети, то есть *информация*, размещаемая в сети Интернет.

В современных информационных системах фактическим стандартом стал мультимедийный характер информации, приведший к созданию *гипермедиа – систем*.

Информация, связанная с гипертекстовым полем какого – либо документа, может находиться на различных компьютерах Сети, расположенных в различных странах и даже континентах. Если пользователь щелкнет по гипертекстовому полю, то произойдет автоматическое обращение к нужному компьютеру Сети, и информация, связанная с этим полем, будет передана по сети к затребовавшему ее компьютеру.

Программы, с помощью которых просматриваются гипертексты в WWW, принято называть *браузерами* (browse – просмотр) или *обозревателями*. Браузеры автоматически осуществляют все необходимые операции по поиску и передаче по сети информации, связанной с гипертекстовыми полями.

WWW была создана в 1989 – 1991 годах швейцарским физиком Томом Бернесом Ли и его коллегой Робертом Кейлау. Они разработали специальный протокол *HTTP* (Hyper Text Transfer Protocol) – протокол передачи гипертекстов. Практически этот протокол представляет собой надстройку над протоколом TCP / IP.



В настоящее время используются обозреватели значительно мощнее, HTTP, такие, как Internet Explorer, Opera, Mozilla и др.

26.5. Web – страница

Основными представителями гипертекстовых документов являются *Web – страница* и *сайт*.

Web – страницей называется документ, содержащий гипертекстовые ссылки, базирующиеся на протоколе HTTP. Web – страница может иметь *любую длину* и представляет собой отдельный документ (файл), в котором имеются ссылки на связанные с ним другие документы. Для просмотра длинной *Web – страницы* ее можно прокручивать на экране участками.

Сайт или *Web – узел* представляет собой группу Web – страниц. Сайт может содержать произвольное количество Web – страниц. Его можно представить как обычную папку, содержащую группу как взаимосвязанных, так и не связанных документов.

Доступ на каждый сайт всегда производится через так называемую *начальную* (домашнюю, стартовую) *страницу*. Любой сайт имеет *только одну начальную страницу*. Просматривая ее, пользователь может получить исчерпывающее представление о содержании сайта и обратиться к любой его странице.

Сайты размещаются на компьютере узла Интернет, оснащенный специальной аппаратурой и программным обеспечением. Такой компьютер называют *Web – сервером*. Провайдеры должны иметь хотя бы один *Web – сервер*, иначе они не смогут обеспечить своим пользователям доступ к Всемирной Паутине.

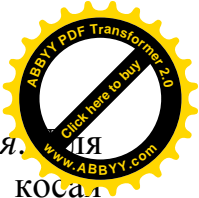
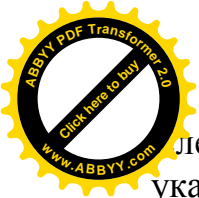
Доступ к различным ресурсам Интернет осуществляется по разным протоколам. Для упрощения работы пользователя создатели современных браузеров предусмотрели возможность запроса из Web – обозревателей не только Web – страниц, но и других наиболее популярных ресурсов Сети.

В последнее время появилась особая разновидность сайтов, называемых *порталами*. Они обеспечивают удобный доступ к самим разным ресурсам Сети и Паутины.

26.5.1. Адрес ресурса

Каждая Web – страница представляет собой файл, размещенный на одном из дисковых устройств Web – сервера. Следовательно, чтобы получить доступ к интересующей Web – странице, нужно указать на файл, содержащий эту Web – страницу. Имена файлов, используемых в сети Интернет, могут содержать произвольное количество символов, в том числе и точки. Например,

TextWin. Oct07.Zipx



следует иметь в виду, что в системе Интернет *регистры букв различаются*. Указание маршрута к файлу при отделении подкаталогов используется не косая черта \ как в Windows, а черта с обратным наклоном / (знак деления). Например,

`/Util/drivers/mouse.com`

Чтобы получить доступ из Web – браузера к тому или иному, связанному с файлом, ресурсу, в Интернете используются так называемые *унифицированные указатели ресурсов* – **URL** (Uniform Resource Locator).

Фактически URL является адресом запрашиваемого ресурса в Интернете или просто адресом ресурса.

URL – это адрес, который указывает на определенную страницу или ресурс во Всемирной паутине или в Internet в целом.

В общем случае URL имеет достаточно сложную структуру. В простейших случаях, при запросе наиболее популярных ресурсов, например, для доступа к начальным страницам сайтов, URL содержит:

- схему протокола, по которому обеспечивается доступ к ресурсу,
- доменный адрес компьютера, содержащего затребованный ресурс,
- спецификацию файла (путь к файлу и имя) или название ресурса.

Имя протокола, определяющего тип затребованного ресурса вместе с ограничивающими символами **://**, называют *схемой протокола*. Например, схема **http://** определяет запрос Web – страницы по протоколу HTTP, а схема **file://** используется для указания на файл, находящийся в компьютере пользователя. Пример URL:

http://sunsite.unc.edu/boutell/faq/www_faq.html

Здесь

http:// – протокол всемирной паутины,
sunsite.unc.edu – доменный адрес,
boutell/faq – путь к файлу,
www_faq.html – имя файла.

Судя по названию файла, он содержит Web – страницу.

26.6. Электронная почта

Электронная почта или *e – mail* (electronic mail – электронная почта) представляет собой один из способов связи. Любой пользователь Интернет может со своего компьютера послать сообщение на другой компьютер. При этом сообщение не будет сразу переправлено на компьютер – получатель, а вначале будет помещено в «почтовый ящик», где оно будет находиться до тех пор, пока его не прочтет пользователь компьютера – получателя при просмотре почты. Такие сообщения передаются за весьма короткое время (от нескольких секунд до



нескольких минут). Кроме текстовых сообщений электронной почтой можно передавать аудио – и видеоматериалы, чертежи, фотографии. В роли «почтового ящика» используется специальный компьютер, называемый *почтовым сервером* (mail server), в котором и хранится электронная почта. Этот компьютер постоянно подсоединен к сети Интернет и обрабатывает все входящие и исходящие сообщения. Доступ к почте можно получить только после указания имени и пароля пользователя – получателя.

Для отправки электронных сообщений по сети Интернет используется протокол *SMTP* (Simple Mail Transfer Protocol – простой протокол электронной почты). Для приема сообщений используют такие протоколы как *POP* (Post Office Protocol – почтовый протокол) или *IMAP* (Internet Message Access Protocol – протокол доступа к сообщениям Интернета).

Для реализации электронной почты используются почтовые программы. В настоящее время используются такие программы как Internet Mail, Microsoft Outlook, Microsoft Exchange. The Bat и др. В стандартный комплект операционной системы Windows XP входит мощная почтовая программа Outlook Express 6.0.

26.7. Сообщения

Основным понятием электронной почты является *сообщение*. Под сообщением понимают *текст*, передаваемый по линиям связи от одного пользователя к другому. Проще говоря, сообщение – это электронный аналог обычного письма.

По требованиям протокола TCP / IP текст сообщения должен состоять из символов кода ASCII. Это ограничение вызвано тем, что сообщение проходит на своем пути большое количество компьютеров с различным аппаратным и программным обеспечением.

Сообщения могут иметь приложения в виде чертежей, фотографий, аудио– и видеозаписей, которые называют *присоединенными файлами*.

Различают два вида сообщений:

- *входящие*,
- *исходящие*.

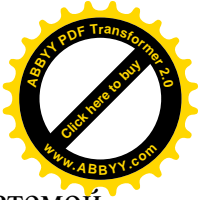
Входящие – это те сообщения, которые *пользователь получает*.

Исходящие – это те сообщения, которые *пользователь отправляет*.

Структура сообщений

Каждое сообщение состоит из:

- *заголовка*,
- *тела сообщения*,
- *подписи отправителя*.



Заголовок включает:

- *идентификатор сообщения*, который формируется почтовой системой автоматически;
- *адрес отправителя* (формируется автоматически);
- *адрес получателя* или *адреса получателей*, который задается пользователем;
- *тема сообщения* (задается пользователем или формируется автоматически);
- *время и дата отправления* (формируется автоматически);
- *информация о маршруте*, по которому пришло сообщение от отправителя к адресату (некоторыми системами формируется как дополнительная информация).

Для своевременной и точной доставки сообщения пользователю достаточно указать *правильный адрес электронной (e – mail) почты получателя*.

Адрес электронной почты состоит из двух частей:

- *кому*,
- *куда*.

Часть «*кому*» представляется учетным именем получателя, а часть «*куда*» представляется доменным адресом компьютера – получателя. Эти две части адреса разделяются символом @.

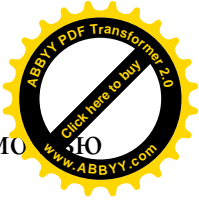
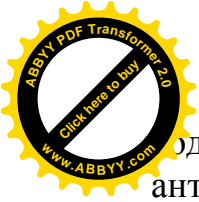
Например:

askbill@microsoft.com

Здесь первая часть («*кому*») askbill представляет собой учетное имя пользователя, а вторая часть («*куда*») microsoft.com является доменным адресом компьютера. Таким образом, чтобы направить кому – либо сообщение по Интернету, достаточно знать его электронный адрес.

Тело сообщения – это направляемый адресату текст. В сообщение, кроме основного текста, может быть включена так называемая *сигнатура* (signature – подпись) или *файл подписи*. Файл подписи может содержать любые о себе сведения, которые пожелает включить отправитель: имя, почтовый или электронный адрес, телефонный номер, образец подписи, небольшую картинку, составленную из символов и др.

Присоединенные файлы – это файлы различного содержания, подключаемые в дополнение к тексту сообщения. Присоединение файлов в различных программах электронной почты осуществляется различными способами. Чаще всего требуется только указания спецификации присоединяемого файла (путь, имя) в специальной команде или в специальном поле. Следует также помнить о возможном заражении вирусами присоединенных файлов. Поэтому, получив по почте сообщение,



содержащее присоединенные файлы, следует их проверить с помощью антивирусных программ.

ЛЕКЦИЯ № 27

27. ЭЛЕКТРОННАЯ ПОЧТА. ОБОЗРЕВАТЕЛИ СЕТИ ИНТЕРНЕТ

27.1. Почтовый сервер

Программы, обеспечивающие работу электронной почты в Интернет, построены по принципу *клиент – сервер*. Любое сообщение, отправленное пользователем сети Интернет, сначала попадает на специальный компьютер узла Интернет, называемый *почтовым сервером*. В дальнейшем почтовый сервер обеспечивает отправку *исходящих сообщений* по наиболее подходящим маршрутам. Кроме этого, почтовый сервер производит *прием* всех *входящих сообщений* и *хранит* их до прочтения адресатами, обслуживаемыми этим сервером.

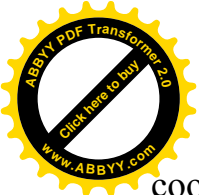
Обычно все отправляемые пользователями исходящие сообщения и принимаемые входящие сообщения не сразу передаются по сетевым линиям к своим адресатам. Некоторое время они хранятся в почтовом сервере узла Интернет. Поэтому почтовый сервер должен иметь определенный размер дискового пространства или обладать определенным ресурсом. Именно ресурс почтового сервера устанавливает максимальный объем сообщений.

Серверная часть пакета электронной почты обычно состоит из трех основных подсистем:

- *подсистемы хранения сообщений,*
- *транспортной подсистемы,*
- *службы каталогов.*

Подсистемы хранения сообщений обеспечивает хранение *входящих сообщений* до прочтения их пользователями. Чтобы входящие сообщения попадали строго только к своим адресатам, на почтовом сервере ведется учет всех обслуживаемых пользователей. На каждого пользователя заводится *учетная запись*, содержащая его *учетное имя* и другую информацию. Каждому пользователю выделяется *личный почтовый ящик*, где хранятся адресованные ему сообщения. *Учетное имя пользователя* одновременно является и *именем его почтового ящика*.

Почтовый ящик представляет собой отдельный файл или папку на магнитном диске или же запись в базе данных сервера. Входные сообщения хранятся в почтовом ящике до тех пор, пока их не прочтет пользователь со своего компьютера.



Транспортная подсистема обеспечивает пересылку как исходящих сообщений, отправляемых пользователями данного узла Интернет, так и транзитных сообщений, получаемых из других узлов Интернет, по наиболее оптимальным маршрутам.

Служба каталогов обеспечивает хранение и корректировку учетных записей пользователей электронной почты, а также направление сообщения строго по указанному адресу.

27.2. Операции с исходящими сообщениями

Под операциями с исходящими сообщениями понимают:

- *подготовку сообщений,*
- *адресацию сообщений,*
- *присоединение файлов к сообщениям,*
- *отправку сообщений.*

Подготовка текста сообщения обычно производится с помощью текстового редактора электронной почты, возможности которого несколько скромнее, чем текстового редактора MS Word. Поэтому перед отправкой сообщения следует внимательно проверить орфографию и пунктуацию текста.

Адресация сообщения предполагает указание точного адреса получателя. Некоторые пакеты электронной почты обеспечивают пользователя *адресной книгой*, являющейся электронным аналогом записной книжки, куда заносятся нужные адреса и краткие комментарии.

В общем случае одно и то же сообщение может направляться сразу по нескольким адресам. Чтобы это сделать, необходимо сформировать *список рассылки*, перечислив в нем адреса всех получателей. Если имеется группа адресов, по которым регулярно производится рассылка сообщений, то можно составить список таких адресов, записать один раз на диск и установить ему имя. В дальнейшем вместо перечисления адресов рассылки достаточно *указывать только имя этого списка*.

В тех случаях, когда требуется ознакомить с текстом сообщения большой круг лиц (реклама, объявление и т. п.), то такое сообщение помещают на специальную *электронную доску объявлений*, к которой имеют доступ все желающие пользователи сети Интернет.

В некоторых почтовых системах можно устанавливать режим пересылки сообщений, указав его приоритет: *низкий, стандартный* или *высокий*. В зависимости от приоритета пересылки сообщения почтовый сервер определяет степень срочности отправления сообщения (задержать отправку до следующего сеанса связи или отправлять немедленно).



Вместо указания приоритета отправки сообщения *можно указывать* время его отправки, например, чтобы оно было отправлено во время действия льготного тарифа.

Можно *установить режим уведомления* о доставке сообщения в почтовый ящик адресата или режим уведомления о прочтении сообщения.

Пользователь может *установить режим сохранения* копии посланного сообщения на почтовом сервере узла или на своем компьютере.

Обычно отправка сообщений производится с помощью стандартной команды отправки почты.

27.3. Операции с входящими сообщениями

К операциям с входящими сообщениями относятся:

- *оповещение о прибытии почты,*
- *чтение почты,*
- *подготовка ответа,*
- *переадресация,*
- *организация хранения и удаления сообщения.*

После того, как получаемое сообщение попадает в почтовый ящик пользователя, производится оповещение о прибытии сообщения. Существует несколько видов оповещения:

- *по запросу пользователя,*
- *при подключении к системе,*
- *в момент поступления.*

Независимо от наличия или отсутствия средств оповещения и режимов их работы, поступившее сообщение хранится в почтовом ящике пользователя до его прочтения. Запустив почтовую программу на своем компьютере, пользователь получит полный список прибывших на его адрес сообщений с указанием даты и времени поступления, имени отправителя и темы сообщения. Прочитанные сообщения определенным образом отмечаются. Некоторые почтовые программы позволяют указывать пользователю срок хранения сообщения, по истечении которого оно может удаляться. Прочитанные сообщения можно *оставлять на хранение*. Для систематизации сохраняемых принятых сообщений можно создавать папки.



27.4. Электронные доски объявлений

Электронные доски объявлений, BBS (Bulletin Board System – система досок объявлений) используются, как уже ранее отмечалось, для размещения сообщений с целью ознакомления ими широкого круга пользователей.

Самой большой глобальной сетью электронных досок объявлений является всемирная сеть *Usenet*, доступ к которой имеется и из сети Интернет. Для получения доступа из сети Интернет к сети Usenet необходимо, чтобы в состав узла Интернет входил *сервер Usenet*, называемый *сервером новостей*.

Для удобства работы с большой массой объявлений все они разделены на *группы новостей* (newsgroup). Таких групп в сети Usenet насчитывается несколько тысяч. Каждая группа имеет свое имя, по структуре похожее на доменный адрес. Группы формируются по самим различным признакам (группа *bionet* – биология и генетика, группа *comp.protocols* – сетевые компьютерные протоколы, группа *rec.arts* – отдых и искусство и т. д.). Выбрав нужную группу новостей, можно прочитать в них любые сообщения и послать в эту же группу свое сообщение. Весьма вероятно, через некоторое время, встретить в этой группе ответ на свое сообщение. Таким образом, можно осуществлять обмен информацией, вести дискуссию.

В связи с огромным потоком ежедневной информации, поступающей через каналы Usenet, многие администраторы узлов Интернета резко ограничивают количество доступных групп новостей. Поэтому для получения доступа к интересующей группе следует сообщить о своем желании администратору «своего» узла Интернет (своему провайдеру).

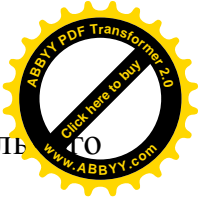
Современные Web – обозреватели обеспечивают доступ к электронным доскам объявлений с помощью URL, которые имеют структуру

news: имя_ группы.

Например, <news:rec.sport> обеспечит доступ к группе новостей «отдых и спорт».

27.5. Телеконференции

С использованием программ электронной почты и других специальных пакетов можно проводить деловые совещания, научные телеконференции, в которых могут участвовать группа пользователей, находящихся в различных странах и даже на различных континентах. Способы организации таких мероприятий могут быть различными. В простейшем случае каждый участник конференции размещает на экране своего дисплея текст, документ или изображение. В других случаях на экране высвечивается обмен репликами с указанием авторов реплик. Наиболее



ожные системы позволяют проводить видеоконференции в режиме реального времени.

27.6. Пересылка файлов

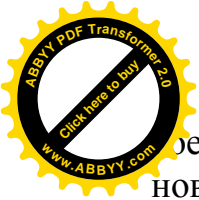
Ранее отмечалось, что к сообщениям, пересылаемым электронной почтой, можно подсоединять любые файлы, но только в автономном (offline) режиме.

В Интернет существует другой способ пересылки файлов между компьютерами. Он основан на протоколе *FTP* (File Transfer Protocol – протокол передачи файлов), который обеспечивает передачу файлов в *оперативном* (online) режиме. Это означает, что при передаче файлов передающий и принимающий компьютеры должны находиться одновременно на связи, то есть *в прямом контакте*. Не каждый головной компьютер Сети может пересылать файлы в таком режиме. Это могут делать только компьютеры, обладающие специальным программным обеспечением, позволяющим пересылать файлы по протоколу *FTP*. Такие компьютеры называют *FTP – серверами*. Администрация узлов Интернет, которые имеют FTP – серверы, обычно регистрирует пользователей, имеющих разрешение на доступ к файлам сервера с правом их копирования на свои компьютеры. Таким пользователям назначают специальные пароли, без которых доступ к FTP – серверу невозможен. Однако в Интернет есть узлы, доступ к FTP – серверам которых разрешен всем пользователям Сети. Такой доступ называется *анонимным*.

Если пользователь знает доменный адрес анонимного FTP – сервера или имеет разрешение на доступ к коммерческому FTP – серверу, на дисковом устройстве которого размещен нужный файл, то для его копирования на собственный компьютер с помощью Web – обозревателя достаточно указать стандартный URL с именем протокола *ftp*. Например, для копирования файлов из FTP – сервера библиотеки университета штата Вашингтон следует указать адрес ресурса <ftp://wuarhive.wustl.edu>. В этом адресе не содержится спецификация конкретного файла. В таких случаях Web – обозреватель подключается к указанному в доменном адресе FTP – серверу и отображает его корневой каталог. Для пересылки конкретного файла следует перейти в содержащий его подкаталог и принятым в обозревателе способом скопировать его в свой компьютер.

27.7. Интернет – пейджеры

В Интернет имеется весьма важная услуга, обеспечивающая общение в реальном времени – Интернет – пейджеры. Она обеспечивает возможность фиксации факта подключения к Интернет нужного пользователя и оповещает об этом всех или отдельных клиентов Сети с целью установления оперативной связи между ними. Для работы в таком режиме необходима специальная программа. Она устанавливается на компьютере пользователя и при подключении к Интернет



соединяется с сервером сообщений, проверяя время от времени, не пришли ли новые сообщения.

Существует несколько таких программ, но наибольшую популярность получила программа *ICQ*. Название этой программы не является аббревиатурой, состоящей из начальных букв каких-то слов. Она просто созвучна с английскими словами *I seek you* (я ищу тебя), называемая в просторечье «аська». Первая версия этой программы была разработана в 1998 году компанией *Mirabilis*. Наряду с этой программой существует программа *Windows Messenger*, которая входит в пакет программ *Windows XP*.

27.8. Базы данных в Интернет

К Интернету подключено множество баз данных, содержащих огромное количество информации по самым различным вопросам. Как правило, они входят в состав распространенной информационной системы *WAIS* (*Wide Area Information System*).

Компьютер, имеющий специальное программное обеспечение и предоставляющий доступ к базам данных системы *WAIS*, называют *wais – сервером*. Система *WAIS* объединяет *wais – серверы* всего мира, которые имеют доступ к более чем 1000 общедоступных и коммерческих баз данных.

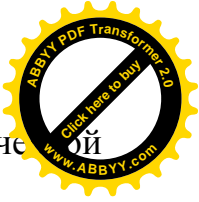
Для доступа к системе *WAIS* необходимо знать доменный адрес какого – либо *wais – сервера*. Этот адрес указывается в URL со схемой *http://*, в результате чего пользователь получает доступ к *web – странице* с перечнем баз данных, доступных из этого *wais – сервера*. С помощью адреса *http://wais.com* можно обратиться к центральному серверу компании *WAIS Inc.* и получить доступ к каталогу всех общедоступных баз данных *WAIS*.

27.9. Обозреватели сети Интернет

Всемирная компьютерная паутина *World Wide Web (WWW)*, первоначально задуманная как средство обмена научной информацией, в настоящее время прочно вошла в повседневную жизнь миллионов людей нашей планеты.

Для перемещения по *WWW* необходимо на компьютере пользователя установить *Web – клиент – программу*. Такая программа играет роль экскурсовода по Сети и называется *обозревателем* или английским словом *браузер*. Существует множество таких программ, каждая из которых имеет свои особенности. Наиболее распространенными среди них являются *Internet Explorer*, которая входит в пакет программ *Microsoft Windows*, *Opera* и *Mozilla*.

Opera – это программа – обозреватель, которая имеет сравнительно небольшой объем (размер версии *Opera 7.11* – 3.3 Мбайт), высокую скорость



работы и невысокие требования к ресурсам работы. Opera является коммерческой программой. Она работает в полном объеме без регистрации.

Mozilla – свободно распространяемый программный продукт. В состав *Mozilla* входит браузер, почтовый клиент, средства обмена мгновенными сообщениями, редактор web – страниц и календарь. К недостаткам следует отнести более высокие требования к аппаратным ресурсам компьютера.

27.9.1. Обзореватель Internet Explorer

Подключение и интерфейс программы Internet Explorer

После подключения к сети Интернет производится запуск программы Internet Explorer. Это осуществляется с помощью команд

Все программы – Internet Explorer

После запуска программы браузер сразу начнет загрузку домашней страницы, то есть web – страницы одного из web – узлов, на которую он настроен по умолчанию.

Internet Explorer располагает широким набором средств (меню, команд и кнопок) для управления работой программы, которые расположены в верхней части окна.

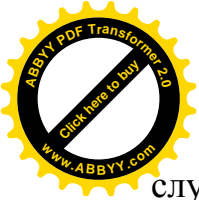
Меню – расположено в верхней строке окна и содержит перечень команд (*Файл, Правка, Вид, Избранное, Сервис, Справка*), которые служат для управления работой браузера.

Ниже расположены *кнопки управления* браузера: *Назад, Вперед, Остановить, Обновить, Домой, Поиск, Избранное, Журнал, Медиа, Во весь экран, Почта, Печать, Править в Microsoft FrontPage, Кодировка, Предварительный просмотр.*

Далее следует *адресная строка*. Она представляет собой строку ввода и начинается со слова «*Адрес:*», после которого пользователь может ввести с клавиатуры доменное имя адресата либо его *IP* – адрес. Если адрес вводится вручную, то он запоминается и при необходимости может быть вставлен в строку ввода из списка, который открывается кнопкой, расположенной справа от строки ввода. В этом окне всегда содержится адрес текущей страницы.

Ссылки – это панель, содержащая группу адресов некоторых стандартных Web – узлов, назначение которых понятно из надписей на кнопках. При необходимости эти кнопки можно заменить содержимым из подкаталога папки «*Избранное*», имеющего ссылки на наиболее интересные страницы, вносимые пользователем.

Все панели или каждую в отдельности можно убирать с экрана или перемещать по экрану с целью улучшения обзора Web – страниц, располагаемых под панелями.



В нижней части рабочей зоны, как правило, находится *строка состо.* служащая для отображения краткой оперативной информации о происходящем в программе в текущий момент.

Стандартные команды управления внешним видом окна обозревателя сосредоточены в меню **Вид**. При необходимости внесения некоторых изменений в настройку программы следует выбрать команду

Сервис – Свойства обозревателя

и в раскрывшемся многостраничном диалоговом окне настройки сделать необходимые установки.

Запрос ресурса

Ресурсы Интернет могут быть запрошены с помощью своих URL – адресов. При этом запрашиваемые ресурсы могут находиться на дисковых устройствах компьютера пользователя, в его локальной сети или на любом Web – сервере сети Интернет.

Существует несколько способов запроса нового ресурса по его адресу. Самый простой из них заключается в указании его адреса в адресной панели. Для этого необходимо:

- щелкнуть по любому участку поля на панели **Адрес**,
- ввести адрес ресурса,
- нажать клавишу **Enter**.

Если запрошенный ресурс доступен, он будет предоставлен пользователю в рабочей зоне окна. Если передача запрашиваемого ресурса затягивается и пользователю по каким – либо причинам (например, финансовым) желательно прекратить процесс передачи, то следует воспользоваться клавишей **Esc**, кнопкой **Остановить** на панели инструментов или командами **Вид – Остановить**. В этом случае передача ресурса будет немедленно прервана, и пользователь сможет возобновить работу с обозревателем, запросив другой ресурс или другую его функцию. В том случае, когда ресурс недоступен, программа извещает пользователя об этом. В таких случаях пользователю следует проверить установлена ли связь с узлом Интернет, а также убедиться в правильности указанного адреса ресурса.

Кодировка текста Web – страницы

Получаемые по запросу Web – страницы, могут содержать текст, представленный в различных системах кодировки. Как правило, используются системы кодировки ASCII, KO18, Windows 1251, MAC и некоторые другие. В некоторых случаях код текста, полученной Web – страницы, может не совпадать с кодировкой, используемой обозревателем, установленным на компьютере пользователя. В таких случаях пользователь вместо нормального текста на английском, русском или другом общепринятом языке увидит странные и непонятные символы. В стандартной настройке обозреватель Internet Explorer осуществляет автоматический выбор кодировки, который в большинстве случаев



казывается правильным. Для включения этого режима следует выполнить команду главного меню программы

Вид – Кодировка – Автовыбор

Если и в этом случае на экране будет бессмысленный текст, то следует воспользоваться командами

Вид – Кодировка – Дополнительно,

раскрыть список доступных кодировок и попытаться методом подбора выбрать нужную кодировку.

Перемещение по гиперссылкам

Особенностью Web – страниц является наличие гипертекстовых ссылок. Гиперссылка или просто ссылка может быть фрагментом текста, который всегда отличается цветом от остального текста и обычно подчеркивается или же значком или рисунком. Указывая на ту или иную ссылку можно просмотреть связанный с нею текст, прослушать аудиозапись, просмотреть какое-либо изображение, видеофрагмент, перейти к другой Web – странице.

Чтобы воспользоваться нужной ссылкой, достаточно щелкнуть по ней. Если указатель мыши установлен правильно на выбранную ссылку, то он принимает форму кисти руки с вытянутым указательным пальцем, что символизирует указание на что-то.

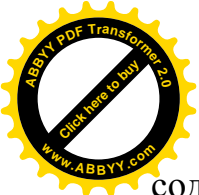
Сразу же после загрузки Web – страницы первая по порядку ссылка страницы окажется выделенной. Признаком выделения обычно служит штриховая рамка, окружающая ссылку, или изменение цвета.

Выделение ссылки можно произвести нажатием клавиши **Tab**. Каждое очередное ее нажатие приводит к перемещению выделения на следующую по порядку ссылку. Перемещение выделения в обратном направлении можно производить нажатием клавиш **Shift + Tab**.

После выделения ссылки необходимо вывести на экран связанную с ней информацию. Это делается нажатием клавиши **Enter**. Если связанная со ссылкой страница доступна, то в строке состояния отображается информация об обнаружении Web – узла и указывается IP – адрес компьютера, с которым устанавливается связь. После установления связи начинается пересылка запрошенной страницы на компьютер пользователя. Признаком завершения ее пересылки является появление в строке состояния слова **Готово**, а в поле **Адрес** отображения адреса вызванной страницы.

Запрошенная с помощью ссылки страница, заменяет собой текстовую страницу в рабочей зоне окна обозревателя. *Чтобы отобразить запрошенную информацию в новом окне*, необходимо, после совмещения указателя мыши со ссылкой, щелкнуть правой клавишей мыши и в появившемся контекстном меню выбрать команду **Открыть в новом окне**.

Для получения дополнительной информации о вызванной странице (размер, время создания и др.) следует выбрать команду **Файл – Свойства**.



Адрес вновь открытой страницы фиксируется в журнале обращения, содержащем список адресов Web – страниц и Web – узлов, которые открывались в обозревателе в течение нескольких последних дней.

Для просмотра адресов всех посещавшихся в течение последнего месяца Web – страниц и Web – узлов *необходимо обратиться к панели журнала*. Чтобы это сделать, следует выполнить одну из операций:

- щелкнуть по кнопке **Журнал** на панели инструментов,
- нажать сочетание клавиш **Ctrl + Н** на клавиатуре,
- выполнить команды **Вид – Панели обозревателя – Журнал**.

Чтобы вернуться к начальной странице исходного сайта, необходимо воспользоваться одной из операций:

- щелкнуть по кнопке **Домой** на панели инструментов,
- использовать сочетание клавиш **Alt + Home**,
- выполнить команды **Вид – Переход – Домашняя страница**.

Папка Избранное

Для хранения страниц, представляющих интерес для пользователя, существует специальная папка **Избранное**. С помощью этой папки пользователь может собирать адреса ресурсов и быстро вызывать любые потребовавшиеся ему Web – страницы независимо от времени их последнего посещения.

Для добавления адреса текущей страницы в эту папку необходимо:

- выполнить команды **Избранное – Добавить в избранное**,
- в раскрывшемся диалоговом окне в поле **Имя** ввести название, под которым программа должна сохранить сведения о текущей странице,
- щелкнуть по кнопке **ОК**.

Для просмотра содержимого папки следует щелкнуть по кнопке **Избранное** на панели инструментов.

Ссылку на текущую страницу можно поместить на панели ссылок, размещенной справа от адресной строки Internet Explorer. Чтобы открыть страницу, на которую имеется ссылка в панели, достаточно щелчком по кнопке >> справа от слова **Ссылки** развернуть панель, а затем щелкнуть по самой ссылке.

Сохранение Web – страницы и печать

Размещенную на экране, полученную по Сети Web – страницу, можно не только просматривать, но и сохранять на своем компьютере. Существует несколько видов сохранения страницы:

- сохранение Web – страницы полностью со всеми графическими, анимационными и мультимедийными элементами,
- сохранение только текстовой части Web – страницы в формате Web – страницы,



- сохранение только текстовой части Web – страницы в формате текстового файла.

В программе Internet Explorer предусмотрены некоторые возможности сохранения отдельных изображений Web – страницы. Изображения могут быть скопированы в буфер обмена или в файл на диск компьютера пользователя.

Для копирования в буфер обмена следует открыть контекстное меню выбранного рисунка и выделить в нем команду Копировать.

Чтобы сохранить такой рисунок в файле, следует выбрать в контекстном меню рисунка команду Сохранить как, а в раскрывшемся окне указать путь к месту расположения рисунка и имя, под которым он должен сохраняться.

Перед направлением на печать Web – страницы следует предварительно установить параметры страницы (размер бумаги, ориентацию листа, поля и др.). Способы их установки аналогичны операциям, выполняемым в текстовом редакторе MS Word. Далее следует оценить ее вид в режиме предварительного просмотра, который устанавливается с помощью команд

Файл – Предварительный просмотр.

Если установленная форма страницы удовлетворяет, то можно отправлять на печать одним из известных способов:

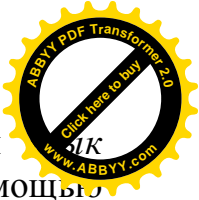
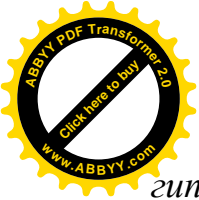
- с помощью команд **Файл – Печать** главного меню,
- щелчком по кнопке **Печать** на панели инструментов.

ЛЕКЦИЯ №28

28. СОЗДАНИЕ WEB – СТРАНИЦ. ПОИСКОВЫЕ СИСТЕМЫ

28.1. создание Web – страниц

Основной информационной единицей данных, передаваемых по Всемирной Паутине, является Web – страница. Известно, что сеть Интернет соединяет большое количество компьютеров с самым различным аппаратным и программным обеспечением. В этом случае очевидной является ситуация, когда Web – страница создается на компьютере с одним аппаратным и программным обеспечением, а просматривается на компьютере с совершенно другими средствами. В этой связи возникает задача выбора такого способа создания, хранения и передачи информации, при использовании которого Web – страница передавалась бы с высокой скоростью по всем линиям связи Интернет и воспринималась бы практически одинаково в любой программной и аппаратной среде.



Средством, обеспечивающим выполнение этих требований, является *гипертекстовой разметки HTML* (Hyper Text Markup Language), с помощью которого создаются Web – страницы. Это язык Всемирной Паутины. Он позволяет создавать Web – страницы даже с помощью самых простых текстовых редакторов. Язык *HTML* не является языком в буквальном смысле, со своим алфавитом, набором слов и фраз. Он представляет собой набор инструментов форматирования текста, основу которого составляют *HTML* – коды или, как их еще называют, *теги* (tag – ярлык, этикетка). *HTML* – документ отличается от простого текстового документа наличием кодов разметки (тегов). Теги позволяют устанавливать статус отдельных фрагментов текста (заголовок, собственно текст, рисунок и т. д.), осуществлять выбор шрифта, выбор цвета шрифта, местоположение и форматы включаемых рисунков, анимационных и мультимедийных элементов, гиперссылок, выбирать способы выравнивания абзацев и др. Это набор значков и небольшого количества стандартных слов.

Название тега и его параметры заключаются в угловые скобки < >. При создании любого *HTML* – документа обязательно должны использоваться теги заголовков и теги, делящие документ на отдельные логические части.

Каждый *HTML* – документ должен начинаться с *тега начала* <HTML> . Наличие такого тега говорит о том, что данный документ является *HTML* – документом. Заканчиваться *HTML* – документ должен *тегом окончания* </HTML>, в котором находится черта /. Например,

<HTML>

Первый *HTML* – документ

</HTML>

Теги <HTML> и </HTML> – это парные теги и их еще называют тегами – контейнерами, так как между ними обязательно содержится какая – то информация. Следует отметить, что *HTML* – коды нечувствительны к регистру. То есть запись <html> и запись <HTML> *совершенно идентичны*.

Каждый *HTML* – документ должен состоять из двух логических частей:

- *заголовка*,
- *тела*.

Заголовок идентифицируется с помощью тега <HEAD>, а тело – с помощью тега <BODY>. Например:

<HTML>

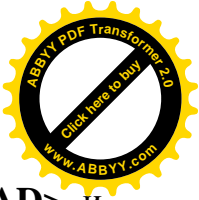
<HEAD>

заголовок документа

</HEAD>

тело HTML – документа

</HTML>



В этом примере заголовок обозначается с помощью пары тегов **<HEAD>** и **</HEAD>**, но для полного завершения *HTML* – документа следует обозначить его тело. Тело обозначается путем заключения между тегами **<BODY>** и **</BODY>**. С учетом этого предыдущий пример представляется в виде:

```
<HTML>  
    <HEAD>  
        заголовок документа  
    </HEAD>  
    <BODY>  
        тело HTML –документа  
    </BODY>  
</HTML>
```

Кроме этих тегов обязательными тегами *HTML* – документа являются теги **<TITLE>** и **</TITLE>**. Между этими тегами помещается название документа. Если название документа не представить между этими тегами, то оно не будет обозначаться при представлении *Web* – страницы, а вместо этого в строке названия будет указываться *URL* – адрес страницы. С учетом этого предыдущий текст можно скорректировать как

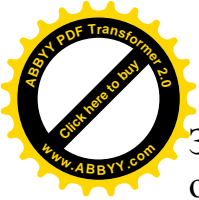
```
<HTML> <HEAD> <TITLE> название документа </TITLE> </HEAD>  
    <BODY>  
        тело документа  
    </BODY>  
</HTML>
```

В этом примере несколько тегов записано подряд, что допускается в *HTML*. Эти теги определяют структуру документа, и они *обязательны*.

Кроме этих тегов имеются и другие, с помощью которых можно придавать желаемую форму создаваемому документу. Программа – обозреватель, получив по линиям связи такой документ, интерпретирует его теги и отображает документ в виде *Web* – страницы (без тегов).

Отдельные приложения пакета *MS Office* такие как *MS Word*, *MS Excel*, *MS Power Point* обладают средствами сохранения созданных в них документов в виде *Web* – страниц. Чтобы преобразовать такой документ в форму *HTML* – документа, который может быть передан по Сети, следует выполнить команды:

Файл – Сохранить как веб – страницу.



Затем эту Web – страницу можно переправить на Web – сервер и стандартным образом запрашивать и просматривать с помощью программ – обозревателей. Для создания более сложных Web – страниц, а также для создания сайтов, состоящих из нескольких страниц, в текстовом редакторе MS Word предусмотрены шаблоны и мастер Web – страниц.

28.2. Поиск Web – страниц в Паутине

По оценкам специалистов в настоящее время на Web – серверах Интернета размещено несколько миллиардов Web – страниц, содержащих информацию практически во всех областях человеческой деятельности. Получение доступа к нужной информации в том случае, если известен адрес Web – страницы или Web – узла, не вызывает затруднений. Для этого достаточно ввести нужный адрес в соответствующее поле Интернет – обозревателя, и через некоторое время страница отобразится в его окне. В тех случаях, когда адрес размещения нужной информации неизвестен, необходимо производить ее поиск.

В настоящее время существуют системы, обеспечивающие быстрый поиск нужной информации во Всемирной Паутине. Существует несколько десятков различных *поисковых систем*, обладающих различными возможностями поиска:

- *по заданию образцов поиска,*
- *по созданию запросов,*
- *разными множествами просматриваемых узлов и страниц.*

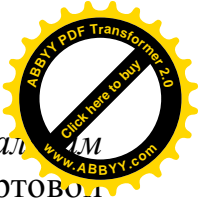
Фактически поисковая система (*поисковик*) является *специализированным Web – сервером*, на котором постоянно накапливается информация о Web – страницах и сайтах, находящихся на обычных Web – серверах Интернета. Эта информация в специальном сокращенном виде сохраняется и периодически обновляется на дисковых устройствах системы. Любой пользователь может обратиться к поисковому серверу и сформулировать запрос на поиск нужной информации. Поисковая система просматривает свои диски, находит и пересылает пользователю обнаруженные адреса Web – страниц, которые содержат требуемую информацию.

Все поисковые системы делятся на две большие группы:

- *поисковые каталоги,*
- *поисковые машины.*

Поисковые каталоги представляют собой огромные базы данных, которые содержат адреса Web – узлов по всему информационному пространству Интернета. Эти базы данных созданы и периодически обновляются специалистами в области систематизации Web – ресурсов.

Каталоги представляют собой иерархическую, многоуровневую структуру (рубрикаторы) и обеспечивают два способа поиска.



Первый способ аналогичен поиску путем перемещения по каталогам (подкаталогам) в окне проводника операционной системы Windows. На стартовой странице поискового каталога выбирается тема, рубрика верхнего уровня. После щелчка по названию выбранной темы раскрывается следующий (более низкий) уровень со списком разделов. В нем выбирается более узкая тема и так далее, перемещаясь по иерархии каталога, производится поиск необходимой информации.

Второй способ предусматривает выбор нескольких ключевых слов, отображающих смысл запрашиваемой информации. Он напоминает поиск в документе редактора MS Word. По этому способу ключевые слова вводятся с клавиатуры в специальное поле Web – страницы поискового каталога, делается щелчок по кнопке **Поиск**. После этого специальные средства поисковой системы автоматически просматривают все уровни каталогов и сообщают пользователю все найденные адреса.

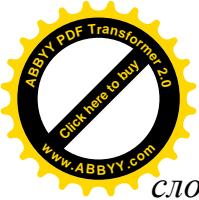
Наиболее популярными поисковыми каталогами являются один из крупнейших в мире англоязычный каталог **Yahoo!** (<http://www.yahoo.com>), а также российские каталоги **MavicaNet** (www.mavicanet.ru), **Каталог@Mail.ru** (list.mail.ru) и **Иван Сусанин** (www.walt.susanin.net).

Принцип действия *поисковых машин* отличается от принципа действия поисковых каталогов. *Поисковые машины* включают в себя *базу данных*, поля записей которой содержат адреса Web – страниц, ключевые слова этих страниц и другую информацию, позволяющую охарактеризовать содержимое страниц. В состав поисковой машины входит *специальная программа* – *поисковый робот* или Spider (spider – паук). Поисковый робот в автоматическом режиме просматривает Web – узлы Интернета и при обнаружении ссылки на какую – либо Web – страницу запоминает ее адрес и другие характеристики в специальной *таблице индексов* базы данных. Такие системы непрерывно отслеживают содержимое Web – страниц, тщательно «прочесывая» их с помощью своих «агентов» («пауков»), которые просматривают и индексируют вновь поступившие страницы.

Пользователь по специальным правилам *оформляет запрос*, в соответствии с которым осуществляется просмотр индексов. Найденные адреса Web – страниц пересылаются пользователю, направившему запрос.

Наиболее популярными в настоящее время являются поисковые машины: **Google** (<http://www.google.com>) (также ищет Web – сайты в русскоязычном сегменте Сети), **AltaVista** (<http://www.altavista.com>), **Northern Light** (<http://www.northernlight.com>). К лучшим российским поисковым серверам относятся **Яндекс** (<http://www.yandex.ru>), **Rambler** (<http://www.rambler.ru>) и **Апорт** (<http://www.aport.ru>).

Внешний вид домашних страниц поисковых серверов может быть различным, но в любом случае в окне имеется *поисковая форма*. Она содержит, как минимум, поле ввода запроса и расположенную рядом кнопку **Найти** или **Поиск**. Правила работы с поисковыми системами обычно излагаются в справочных подсистемах поисковых серверов. Для доступа к ней необходимо щелкнуть по гиперссылке **Помощь** (Справка, Help).



В поисковой системе Google в простейшем виде запрос представляет *с* слово или *последовательность разделенных пробелами ключевых слов*. Последовательность может быть осмысленным сочетанием слов, целым предложением или просто набором важных слов. Например, запрос сформулированный в виде предложения: Кто получил Нобелевскую премию по литературе в 2001 году? Этот запрос можно сформулировать и как последовательность ключевых слов: Нобелевская премия литература 2001 год. Поисковая машина Google не различает регистры букв, поэтому для нее слова Нобелевская и нобелевская будут восприниматься одинаково. Эта поисковая машина воспринимает слова в любой грамматической форме. Например, если в запросе указано слово автоматизация, то будут найдены ссылки на документы, содержащие слова автоматизированный, автоматика, автоматический и т. п. Она игнорирует союзы и предлоги (кто, где, как, по, на и т. п.), единичные цифры и буквы. Чтобы включить такие слова в запрос, необходимо использовать символ +. Для поиска информации о Петре 1 в запросе следует записать: Петр +1. *Обязательно следует перед знаком + ставить пробел, а после него – нет.* Знак + акцентирует поисковые механизмы на отбор документов, которые *обязательно содержат следующее за ним слово*. Например, в ответ на запрос частные объявления продажа велосипедов попадет много ссылок на Web – узлы с разнообразными частными объявлениями. В ответе на запрос частные объявления продажа +велосипедов останутся объявления только о продаже велосипедов.

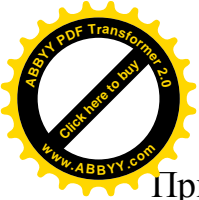
По умолчанию считается, что все ключевые слова запроса связаны логической операцией **И**. Это означает, что все указанные слова в документе должны присутствовать *одновременно* и не обязательно подряд.

Поисковая машина Google поддерживает и логическую операцию **ИЛИ**. В запросе она обозначается как **OR**. Например, для поиска страниц, которые должны содержать либо слово электропривод, либо слово автоматизация следует ставить такой запрос: электропривод OR автоматизация. В результате будут отобраны Web – страницы, содержащие либо слово электропривод, либо слово автоматизация, либо и то и другое вместе.

28.3. Загрузка файлов из сети Интернет

Одной из услуг Интернета является пересылка произвольных файлов между компьютерами Сети. Осуществляется это в соответствии с протоколом FTP. Доступ ко многим FTP – серверам можно получить с обозревателя Internet Explorer. Процесс пересылки файла с FTP – сервера на компьютер пользователя принято называть *загрузкой*. Существует несколько способов загрузки:

- *загрузка файла с FTP – сервера,*
- *загрузка файла с Web – сервера, связанного с FTP – сервером,*
- *пересылкой по почте.*



В первых двух случаях загрузка производится в *оперативном* (online) *режиме*. При почтовом варианте загрузки файл пересылается по указанному в окне адресу в *автономном* (offline) *режиме*, как присоединенный к почтовому сообщению.

28.4. Работа с почтовым сайтом

В последнее время появилось много сайтов, предоставляющих пользователю *бесплатные почтовые услуги*, например, узел свободных коммуникаций *Новая почта* (адрес <http://www.hotmail.ru>). Этот узел *бесплатно* предоставляет каждому желающему почтовый ящик, возможность создания и ведения персонального Web – узла и 32 Мбайт дисковой памяти для личной почты и личного Web – узла. Чтобы получить возможность пользоваться этим Web – узлом, необходимо зарегистрироваться. После проверки учетных данных на экран выводится завершающее окно регистрации. В нем отображаются персональные данные пользователя: его входное имя, почтовый адрес (e – mail), доменный адрес персонального сайта и т. д. В последнем окне регистрации предлагается ввести отображаемое в рамке число. После его ввода следует щелкнуть по кнопке **Зарегистрироваться**. В результате открывается окно заполнения формы персональных данных. Поля этой формы заполняются по желанию пользователя. Далее пользователь может отправлять и принимать почтовые сообщения, формировать свою персональную Web – страницу или завершить работу с сервером.

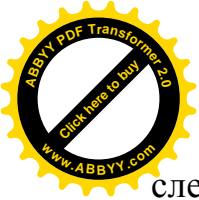
Во время каждого следующего посещения сервера на домашней его странице необходимо в поля **ВАШ ЛОГИН** и **ПАРОЛЬ** ввести собственные учетные данные и щелкнуть по кнопке **Войти в сервер**.

28.5. Программа Outlook Express

Программа Outlook Express является одним из модулей Web – обозревателя Internet Explorer. Она предназначена для обеспечения работы с электронной почтой (e – mail) и телеконференциями в сети Internet. Программа устанавливается автоматически при инсталляции обозревателя Internet Explorer и не требует дополнительного программного обеспечения. Все, что нужно для начала работы с Outlook Express, это адрес почтового ящика и выход в сеть.

28.5.1. Запуск и интерфейс программы Outlook Express

Outlook Express может быть запущена прямо с обозревателя Internet Explorer или из меню **Пуск** рабочего стола. Если выполнена стандартная установка, то программа находится в папке **Program Files\Outlook Express** под именем **msimn.exe**.



Интерфейс программы представлен тремя окнами. Одно окно расположено слева, а два других – справа друг под другом.

В левом окне обычно выводятся *папки*, содержащие *входящие, исходящие, отправленные, удаленные* сообщения и *черновики*. Текущая папка выбирается щелчком мыши. В правом верхнем окне выводится список сообщений, содержащихся в выбранной папке. Щелчок по одному из сообщений обеспечивает показ его текста в нижнем окне.

Интерфейс программы Outlook Express во многом схож с интерфейсом программы Internet Explorer, он достаточно удобен и прозрачен для быстрого восприятия.

28.5.2. Создание учетной записи

Учетной записью называют адрес электронной почты и его характеристики. В почтовой программе Outlook Express может быть создано любое количество учетных записей. Это значит, что с помощью этой программы можно работать одному или нескольким пользователям, имеющим один или несколько различных почтовых адресов. При работе с программой Outlook Express сообщения со всех ящиков собираются в одной папке **Входящие**.

Как уже ранее рассматривалось, почтовый (e – mail) адрес представляет собой двухзвенную конструкцию вида **m@n**, где первое звено m – имя пользователя, а второе звено n – адрес провайдера. Например:

igg@dntu.donetsk.ua

Здесь **igg** – имя пользователя,
dntu.donetsk.ua – доменное имя сети провайдера.

Для создания учетной записи в Outlook Express необходимо выполнить следующие действия:

– щелкнуть по пункту меню **Сервис** и в раскрывшемся окне выбрать команду **Учетные записи**,

– в появившемся диалоговом окне перейти на вкладку **Почта**, в которой должен располагаться список учетных записей электронной почты. Изначально этот список пуст.

– Щелкнуть по кнопке **Добавить**,

– в раскрывшемся контекстном меню щелкнуть по строке **Почта**.

В результате этих операций будет запущен **Мастер** создания нового соединения, с помощью которого нужно ввести:

- имя пользователя, полученное от провайдера,
- адрес электронной почты,
- адрес сервера для входящей почты POP3,



- *адрес сервера исходящей SMTP почты,*
- *имя учетной записи,*
- *пароль пользователя.*

Для того, чтобы получить почту с сервера провайдера, необходимо щелкнуть по кнопке **Доставить почту** или нажать комбинацию клавиш **Ctrl+M**.

28.5.3. Чтение сообщений

Для чтения сообщений необходимо:

- *раскрыть папку **Входящие**,*
- *в раскрывшемся списке сообщений следует щелкнуть по заголовку нужного сообщения.* Ниже, в окне просмотра появится его содержание.

Если необходимо увидеть письмо целиком, то нужно дважды щелкнуть по его заголовку. В этом случае письмо будет представлено в отдельном окне.

Если после прочтения письма нажать клавишу **Delete**, то письмо будет удалено из папки **Входящие** и помещено в папку **Удаленные**.

Чтобы скопировать (переместить) письмо, необходимо щелкнуть по нему правой кнопкой мыши и в контекстном меню выбрать нужную команду **Скопировать в папку (Переместить в папку)** и указать ту папку, куда следует копировать или переместить.

*Если удалить письмо из папки **Удаленные**, то оно теряется безвозвратно.*

28.5.4 Создание и отправка сообщения

Для создания сообщения необходимо выполнить следующую последовательность операций:

- *щелкнуть по кнопке **Создать сообщение**, что вызовет появление окна, которое будет содержать **бланк сообщения** и необходимый инструментарий для его создания;*

- *в поле **Кому** следует ввести **электронный адрес получателя**;*

- *для отправки копии письма в другой адрес **необходимо этот адрес ввести** в поле **Копия**;*

- *в поле **Тема** **вести краткое содержание письма**, из которого можно было бы понять на какую тему письмо. Обычно это одна – три фразы;*

- *в поле сообщения **необходимо ввести текст сообщения**. Для форматирования текста в окне помещена панель инструментов с необходимыми средствами для простейшего форматирования;*



– щелкнуть по кнопке **Отправить**. Если компьютер в этот момент и соединение с Сетью, то письмо будет немедленно отправлено адресату и помещено в компьютере отправителя в папку **«Отправленные»**.

В целях экономии времени нахождения в сети Internet сообщение можно готовить в Outlook Express при отключенном компьютере от Сети (в автономном режиме).

После подготовки таким образом сообщения следует щелкнуть по кнопке **Отправить**. В результате этого сообщение будет помещено в папку **«Исходящие»**. При первом же подключении к Internet все сообщения этой папки будут немедленно отправлены адресатам.

28.5.5. Разметка и форматирование сообщений

Для того, чтобы сообщение имело привлекательный вид можно использовать возможности программы Outlook Express для его разметки и форматирования. При этом следует воспользоваться командами

Формат – Формат HTML

Редактор сообщений имеет небольшой выбор средств, с помощью которых можно придавать сообщениям необходимый внешний вид. Можно устанавливать фон подложки определенного цвета или в виде рисунка. В качестве фона можно использовать звук (при открытии сообщения воспроизводится звуковой файл), можно также менять шрифты отдельных фрагментов текста, создавать гиперссылки, при помощи которых прямо из почтового сообщения можно запустить обозреватель для просмотра Web-страницы или почтовый клиент для создания и отправки почтового сообщения.

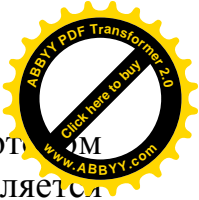
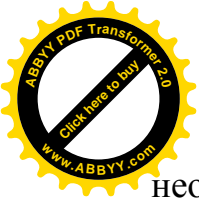
28.6. Работа с FTP – серверами с помощью встроенного FTP – клиента Total Commander

Программа Total Commander имеет встроенный **FTP**– клиент. При его помощи можно обратиться одновременно к десяти **FTP** – серверам по Internet. **FTP**– клиент позволяет не только закачивать и загружать файлы, но также поддерживать прямую передачу файлов из одного удаленного сервера на другой. В **FTP**– клиенте можно использовать стандартные операции с файлами так же, как и в Total Commander, то есть копировать (F5), переименовывать (SHIFT+F6), удалять(F8), создавать каталог (F7) и др.

Для установки соединения с **FTP** – сервером необходимо использовать команды

Сеть – Новое FTP-соединение и

Сеть – Соединиться с FTP-сервером.



При выборе команды **Новое FTP-соединение** появляется окно, в котором необходимо указать адрес *ftp-сервера*. После этого на активной панели появляется содержимое *ftp-сервера*, после чего можно работать с *ftp-сервером* как с диском компьютера.

Команда **Соединиться с FTP-сервером** открывает диалоговое окно (рис. 28.1), в котором хранится список доступных **FTP** – соединений.

Для соединения с *ftp-сервером* необходимо его выбрать в списке и щелкнуть по кнопке **Соединиться**. Изначально этот список пуст. Для создания в списке нового соединения необходимо выполнить команду **Добавить**.

В раскрывшемся диалоговом окне **Настройка FTP – соединения** необходимо установить параметры **FTP** – соединения:

– **Заголовок**. В этом поле можно указать произвольное имя соединения, например, Microsoft FTP – server.

– **Имя сервера**. Это поле позволяет ввести имя сервера или IP – адрес FTP-сервера, например, <ftp.firma.com:1037>.

– **Анонимное соединение**. Для конфигурации анонимного подключения Total Commander следует нажать эту кнопку. Это означает, что в имени пользователя для сервера будет использовано слово *anonymous*, а E – mail пользователя – как пароль. При этом Total Commander будет запрашивать E – mail пользователя при каждом нажатии на эту кнопку.

– **Учетная запись** (имя пользователя). Здесь можно ввести имя пользователя (*anonymous*, если не имеется учетной записи на сервере).

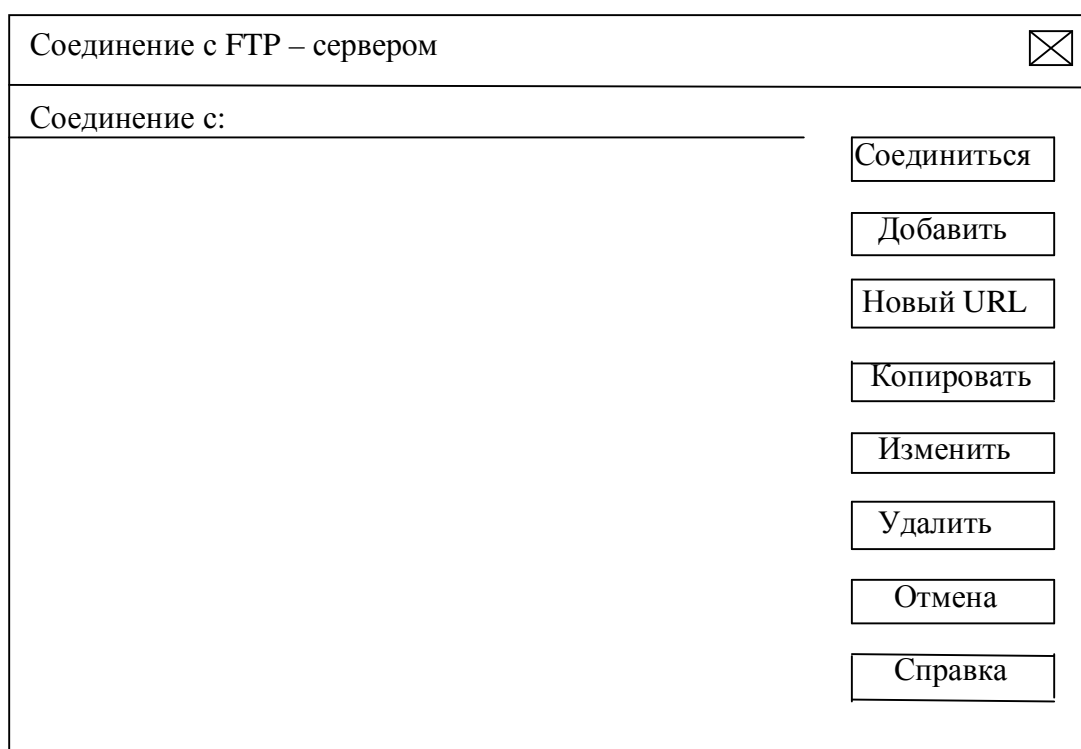


Рис. 28.1 Внешний вид диалогового окна

- - **Пароль.** Это поле для ввода пароля, который следует ввести, если есть уверенность в безопасности собственного компьютера! Сохранять пароли на компьютере открытых для других персон – не самый лучший способ. Если это поле оставить пустым, то Total Commander запросит при соединении пароль пользователя.

- **Удаленный каталог.** Если каталог определить здесь, то Total Commander сделает попытку открыть его на удаленном головном компьютере сразу после соединения. Следует иметь в виду, что при указании пути используется косая черта с наклоном вправо / .

- **Локальный каталог.** Если локальный каталог задан, то Total Commander переключится на другую файловую панель и откроет локальный каталог (если он задан). В пути локального каталога используется косая черта с наклоном влево \ .

- **Послать команды.** После соединения будут посланы указанные команды на **FTP-сервер**. Если посылаются несколько команд, то они отделяются друг от друга точкой с запятой.

- **Тип сервера.** Большинство общественных серверов в Internet поддерживают режим «Автоопределение» Total Commander.



– **Пассивный режим обмена (как Web – браузер).** Каждая передача (закачка или загрузка) нуждается в дополнительном соединении для потока данных. Обычно (в активном режиме) это соединение устанавливает сервер. Однако в пассивном режиме клиент устанавливает соединение самостоятельно. Такое возможно через некоторые firewalls, которые не позволяют соединения извне.

– **Посылать команды.** Это поле служит для поддержания соединения. Оно позволяет посылать определяемые пользователем команды каждые XX секунд для моделирования трафика. Это полезно для **FTP-серверов** с очень коротким временем ожидания, например, в несколько минут. Не более одного часа позволяет другим пользователям выполнить соединение.



ЛИТЕРАТУРА

1. Степанов А.Н. Информатика: учебник для ВУЗов. 4–е изд. – СПб.: Питер, 2005. – 684 с.: ил.
2. Еремин Е. А. Популярные лекции об устройстве компьютера. – СПб.: БХВ – Петербург, 2003. – 272 с.: ил.
3. Каган Б.М. Электронные вычислительные машины и системы: учеб. пособие для ВУЗов. – М.: Энергия, – 528 с.: ил.
4. Алексеев Е.Р., Анохина И.Ю., Чеснокова О.В. Работа на персональном компьютере IBM PC под управлением Windows. Донецк, 2003. – 189 с.: ил.
5. Самоучитель работы на компьютере. Начинаем с Windows. 2–е изд. / А. Левин. – СПб.: Питер, 2004 – 697 с.: ил.
6. Windows XP. Шаг за шагом / И.Г. Пашенко. – М.: Эксмо, 2008. – 368 с.
7. Под редакцией В.Б. Комягина. Компьютер для студентов. Самоучитель. Быстрый старт.: учебное пособ. – М.: издательство ТРИУМФ, 2003 – 400 с.: ил.
8. Сафронов И.К. Задачник–практикум по информатике. – Спб.: БХВ – Петербург, 2002. – 432 с.: ил.
9. Решение математических задач средствами Excel: практикум / В.Я. Гельман. – СПб.: Питер 2003. – 237 с.: ил.
10. Ларсен, Рональд У. Инженерные расчеты в Excel.: пер. с англ. – М.: издательский дом «Вильямс», 2004. – 544 с.: ил. парал. тит. англ.
11. Эффективная работа в Интернете / / В. Пасько. – СПб.: Питер; Киев: издательская группа ВНУ, 2003. – 544 с.: ил.
12. Интернет. Энциклопедия. 3–е изд. / Ю. Солоницын, В. Холмогоров. – СПб.: Питер, 2003. – 592 с.: ил.