

ФГБОУ ВПО  
КЕМЕРОВСКИЙ ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ  
ПИЩЕВОЙ ПРОМЫШЛЕННОСТИ

**Т.П. Крюкова, И.А.Печерских, В.В.Романова,  
А.Г.Семенов, Е.А.Столетова, Л.А.Яковлева**

# **ИНФОРМАТИКА**

## **Теория, вычисления, программирование**

Учебное пособие  
для практических и  
лабораторных работ

Кемерово 2015

УДК 004 (076)  
ББК 32.81я7

Рецензенты:

Ю.Н.Захаров, доктор физ.-мат. наук, профессор;  
Т.Ф.Лебедева, канд. техн. наук, доцент.

Рекомендовано редакционно-издательским советом Кемеровского технологического института пищевой промышленности

И 74 Информатика: Теория, вычисления, программирование: Учебное пособие для практических и лабораторных работ / Т.П. Крюкова, И.А.Печерских, В.В.Романова, А.Г.Семенов, Е.А.Столетова, Л.А.Яковлева; под ред. А.Г.Семенова. – Кемерово: КемТИПП, 2014. – 228 с.  
ISBN ...

В предлагаемом пособии описан комплекс практических и лабораторных работ по отдельным разделам курса информатики. Пособие охватывает следующие темы: математические основы информатики, вычислительный пакет MathCAD, основы алгоритмизации и программирования на языке Паскаль. Приведены краткие теоретические сведения, примеры, описания отдельных работ. Даны индивидуальные варианты заданий для выполнения работ.

Пособие предназначено для студентов всех специальностей инженерно-технических вузов.

Табл. – 28, библи. назв. – 8.

УДК 004 (076)  
ББК 32.81я7

ISBN ...

© КемТИПП, 2015

## ОГЛАВЛЕНИЕ

|                                                                              |    |
|------------------------------------------------------------------------------|----|
| ВВЕДЕНИЕ .....                                                               | 6  |
| Глава 1. МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ .....                             | 8  |
| 1.1. Позиционные системы счисления .....                                     | 8  |
| 1.1.1. Перевод чисел из одной системы счисления в другую..                   | 8  |
| 1.1.2. Двоичная и связанные с ней системы счисления. ....                    | 12 |
| 1.1.3. Задание для самостоятельной работы .....                              | 16 |
| 1.2. Измерение количества информации .....                                   | 19 |
| 1.2.1. Содержательный (энтропийный) подход к измерению информации .....      | 19 |
| 1.2.2. Алфавитный подход к измерению информации .....                        | 22 |
| 1.2.3. Единицы измерения информации .....                                    | 23 |
| 1.2.4. Задачи для самостоятельной работы .....                               | 23 |
| 1.3. Двоичное кодирование данных в ЭВМ.....                                  | 29 |
| 1.3.1. Кодирование целых чисел .....                                         | 29 |
| 1.3.2. Кодирование вещественных чисел .....                                  | 29 |
| 1.3.3. Кодирование символьных данных (текста) .....                          | 30 |
| 1.3.4. Кодирование графических данных .....                                  | 31 |
| 1.3.5. Задачи для самостоятельной работы .....                               | 33 |
| 1.4. Основы математической логики.....                                       | 39 |
| 1.4.1. Высказывания. ....                                                    | 39 |
| 1.4.2. Бинарные логические операции. ....                                    | 40 |
| 1.4.3. Законы математической логики. ....                                    | 44 |
| 1.4.4. Задания для самостоятельной работы .....                              | 47 |
| 1.5. Логические схемы.....                                                   | 56 |
| 1.5.1. Связь логических операций и двоичной арифметики..                     | 56 |
| 1.5.2. Логические элементы и логические схемы.....                           | 57 |
| 1.5.3. Задания для самостоятельной работы .....                              | 60 |
| Глава 2. ВЫЧИСЛИТЕЛЬНЫЙ ПАКЕТ MathCAD .....                                  | 65 |
| 2.1. Знакомство с вычислительным пакетом MathCAD .....                       | 65 |
| 2.1.1. Интерфейс и структура документов MathCAD .....                        | 65 |
| 2.1.2. Практическая работа .....                                             | 69 |
| 2.1.3. Задания для самостоятельной работы .....                              | 72 |
| 2.2. Табулирование функций и построение графиков .....                       | 76 |
| 2.2.1. Табулирование функций одной переменной и построение их графиков. .... | 76 |
| 2.2.2. Построение графика функции двух переменных. ....                      | 82 |

|                                                                               |     |
|-------------------------------------------------------------------------------|-----|
| 2.2.3. Задания для самостоятельной работы. ....                               | 85  |
| 2.3. Действия с векторами и матрицами .....                                   | 89  |
| 2.3.1. Палитра матричных операций.....                                        | 89  |
| 2.3.2. Создание матрицы .....                                                 | 90  |
| 2.3.3. Построение графика функции на основе матрицы .....                     | 92  |
| 2.3.4. Примеры действий с матрицами .....                                     | 93  |
| 2.3.5. Решение систем линейных уравнений с помощью<br>матричных операций..... | 95  |
| 2.3.6. Примеры действий с векторами.....                                      | 97  |
| 2.3.7. Задания для самостоятельной работы. ....                               | 98  |
| 2.4. Решение задач математического анализа .....                              | 101 |
| 2.4.1. Палитра операций математического анализа .....                         | 102 |
| 2.4.2. Задания для самостоятельной работы. ....                               | 105 |
| 2.5. Символьные вычисления в MathCad.....                                     | 109 |
| 2.5.1. Палитра символьных операций.....                                       | 109 |
| 2.5.2. Порядок выполнения символьных вычислений .....                         | 110 |
| 2.5.3. Алгебраические преобразования .....                                    | 111 |
| 2.5.4. Символьные операции математического анализа .....                      | 113 |
| 2.5.5. Задания для самостоятельной работы. ....                               | 113 |
| Глава 3. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ<br>ПАСКАЛЬ.....                     | 122 |
| 3.1. Общие сведения .....                                                     | 122 |
| 3.2. Типы данных в Паскале.....                                               | 122 |
| 3.3. Управляющие структуры языка Паскаль.....                                 | 124 |
| 3.4. Алфавит языка Паскаль.....                                               | 125 |
| 3.5. Алгоритм и его описание .....                                            | 127 |
| 3.6. Общая структура программы на языке Паскаль .....                         | 130 |
| 3.7. Раздел описаний.....                                                     | 131 |
| 3.8. Простые операторы.....                                                   | 132 |
| 3.9. Процедуры ввода-вывода.....                                              | 136 |
| 3.10. Структурные операторы.....                                              | 139 |
| 3.10.1. Условный оператор и оператор выбора .....                             | 139 |
| 3.10.2. Составной оператор (блок).....                                        | 145 |
| 3.10.3. Операторы цикла .....                                                 | 145 |
| 3.11. Подпрограммы. Процедуры и функции.....                                  | 150 |
| 3.12. Системы программирования на языке Паскаль .....                         | 156 |
| 3.13. Набор и редактирование программы .....                                  | 165 |
| 3.14. Этапы разработки компьютерной программы .....                           | 167 |

|                                                                                                      |     |
|------------------------------------------------------------------------------------------------------|-----|
| 3.15. Общие требования к выполнению лабораторных работ .....                                         | 168 |
| 3.16. Лабораторная работа 1: Запись сложных выражений.<br>Программирование линейного алгоритма ..... | 169 |
| 3.17. Лабораторная работа 2: Программирование алгоритмов с<br>разветвлением .....                    | 183 |
| 3.18. Лабораторная работа 3: Одномерные массивы и их<br>обработка.....                               | 190 |
| ЛИТЕРАТУРА .....                                                                                     | 225 |

## ВВЕДЕНИЕ

Предлагаемое учебное пособие представляет собой практикум по отдельным разделам курса информатики, которые связаны с использованием законов математики или ориентированы на применение компьютерной техники для проведения математических расчетов. Вместе с ранее изданным лабораторным практикумом "Информатика: Microsoft Office 2007" данное пособие обеспечивает проведение лабораторных и практических занятий по курсу "Информатика".

В первой главе рассматриваются математические основы предмета информатики – системы счисления, элементы математической логики. Показано, как эти математические законы используются при обработке информации.

Вторая глава посвящена знакомству с одним из наиболее распространенных прикладных программных продуктов, предназначенных для выполнения математических расчетов в числовой и символьной формах – пакету MathCAD. Рассмотрен интерфейс пакета и основные принципы его функционирования. В практических работах этой главы студенты могут получить навыки практического применения пакета MathCAD для выполнения расчетов. Поскольку дисциплина "Информатика" обычно изучается студентами I курса, используемые в работах математические знания не выходят за рамки школьного курса алгебры и начальных разделов курса высшей математики.

Третья глава посвящена знакомству с основами алгоритмизации и программирования. В качестве языка программирования для практической работы выбран язык Паскаль, традиционно используемый при изучении основ программирования. В начале главы приведено краткое описание языка Паскаль и современных систем программирования на этом языке.

Материалы каждого практического занятия включают необходимые теоретические сведения, примеры решения задач и наборы вариантов заданий для самостоятельной работы. Количество вариантов подобрано так, чтобы обеспечить индивидуальными заданиями стандартную учебную группу численностью 25 человек.

В тексте практикума используются обозначения:

- названия пунктов меню и подменю операционной системы Windows и приложений, диалоговых окон и их вкладок приводятся в кавычках: "Файл", "Формат" и т.п.;

- названия виртуальных кнопок и клавиш (в диалоговых окнах) выделяются квадратными скобками: [Отмена], [OK] и т.п.;

- названия клавиш клавиатуры компьютера выделяются угловыми скобками: <Alt>, <Shift>, <C> и т.п.

- отдельные символы и строки символов (тексты), набираемые с клавиатуры, выделяются особым шрифтом: =F^2.

Например, запись: **"Файл" – "Открыть" – [Новая папка]** – обозначает указание щелкнуть мышью виртуальную кнопку [Новая папка], которая находится в окне диалога команды "Открыть" из раздела меню "Файл".

### Глава 1. МАТЕМАТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ

#### 1.1. Позиционные системы счисления

Цель работы: знакомство с организацией позиционных систем счисления; освоение приемов вычислений в разных позиционных системах.

Система счисления - это способ записи чисел и соответствующие ему правила действий над числами. Системы счисления бывают позиционными и непозиционными.

В позиционной системе счисления значение, соответствующее какой-либо цифре в записи числа, зависит от расположения (позиции) этой цифры в записи. Используемая нами десятичная система счисления является позиционной, поэтому, например, две цифры 3 в записи числа 3123 имеют разные значения (первая обозначает 3 тысячи, а вторая – 3 единицы). Используемая иногда запись чисел "римскими цифрами" дает пример непозиционной системы – цифра X всегда имеет значение "десять", независимо от места, на котором она стоит. Далее речь будет идти только о позиционных системах счисления.

Всякая система счисления базируется на основании – некотором целом числе (например, основанием десятичной системы является число 10). Совокупность цифр, используемых для записи чисел, называется алфавитом системы счисления. Размер алфавита (количество используемых в системе цифр, включая 0) равен основанию системы. Максимальная цифра в  $r$ -ичной системе счисления имеет значение, равное  $(r - 1)$ .

1.1.1. Перевод чисел из одной системы счисления в другую.

Число в позиционной системе представляется в виде суммы произведений значений составляющих его цифр на соответствующие степени основания системы. В системе с основанием



## Глава 1. Математические основы информатики

р некоторое число  $X_p^1$ , имеющее  $n$  цифр в целой части и  $k$  цифр в дробной части, записывается в виде последовательности цифр:

$$X_p = a_n a_{n-1} a_{n-2} \dots a_1 a_0 , a_{-1} a_{-2} \dots a_{-k} \quad (1.1)$$

а его значение вычисляется так:

$$\begin{aligned} X_p = & a_n p^n + a_{n-1} p^{n-1} + \dots + a_1 p^1 + a_0 + \\ & + a_{-1} p^{-1} + a_{-2} p^{-2} + \dots + a_{-k} p^{-k} \end{aligned} \quad (1.2)$$

**Пример 1.1.** Записанное в пятеричной системе (с основанием 5) число  $20314_5$  имеет в десятичной системе значение

$$\begin{aligned} 20314_5 &= 2 \cdot 5^4 + 0 \cdot 5^3 + 3 \cdot 5^2 + 1 \cdot 5 + 4 = \\ &= 2 \cdot 625 + 3 \cdot 25 + 1 \cdot 5 + 4 = \\ &= 1250 + 75 + 5 + 4 = 1334_{10} \end{aligned}$$

**Пример 1.2.** Записанное в четверичной системе (с основанием 4) число  $132,12_4$  имеет в десятичной системе значение

$$\begin{aligned} 132,12_4 &= 1 \cdot 4^2 + 3 \cdot 4 + 2 + \frac{1}{4} + \frac{2}{4^2} = \\ &= 16 + 12 + 2 + 0,25 + 0,125 = 30,375_{10} \end{aligned}$$

Прежде, чем рассмотреть алгоритм перевода числа в ричную систему счисления, вспомним некоторые понятия целочисленной арифметики – целочисленное деление ("деление нацело") и остаток от такого деления.

Разделить число  $A$  на число  $B$  нацело (записывается:  $A : B$ ) – значит определить, сколько в числе  $A$  содержится целых порций, равных числу  $B$ . То, что останется, если от числа  $A$  отнять все эти целые порции, равные  $B$  – это остаток от деления нацело  $(A \bmod B)^2$ .

---

<sup>1</sup> При необходимости основание системы счисления, в которой записано число, указывается в нижнем индексе.

<sup>2</sup> Поскольку для операции определения остатка нет общепринятого обозначения, здесь и далее она обозначается служебным словом **mod**, принятым в языке программирования Паскаль.

## Глава 1. Математические основы информатики

**Пример 1.3.** Требуется разделить нацело число 460 на 50. Число 460 можно представить в виде  $9 \cdot 50 + 10$ . Т.о., в числе 460 содержится 9 порций по 50 и остаток, равный 10. Значит,  $460:50=9$ ,  $460 \bmod 50 = 10$ .

Произвести целочисленное деление и выделить остаток можно с помощью калькулятора. Допустим, надо разделить нацело 2345 на 17. Проведем с помощью калькулятора обычное деление:  $2345 / 17 = 137,9411\dots$ . Целая часть этого результата как раз и равна результату деления нацело:  $2345 : 17 = 137$ . Остаток вычисляется, как указано выше:

$$2345 \bmod 17 = 2345 - 137 \cdot 17 = 2345 - 2329 = 16.$$

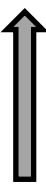
Для перевода числа А из десятичной системы в р-ичную используется следующий алгоритм:

1. Число А делится нацело на р и записывается остаток от деления.
2. Результат деления снова делится нацело на р и записывается остаток от деления.
3. Процесс заканчивается тогда, когда результатом деления оказывается ноль.
4. Остатки от деления записываются по порядку от последнего к первому. Это и будет р-ичная запись числа.

**Пример 1.4.** Перевести число  $3456_{10}$  в пятеричную систему счисления:

Таблица 1.1

Перевод числа в пятеричную систему

| Делимое | Результат деления на 5 | Остаток от деления на 5 | Направление записи числа                                                            |
|---------|------------------------|-------------------------|-------------------------------------------------------------------------------------|
| 3456    | 691                    | 1                       |  |
| 691     | 138                    | 1                       |                                                                                     |
| 138     | 27                     | 3                       |                                                                                     |
| 27      | 5                      | 2                       |                                                                                     |
| 5       | 1                      | 0                       |                                                                                     |
| 1       | 0                      | 1                       |                                                                                     |

## Глава 1. Математические основы информатики

Выписав остатки от деления в обратном порядке, получим пятеричную запись числа:

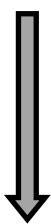
$$3456_{10} = 102311_5.$$

Для перевода правильной десятичной дроби в  $r$ -ичную систему счисления ее умножают на  $r$  и выписывают целую часть результата умножения. Дробную часть результата снова умножают на  $r$ , выписывают целую часть результата и т.д. Операцию повторяют до тех пор, пока дробная часть результата умножения не окажется равной 0, или пока в последовательности целых частей результата не выделится период дроби. Запись дробного числа в  $r$ -ичной системе выглядит так: ноль, десятичный разделитель(запятая или точка, в зависимости от принятых обозначений<sup>1</sup>), последовательность целых частей результатов умножения.

**Пример 1.5.** Перевести число  $0,789_{10}$  в пятеричную систему счисления:

Таблица 1.2

Перевод дроби в пятеричную систему

| Дробная часть числа | Результат умножения на 5 | Целая часть результата | Направление записи числа                                                           |
|---------------------|--------------------------|------------------------|------------------------------------------------------------------------------------|
| 0,789               | 3,945                    | 3                      |  |
| 0,945               | 4,725                    | 4                      |                                                                                    |
| 0,725               | 3,625                    | 3                      |                                                                                    |
| 0,625               | 3,125                    | 3                      |                                                                                    |
| 0,125               | 0,625                    | 0                      |                                                                                    |
| 0,625               | 3,125                    | 3                      |                                                                                    |
| 0,125               | 0,625                    | 0                      |                                                                                    |

Дальнейшее умножение можно не проводить, т.к. последние две строки повторяют предыдущие две и очевидно, что дальше эти пары строк будут повторяться до бесконечности.

---

<sup>1</sup>В русскоязычной литературе десятичным разделителем обычно является запятая, в зарубежной – точка. В компьютерных программах разделителем является точка, за исключением программ пакета MicrosoftOffice, где для этого используется запятая.

## Глава 1. Математические основы информатики

Иначе говоря, в дроби выделился период. Запись дробного числа будет выглядеть так:

$$0,789_{10} = 0,343303030\dots_5 = 0,343(30)_5$$

Если десятичное число содержит как целую, так и дробную части, они переводятся в новую систему счисления по отдельности и записываются через десятичный разделитель. Например, число  $3456,789_{10}$  в пятеричной системе счисления запишется так:  $102311,343(30)_5$ .

### 1.1.2. Двоичная и связанные с ней системы счисления.

Особую роль в информатике играет двоичная система счисления, использующая только две цифры – 0 и 1. Это связано с двумя обстоятельствами:

- Выбор одной из двух возможностей образует естественную единицу измерения информации (об этом подробно будет сказано далее);
- Различные электронные устройства имеют два легко различимых состояния (например, конденсатор: "заряжен-не заряжен"; ток через проводник: "течет-не течет" – и т.п.), поэтому система с двумя цифрами легко реализуется с технической точки зрения.

Перевод числе из десятичной системы счисления в двоичную и обратно проводится по правилам, описанным выше. Однако двоичная запись более или менее крупного числа представляет собой длинную цепочку нулей и единиц, чтение которой затруднительно. Поэтому в информатике для сокращенной записи двоичных чисел используются связанные с двоичной системой восьмеричная<sup>1</sup> и шестнадцатеричная системы счисления.

Восьмеричная система использует цифры 0, 1, 2, 3, 4, 5, 6, 7. Если перевести числовые значения этих цифр в двоичную систему, получим следующую таблицу:

---

<sup>1</sup> Восьмеричная система счисления использовалась, например, при описании операционной системы UNIX.

Таблица 1.3

Двоичные эквиваленты восьмеричных цифр

| Восьмеричная цифра | Двоичный эквивалент | Восьмеричная цифра | Двоичный эквивалент |
|--------------------|---------------------|--------------------|---------------------|
| 0                  | 000                 | 4                  | 100                 |
| 1                  | 001                 | 5                  | 101                 |
| 2                  | 010                 | 6                  | 110                 |
| 3                  | 011                 | 7                  | 111                 |

Как видно из таблицы, эквивалент любой восьмеричной цифры состоит из трех двоичных (с учетом незначащих нулей в записи эквивалентов цифр 0 – 3). Для перевода двоичного числа в восьмеричное применяется следующий алгоритм:

1. Двоичная запись разбивается на группы по три двоичных цифры – "триады" – справа налево (от конца к началу). Если при этом в первой триаде не хватит цифр, спереди дописываются незначащие нули.

2. Каждая триада заменяется восьмеричной цифрой в соответствии с таблицей 1.3.

Для перевода восьмеричного числа в двоичное каждую восьмеричную цифру просто заменяют эквивалентной двоичной триадой согласно таблице 1.3.

Шестнадцатеричная система счисления использует 16 цифр с числовыми значениями 0, ..., 15. Первые десять цифр совпадают с обычными (арабскими), для записи остальных 6 используются начальные буквы английского алфавита от А до F<sup>1</sup>.

Каждая шестнадцатеричная цифра также имеет двоичный эквивалент. Они приведены в таблице 1.4.

Перевод двоичного числа в шестнадцатеричное и обратный перевод из шестнадцатеричной системы в двоичную произ-

---

<sup>1</sup> Аналогичным образом (на основе английского алфавита) можно образовать цифры и в других системах счисления с основанием  $p > 10$

## Глава 1. Математические основы информатики

водятся по правилам, аналогичным переводу в восьмеричную систему и обратно. Разница состоит в том, что двоичную запись при этом надо разбивать не на триады, а на группы по четыре двоичных цифры – "тетрады". Затем каждая тетрада заменяется эквивалентной шестнадцатеричной цифрой согласно таблице 1.4.

Таблица 1.4

Десятичные и двоичные эквиваленты шестнадцатеричных цифр

| Шестнадцатеричная цифра | Десятичное значение | Двоичный эквивалент | Шестнадцатеричная цифра | Десятичное значение | Двоичный эквивалент |
|-------------------------|---------------------|---------------------|-------------------------|---------------------|---------------------|
| 0                       | 0                   | 0000                | 8                       | 8                   | 1000                |
| 1                       | 1                   | 0001                | 9                       | 9                   | 1001                |
| 2                       | 2                   | 0010                | A                       | 10                  | 1010                |
| 3                       | 3                   | 0011                | B                       | 11                  | 1011                |
| 4                       | 4                   | 0100                | C                       | 12                  | 1100                |
| 5                       | 5                   | 0101                | D                       | 13                  | 1101                |
| 6                       | 6                   | 0110                | E                       | 14                  | 1110                |
| 7                       | 7                   | 0111                | F                       | 15                  | 1111                |

**Пример 1.6.** Перевести число  $3853_{10}$  в двоичную, восьмеричную и шестнадцатеричную системы счисления:

Сначала переведем число в двоичную систему (см. табл. 1.5):

$$3853_{10} = 111100001101_2.$$

Для перевода в восьмеричную систему разобьем двоичное число на триады и заменим каждую триаду на восьмеричный эквивалент в соответствии с таблицей 1.3:

|                       |     |     |     |     |
|-----------------------|-----|-----|-----|-----|
| Двоичная запись триад | 111 | 100 | 001 | 101 |
| Восьмеричная запись   | 7   | 4   | 1   | 5   |

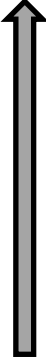
## Глава 1. Математические основы информатики

Таким образом

$$3853_{10} = 7415_8.$$

Таблица 1.5

Перевод числа в двоичную систему

| Делимое | Результат деления на 2 | Остаток от деления на 2 | Направление записи числа                                                          |
|---------|------------------------|-------------------------|-----------------------------------------------------------------------------------|
| 3853    | 1926                   | 1                       |  |
| 1926    | 963                    | 0                       |                                                                                   |
| 963     | 481                    | 1                       |                                                                                   |
| 481     | 240                    | 1                       |                                                                                   |
| 240     | 120                    | 0                       |                                                                                   |
| 120     | 60                     | 0                       |                                                                                   |
| 60      | 30                     | 0                       |                                                                                   |
| 30      | 15                     | 0                       |                                                                                   |
| 15      | 7                      | 1                       |                                                                                   |
| 7       | 3                      | 1                       |                                                                                   |
| 3       | 1                      | 1                       |                                                                                   |
| 1       | 0                      | 1                       |                                                                                   |

Для перевода в шестнадцатеричную систему разобьем двоичное число на тетрады и заменим каждую на шестнадцатеричный эквивалент в соответствии с таблицей 1.4:

|                          |      |      |      |
|--------------------------|------|------|------|
| Двоичная запись тетрад   | 1111 | 0000 | 1101 |
| Шестнадцатеричная запись | F    | 0    | D    |

Итак:

$$3853_{10} = F0D_{16}.$$

**Пример 1.7.** Вычислить сумму  $31_{10} + 101_2 + 75_8 + AD_{16}$  и записать результат в шестнадцатеричной системе счисления.

**Решение.** Сначала переведем все числа в десятичную систему (при переводе шестнадцатеричного числа используем таблицу 1.4):

$$101_2 = 1 \cdot 2^2 + 1 \cdot 2^0 = 4 + 1 = 5_{10}$$

## Глава 1. Математические основы информатики

$$175_8 = 1 \cdot 8^2 + 7 \cdot 8^1 + 5 \cdot 8^0 = 7 \cdot 8 + 5 = 64 + 56 + 5 = 125_{10}$$

$$AD_{16} = 10 \cdot 16^1 + 13 \cdot 16^0 = 10 \cdot 16 + 13 = 160 + 13 = 173_{10}.$$

Результат сложения:  $31 + 5 + 125 + 173 = 334_{10}$ . Переводим его в шестнадцатеричную систему:

| Делимое | Результат деления на 16 | Остаток от деления на 16 | Направление записи числа                                                          |
|---------|-------------------------|--------------------------|-----------------------------------------------------------------------------------|
| 334     | 20                      | $14_{10} = E_{16}$       |  |
| 20      | 1                       | 4                        |                                                                                   |
| 1       | 0                       | 1                        |                                                                                   |

Итак:

$$31_{10} + 101_2 + 75_8 + AD_{16} = 14E_{16}.$$

### 1.1.3. Задание для самостоятельной работы

Решить 5 задач из приведенного ниже списка согласно индивидуальному варианту задания (табл. 1.6).

#### Задачи для решения:

1. Записать десятичное число 2009 в пятеричной системе счисления
2. Сколько и каких требуется цифр, чтобы можно было записать какое-то число в двенадцатеричной системе счисления?
3. Укажите, какие числа записаны с ошибкой:  $123_7$ ,  $3005_4$ ,  $12AAC09_{20}$ ,  $13476_7$ .
4. Известно, что алфавитом некоторой нетрадиционной системы счисления являются символы: 0, 1, ☺, \*, ∇, ⇐, ⊗, ⇒. Каково основание этой системы? Запишите числа 8, 13, 66 в этой системе.
5. Запишите в шестеричной системе число, следующее за числом 5.
6. Запишите три числа, которые следуют за числом  $111_4$  в четверичной системе?
7. Какое число в восьмеричной системе предшествует числу  $10_8$ ?



## *Глава 1. Математические основы информатики*

8. Какое минимальное основание может иметь система, если в ней записаны числа: 432, 120, 111, 2331?
9. В некоторой системе счисления десятичное число 26 записывается в виде 101. Чему равно основание системы?
10. Какое число является наибольшим из трех:  $66_9$ ,  $144_5$  или  $123_6$ ?
11. Записать число  $83_{10}$  в двоичной системе счисления.
12. Записать число  $163_{10}$  в двоичной системе счисления.
13. Записать число  $5678_{10}$  в двоичной системе счисления.
14. Записать число  $3287_{10}$  в двоичной системе счисления.
15. Записать число  $6666_{10}$  в двоичной системе счисления.
16. Перевести число  $11011011_2$  в десятичную систему счисления.
17. Перевести число  $110001110010_2$  в десятичную систему счисления.
18. Как будет выглядеть в двоичной системе десятичное число 0,125?
19. Как будет выглядеть в двоичной системе десятичное число 0,015625?
20. Перевести число  $111011101_2$  в десятичную систему счисления.
21. Перевести число  $668_{10}$  в восьмеричную систему счисления.
22. Перевести число  $1053_8$  в двоичную систему счисления.
23. Перевести число  $1053_8$  в десятичную систему счисления.
24. Каково количество значащих цифр в двоичной записи восьмеричного числа  $32_8$ ?
25. Перевести число  $385_{10}$  в восьмеричную систему счисления.
26. Перевести число  $765_8$  в десятичную систему счисления.
27. Перевести число  $765_8$  в двоичную систему счисления.
28. Какая цифра будет последней в восьмеричной записи числа  $519_{10}$ ?
29. Какая цифра будет предпоследней в восьмеричной записи двоичного числа  $1100010101001010101$ ?
30. Сколько значащих цифр будет содержать восьмеричная запись двоичного числа  $1101010110001010$ ?
31. Перевести двоичное число  $1100101001101010111$  в шестнадцатеричную систему счисления.
32. Переведите число  $4AD3_{16}$  в двоичную систему счисления

## Глава 1. Математические основы информатики

33. Переведите число  $F4CD3_{16}$  в двоичную систему счисления
34. Чему равна (в десятичной записи) разность шестнадцатеричных чисел  $7777_{16} - 887_{16}$ ?
35. Переведите число  $1234567_8$  в шестнадцатеричную систему счисления.
36. Переведите число  $A3CD8F_{16}$  в восьмеричную систему счисления.
37. Какова в шестнадцатеричной записи последняя цифра числа  $4536_8$ ?
38. Какова в восьмеричной записи первая значащая цифра числа  $DE38C_{16}$ ?
39. Верно ли неравенство  $FD2_{16} > 1011001010_2$ ?
40. Переведите число  $12AF_{16}$  в десятичную систему счисления.
41. Вычислите сумму чисел  $271_8$  и  $11110100_2$  и представьте результат в шестнадцатеричной системе счисления.
42. Найдите значение суммы:  $112_{16} + 112_8$ . Результат представьте в восьмеричной системе счисления.
43. Найдите значение суммы:  $101011_2 + 1C_{16}$ . Результат представьте в восьмеричной системе счисления.
44. Дано:  $a=D7_{16}$ ,  $b=331_8$ . Приведите пример числа  $c$ , записанного в двоичной системе и отвечающего условию  $a < c < b$ .
45. Чему равна сумма  $(43_8 + 56_{16})$  в двоичной записи?
46. Сумма  $111_2 + 111_8$  в шестнадцатеричной системе счисления запишется: ...
47. Записанный в десятичной системе счисления результат операции  $55_8 + 55_{16}$  равен...
48. Как запишется в шестнадцатеричной системе счисления результат операции  $277_8 - 27_{10}$ ?
49. Какова будет первая цифра шестнадцатеричного числа  $88_{16}$ , представленного в восьмеричной записи?
50. Значение суммы чисел  $1110101_2 + 1011_3$  в восьмеричной системе счисления равно ...

Таблица 1.6

Варианты выбора задач.

| № варианта | Номера задач из списка | № варианта | Номера задач из списка |
|------------|------------------------|------------|------------------------|
| 1          | 1, 11, 21, 31, 41      | 14         | 8, 16, 28, 40, 42      |
| 2          | 2, 12, 22, 32, 42      | 15         | 10, 11, 22, 34, 46     |
| 3          | 3, 13, 23, 33, 43      | 16         | 1, 20, 27, 39, 41      |
| 4          | 4, 14, 24, 34, 44      | 17         | 3, 15, 21, 33, 45      |
| 5          | 5, 15, 25, 35, 45      | 18         | 5, 19, 25, 37, 49      |
| 6          | 6, 16, 26, 36, 46      | 19         | 7, 13, 29, 31, 43      |
| 7          | 7, 17, 27, 37, 47      | 20         | 9, 17, 23, 35, 47      |
| 8          | 8, 18, 28, 38, 48      | 21         | 4, 15, 28, 39, 42      |
| 9          | 9, 19, 29, 39, 49      | 22         | 8, 19, 22, 33, 46      |
| 10         | 10, 20, 30, 40, 50     | 23         | 2, 13, 26, 37, 50      |
| 11         | 2, 14, 26, 38, 50      | 24         | 6, 17, 30, 31, 44      |
| 12         | 4, 18, 30, 32, 44      | 25         | 10, 11, 24, 35, 48     |
| 13         | 6, 12, 24, 36, 48,     |            |                        |

## 1.2. Измерение количества информации

Цель работы: освоение методов оценки объема информации, содержащегося в сообщении.

Для измерения количества информации используют разные подходы.

### 1.2.1. Содержательный (энтропийный) подход к измерению информации

Всякое сообщение приводит к тому, что неопределенность наших знаний об окружающем мире (называемая также "информационной энтропией") уменьшается. Сообщение можно

## Глава 1. Математические основы информатики

рассматривать, как ответ на какой-то (заданный или воображаемый) вопрос. Простейший вопрос предусматривает только два варианта ответа: "Да" или "Нет". Количество информации, содержащееся в таком ответе, принимается за единицу и называется бит. Один бит уменьшает неопределенность знаний о какой-то ситуации в два раза.

Например, один бит информации содержит сообщение о том, что подброшенная монета упала "решкой", т.к. всего имелось два возможных результата падения и сообщение позволило выбрать один из них. Если число возможных вариантов  $N > 2$ , количество информации в сообщении определяется по одной из двух формул:

1) если все варианты равновероятны, количество информации определяется формулой Хартли

$$I = \log_2 N \quad (1.3)$$

Эту формулу удобно использовать также в обратном виде: количество информации, равное  $I$  бит, позволяет сделать выбор одного из

$$N = 2^I \quad (1.4)$$

вариантов.

Полезно в этой связи запомнить таблицу различных степеней числа 2 (табл. 1.7).

Таблица 1.7

Степени числа 2

| I | $N=2^I$ | I   | $N=2^I$ |
|---|---------|-----|---------|
| 0 | 1       | 7   | 128     |
| 1 | 2       | 8   | 256     |
| 2 | 4       | 9   | 512     |
| 3 | 8       | 10  | 1024    |
| 4 | 16      | ... | ...     |
| 5 | 32      | 16  | 65 536  |
| 6 | 64      | ... | ...     |

## Глава 1. Математические основы информатики

2) если вероятности отдельных исходов не равны, количество информации определяется формулой Шеннона

$$I = - \sum_k (p_k \cdot \log_2 p_k), \quad (1.5)$$

где  $p_k$  – вероятность  $k$ -го исхода.

**Замечание.** Теоретически количество информации, вычисленное по формулам Хартли или Шеннона, может оказаться дробным. Однако в вычислительной технике информация записывается с помощью устройств памяти, каждое из которых может хранить только целый бит. Поэтому технически возможно записать только целое число битов. Если вычисленное количество информации оказывается дробным, при определении объема требуемой компьютерной памяти его округляют до ближайшего целого в большую сторону.

**Пример 1.8.** Каково минимальное количество битов, которое потребуется для записи в компьютере сообщения о результате падения игральной кости (кубика).

Варианты решения:

а) поскольку любая из 6 граней кубика может выпасть с одинаковой вероятностью, используем формулу Хартли (1.3) при  $N = 6$ :  $I = \log_2 6 = 2,585$ .

Т.к. получилось дробное число, округляем до ближайшего целого вверх:  $I = 3$ .

б) используем формулу (1.4). Из нее следует, что 2 бита информации позволяют описать только  $2^2 = 4$  возможных варианта падения кубика, тогда как необходимо описать 6 вариантов. В то же время 3 бита позволяют описать уже  $2^3 = 8$  возможных вариантов, так что этого количества целых битов хватает даже с избытком. Но т.к. технически количество информации должно выражаться целым числом битов, принимаем, что указанное сообщение содержит 3 бита.

**Пример 1.9.** В корзине лежат шары. Все они – разного цвета. Сообщение о том, что из корзины достали синий шар, содержит ровно 5 битов информации. Сколько всего шаров в корзине?

### **Решение.**

Т.к. выбор любого из шаров наугад равновероятен, воспользуемся формулой (1.4). Подставим в нее значение  $I = 5$  (битов). Получается, что количество шаров в корзине:

$$N = 2^I = 2^5 = 32.$$

### 1.2.2. Алфавитный подход к измерению информации

Алфавитный подход к измерению информации не связывает количество информации с содержательным аспектом сообщения.

Множество используемых в языке сообщения символов называется алфавитом. Общее количество символов алфавита  $N$  называется мощностью алфавита. Информационный объем одного символа определяется формулой Хартли<sup>1</sup>:  $I = \log_2 N$ . Если сообщение содержит  $K$  символов, то информационный объем всего сообщения равен  $K \cdot I$ .

**Пример 1.10.** Книга состоит из 64 страниц. На каждой странице 256 символов. Какой объем информации содержится в книге, если используемый алфавит состоит из 32 символов?

**Решение.** По условию задачи мощность алфавита равна 32 символам. Найдем информационную емкость одного символа:

$I = \log_2 N = \log_2 32 = 5$  (бит). Определим информационную емкость одной страницы: поскольку на странице 256 символов, то имеем  $256 \cdot 5 = 1280$  (бит). Определим информационную емкость всей книги:  $64 \cdot 1280 = 81920$  (бит).

**Пример 1.11.** Рисунок на дисплее сотового телефона состоит из 480 строк по 240 пикселей в строке. Каждый пиксель может иметь один из 65536 цветов. Определить объем информации, содержащейся в рисунке.

---

<sup>1</sup> На самом деле в тексте на любом языке разные буквы встречаются с разной вероятностью, поэтому может показаться, что здесь надо использовать формулу Шеннона. Однако, поскольку алфавитный подход не рассматривает содержание сообщения, вероятность появления отдельных букв не принимается во внимание.

## Глава 1. Математические основы информатики

**Решение.** В данном случае алфавит – это множество возможных пикселей разного цвета. Т.о., мощность алфавита равна 65 536. Значит, информационная емкость одного пикселя  $I = 16$  бит (это видно из табл. 1.6). Информационный объем рисунка:  $480 \cdot 240 \cdot 16 = 1\,843\,200$  бит.

### 1.2.3. Единицы измерения информации

Как уже сказано, основной единицей измерения информации является бит. В информатике широко используется еще одна единица измерения информации – байт. 1 байт равен 8 битам. Из формулы (1.4) следует, что информация объемом в 1 байт позволяет описать 256 вариантов чего-либо (вариантов выбора, исходов события и т.п.)

Для измерения значительных количеств информации используются кратные единицы:

1 килобайт (Кбайт) =  $2^{10}$  байт = 1024 байта;

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

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

1 терабайт (Тбайт) =  $2^{40}$  байт = 1024 Гбайт;

1 петабайт (Пбайт) =  $2^{50}$  байт = 1024 Тбайт.

Так, количество информации в примере 1.9 равно, как видно из решения,  $5 \cdot 256 \cdot 64 = 5 \cdot 2^8 \cdot 2^6 = 5 \cdot 2^{8+6} = 5 \cdot 2^{14}$  бит. Это число можно записать так:  $5 \cdot 2 \cdot 2^3 \cdot 2^{10}$  бит. Но

$$2^3 \text{ бит} = 8 \text{ бит} = 1 \text{ байт},$$

$$2^{10} \text{ байт} = 1 \text{ Кбайт}.$$

Тогда информационный объем книги в этом примере равен  $(5 \cdot 2) \text{ Кбайт} = 10 \text{ Кбайт}$ .

### 1.2.4. Задачи для самостоятельной работы

1. Скорость передачи данных через ADSL-соединение равна 256000 бит/сек. Передача файла через это соединение по времени заняла 2 мин. Определите размер файла в килобайтах.

## Глава 1. Математические основы информатики

2. В книге 500 страниц. На каждой странице книги 20 строк по 64 символа в кодировке ASCII<sup>1</sup>. Объем книги равен ... килобайт.
3. 28 триггеров (устройств памяти, сохраняющих 1 бит) могут хранить ... байт информации.
4. Объем текстовой информации в сообщении на 40 страницах (на странице 40 строк по 80 символов в каждой) в кодировке ASCII равен...
5. Для кодирования 20 различных состояний достаточно использовать... двоичных разрядов.
6. Сколько существует различных последовательностей из символов «а» и «б» длиной ровно в 10 символов?
7. Сколько существует различных последовательностей из символов «а» и «б» длиной ровно в 10 символов?
8. При перекодировке сообщения из кода UNICODE в код ASCII объем сообщения изменился на 1/512 Мб. Сообщение содержит ... символов.
9. Сообщение, записанное буквами 64-символьного алфавита, содержит 40 символов. Какой объем информации оно содержит?
10. Какое количество информации содержит сообщение, уменьшающее неопределенность знаний в 8 раз?
11. Сколько различных последовательностей длиной в 7 символов можно составить из цифр 0 или 1?
12. Сколько битов содержит 1 Гбайт?
13. Сколько Мбайт информации содержит сообщение объемом  $2^{23}$  битов?
14. Объем сообщения, содержащего 4096 символов, составил 1/512 часть Мбайта. Какова мощность алфавита, с помощью которого записана это сообщение?
15. Объем сообщения 7,5 Кбайт. Известно, что сообщение содержит 7680 символов. Какова мощность алфавита данного сообщения?
16. Объем информационного сообщения равен 40960 бит. Чему равен объем этого сообщения в Кбайтах?

---

<sup>1</sup> Таблица ASCII-кодов содержит 256 символов, упоминаемая далее таблица UNICODE – 65 536 символов.



## *Глава 1. Математические основы информатики*

17. Дан некий текст размером 600 разнообразных символов. Известно, что все символы взяты из таблицы, содержащей 16 строк и 32 столбца. Определить информационный объем текста в битах.
18. Какое минимальное количество битов потребуется для кодирования целых положительных чисел, меньших 50?
19. Растровый графический файл содержит черно-белое изображение размером 800х600 точек (черных или белых). Определите размер этого файла на диске в байтах.
20. Расположите следующие значения объемов информации:  $2^{10}$  байт; 20000 бит; 2001 байт; 2 Кбайт – в порядке убывания.
21. В зрительном зале две прямоугольные области зрительских кресел: одна – 6 на 12, а другая – 8 на 4. Минимальное количество бит, которое потребуется для кодирования каждого места в автоматизированной системе, равно ...
22. В зрительном зале две прямоугольные области зрительских кресел: одна размером 10х12 (10 рядов по 12 кресел), а другая 17х8. Какое минимальное количество бит потребуется для кодирования каждого места в автоматизированной системе?
23. Имеется колода из 36 игровых карт. Загадывается одна из карт. Загадавший на все вопросы отвечает только «Да» или «Нет». Чтобы гарантированно угадать задуманную карту, нужно задать как минимум ... вопросов.
24. Максимальное количество страниц книги (32 строки по 64 символа в кодировке ASCII), которая может быть записана в файле объемом 640 Кбайт, составляет ...
25. Если средняя скорость чтения составляет 160 слов в минуту, а одно слово содержит в среднем 6 символов, то за четыре часа непрерывного чтения можно прочесть ... Кбайт текста в кодировке ASCII.
26. Сообщение объемом  $2^{33}$  бит содержит ... Гбайт информации.
27. Световое табло состоит из лампочек, каждая из которых может находиться в двух состояниях («включено» или «выключено»). Какое наименьшее количество лампочек должно

## *Глава 1. Математические основы информатики*

- находиться на табло, чтобы с его помощью можно было передать 60 различных сигналов?
28. Сколько информации несет сообщение о том, что было угадано некое целое число в диапазоне от 684 до 811 включительно?
  29. В организации работает 129 сотрудников. Каков должен быть минимальный размер (в битах) поля базы данных, содержащего номер сотрудника в полном списке работающих?
  30. Богатырь стоит перед камнем на распутье. На камне надпись: "Налево пойдешь – жену найдешь, направо пойдешь – богатство найдешь, прямо пойдешь – славу найдешь". Сколько бит информации будет содержать сообщение о выбранном пути?
  31. Метеорологическая станция ведет наблюдение за влажностью воздуха. Результатом одного измерения является целое число от 0 до 100 процентов, которое сохраняется некоторым цифровым запоминающим устройством. Станция сделала 80 измерений. Определите информационный объем результатов наблюдений.
  32. Два текста содержат одинаковое количество символов. Первый текст составлен в алфавите мощностью 16 символов. Второй текст в алфавите мощностью 256 символов. Во сколько раз количество информации во втором тексте больше, чем в первом?
  33. В велокроссе участвуют 119 спортсменов. Специальное цифровое устройство регистрирует прохождение каждым из участников промежуточного финиша, записывая его номер. Каков информационный объем сообщения, записанного устройством после того, как промежуточный финиш прошли 70 велосипедистов?
  34. Сколько битов информации несет сообщение о том, что тетраэдр, у которого все грани окрашены в разные цвета, после подбрасывания упал на синюю грань?

## *Глава 1. Математические основы информатики*

35. Сообщение передано с помощью пятибитного кода Бодо<sup>1</sup>. Каков его информационный объём в байтах, если известно, что передано 4000 символов?
36. В алфавите формального (искусственного) языка имеется всего два знака – "крестик" и "нолик". Каждое слово этого языка состоит ровно из пяти знаков. Какое максимальное число слов возможно в таком языке?
37. Алфавит языка племени "Мумбо-Юмбо" содержит всего 8 букв. Какое количество информации несет одна буква этого алфавита?
38. Световое табло состоит из лампочек. Каждая лампочка может находиться в одном из трех состояний («включено», «выключено» или «мигает»). Какое наименьшее количество лампочек должно находиться на табло, чтобы с его помощью можно было передавать 18 различных сигналов?
39. Для передачи секретного сообщения используется код, состоящий из десятичных цифр. При этом все цифры кодируются одним и тем же минимально возможным количеством битов. Определите информационный объём сообщения длиной в 150 символов.
40. В корзине лежат 16 шаров. Все шары разного цвета. Сколько информации несет сообщение о том, что из корзины выкатился красный шар?
41. Какое количество информации будет получено вторым игроком в игре «Крестики-нолики» на поле размером 4×4 после первого хода первого игрока?
42. Шахматная доска состоит из 64 полей: 8 столбцов и 8 строк. Какое минимальное количество битов потребуется для кодирования координат одного шахматного поля?
43. Обычный дорожный светофор без дополнительных секций подает шесть видов сигналов (непрерывные красный, желтый и зеленый, мигающий желтый, мигающий зеленый, мигающие красный и желтый одновременно). Электронное устройство управления светофором последовательно воспроизводит записанные сигналы. В устройстве записан по-

---

<sup>1</sup> Код Бодо был разработан в начале XX века для нужд телеграфной связи.

## Глава 1. Математические основы информатики

- рядок поочередной подачи 100 сигналов светофора. Информационный объем записи составляет ... байт.
44. В лексиконе Элочки-Людоедки, как известно, было 30 слов. Она произносит фразу, состоящую из 50 слов. В этом случае количество информации, которое сообщает Элочка, составляет \_\_\_\_\_ бит. Считать, что выбор любого из 30 слов равновероятен.
  45. Словное индийское письмо "Деванагари" использует комбинации 89 различных символов. Какое минимальное количество битов требуется для записи одного символа?
  46. Определить, какое количество информации из перечисленных является максимальным: 2000 бит, 250 байт,  $\frac{1}{4}$  Кбайта.
  47. Определить, какое количество информации из перечисленных является максимальным: 4000 байт, 4 Кбайта,  $2^{16}$  бит, 32 Кбита.
  48. В романе американского писателя Нила Стивенсона "Ртуть" героиня создает зашифрованное сообщение, вышивая подушечку "крестиком" (вышитый "крестик" означает 1, оставленное пустое место такого же размера – 0). Пусть фрагмент подушечки для вышивки образован переплетением 640 продольных и 480 поперечных нитей, для вышивки крестика используется фрагмент размером  $4 \times 4$  нити. Какое максимальное количество букв может содержать сообщение, если оно составлено на английском языке, в алфавите которого – 26 букв?
  49. В доме живут 7 коротышек: Незнайка, Знайка, Пончик, Цветик, Винтик, Шпунтик и Пилюлькин. Каждое утро они кидают жребий, кому готовить завтрак. Сколько бит содержит сообщение "Завтрак готовит Незнайка"?
  50. Трехдюймовая дискета (гибкий магнитный диск, в настоящее время вышел из употребления) могла содержать до 1,44 Мбайт информации. Современный оптический DVD-диск имеет емкость 4,70 Гбайт. Содержимое какого количества дискет может вместить один DVD-диск?

### 1.3. Двоичное кодирование данных в ЭВМ

Обработка информации в средствах вычислительной техники основана на представлении данных последовательностью символов 0 и 1. Такое представление называется двоичным кодированием. Количество различных значений, которое может быть закодировано, зависит от количества разрядов (битов)  $I$ , и определяется формулой (1.4).

Рассмотрим по порядку принципы кодирования разных видов информации.

#### 1.3.1 Кодирование целых чисел

Из формулы (1.4) следует, что ячейка памяти размером 8 битов позволяет закодировать 256 разных целых чисел – например, в диапазоне  $0...255$ , или  $-128...+127$ . Используя 16-разрядную кодировку, можно закодировать 65536 разных значений ( $0...65535$  или  $-32768...+32767$ ), а 24 бита позволяют закодировать уже более 16,5 миллионов разных значений. Если необходимо использовать как положительные, так и отрицательные числа, первый бит ячейки памяти используется для записи знака числа<sup>1</sup>.

#### 1.3.2. Кодирование вещественных чисел

Вещественное число предварительно преобразуется в нормализованную форму:

$$56,7834 = 0,567834 \cdot 10^{+2}$$

$$-0,00812 = -0,812 \cdot 10^{-2}$$

$$453\,567\,123 = 0,453567123 \cdot 10^{+9}$$

В общем виде запись вещественного числа  $N$  в нормализованной форме имеет структуру:

$$N = \pm M \cdot 10^{\pm P} \quad (1.6)$$

---

<sup>1</sup> В рамках данного пособия не рассматриваются технические способы кодирования отрицательных чисел – с использованием т.н. обратного и дополнительного кодов.

## Глава 1. Математические основы информатики

Здесь  $M$  – мантисса числа,  $P$  – порядок числа (по абсолютной величине). Порядок всегда имеет целое значение. Мантисса обязана удовлетворять условию:  $0,1 \leq m < 1$ , т.е., первая цифра после десятичного разделителя должна быть значащей. В этом случае последовательность цифр мантиссы можно рассматривать, как некоторое положительное целое число.

Значит, мантиссу и порядок можно закодировать двоичным кодом, как обычные неотрицательные целые числа. Ячейка памяти для записи вещественного числа состоит из 4 зон:

1. Зона хранения знака числа (1 бит)
2. Зона хранения последовательности цифр мантиссы
3. Зона хранения знака порядка (1 бит)
4. Зона хранения абсолютной величины порядка

Размер ячейки может различаться и составляет в разных случаях от 6 до 10 байт. Это обеспечивает запоминание от 11 до 20 цифр мантиссы (более длинные мантиссы округляются) и значений абсолютной величины порядка в диапазоне от 39 до примерно 4900.

### 1.3.3. Кодирование символьных данных (текста)

Если каждому символу алфавита сопоставить целое число (порядковый номер в некоторой таблице), то с помощью двоичного кода можно кодировать текстовые данные.

По международным соглашениям наиболее употребляемыми в настоящее время являются:

- восьмиразрядная система кодирования ASCII ("American Standard Code for Information Interchange" – "Стандартный код информационного обмена США");
- шестнадцатиразрядная универсальная система кодирования UNICODE.

Кодовая таблица ASCII включает  $2^8 = 256$  различных символов. В ней можно выделить две отдельных части – базовую и расширенную. Базовая таблица определяет значения кодов от 0 до 127 – заглавные и строчные буквы английского алфавита, цифры, знаки препинания и ряд служебных символов. Расширенная часть соответствует символам с номерами от 128 до 255

и используется в разных странах для кодирования прежде всего национальных алфавитов<sup>1</sup>.

Таблица UNICODE содержит  $2^{16} = 65536$  различных символов. Она обеспечивает кодирование большинства распространенных национальных алфавитов.

### 1.3.4. Кодирование графических данных

В процессе кодирования изображения производится его пространственная дискретизация – разбиение изображения на отдельные мельчайшие точки – пиксели. Каждой точке присваивается определенный код цвета. Совокупность используемых цветов образует цветовую палитру.

Графическое изображение, состоящее из пикселей, называется растром (bitmap). Метод создания изображения на основе растра называется растровым методом кодирования.

Качество двоичного кодирования изображения определяется:

- разрешающей способностью, например:  $800 \times 600$ ,  $1024 \times 768$ , или  $1280 \times 1024$ . Первое число в записи разрешающей способности показывает количество пикселей в строке, второе – количество строк в изображении;
- глубиной цвета – количеством битов, используемым для кодирования цвета точки.

Возможное количество цветов  $N$  и глубина цвета  $I$  связаны соотношением (1.4):

$$N = 2^I$$

При чёрно-белом изображении без градаций серого цвета каждая точка экрана может иметь одно из двух состояний – «чёрная» или «белая», то есть глубина цвета такого изображения равна 1 (для описания состояния точки используется 1 бит).

---

<sup>1</sup> В СССР в свое время было разработано несколько различных вариантов расширения таблицы ASCII для записи символов русского алфавита ("кириллицы"); кроме того, существуют варианты кириллических расширений зарубежного происхождения – "DOS" и "Windows". Все они отличаются друг от друга. Это иногда приводит к сбоям при раскодировании текстов, использующих кириллицу.

## Глава 1. Математические основы информатики

Цветное изображение формируется в соответствии с двоичным кодом цвета каждого пикселя, хранящимся в памяти. Для создания различных цветов производится смешивание трех основных цветов. Степень присутствия отдельного основного цвета в составе общего цвета пикселя определяется некоторым уровнем – от 0 до  $K$  (ноль означает полное отсутствие данного основного цвета,  $K$  – его максимальное присутствие).

Система цветового кодирования, использующая в качестве основных цвета: красный ("**Red**"), зелёный ("**Green**"), синий ("**Blue**") – называется системой RGB. Она применяется при создании цветных изображений на изначально черном экране (мониторе компьютера, дисплее телефона или фотоаппарата и т.п.) путем подсветки отдельных точек. В современных устройствах для кодирования уровня каждого основного цвета используется 1 байт ( $K = 255$ ), т.е., всего на описание одного пикселя требуется 3 байта. Это обеспечивает глубину цвета 24 бита, т.е., создание  $2^{24} = 16\,777\,216$  различных цветов. Код цвета записывается с использованием шестнадцатеричной системы счисления по схеме RRGGBB (RR– шестнадцатеричный код уровня красного цвета, GG– зеленого, BB – синего). Так, код 000000 означает, что точка на экране не подсвечена ни одним из основных цветов, т.е., осталась черной. Код FFFFFFFF означает максимальную подсветку точки всеми цветами – при этом на экране она будет выглядеть белой. Код FF0000 соответствует ярко-красному цвету, и т.д.

Каждой паре основных цветов системы RGB можно поставить в соответствие дополнительный цвет, дополняющий эту пару до белого. Список дополнительных составляют цвета: голубой ("**Cyan**" – к синему и зеленому), малиновый ("**Magenta**" – к синему и красному), жёлтый ("**Yellow**" – к зеленому и красному). Эти цвета используются как основные в системе кодирования CMYK, применяемой при печати на белой бумаге цветных изображений, которые рассматриваются в отраженном свете. По ряду причин к тройке основных цветов в системе CMYK добавлен черный цвет ("**black**" – буква **K** в названии системы). Для кодирования цвета одной точки в этой системе требуется 4 байта, т.е., 32 бита.



**Пример 1.12.** Скорость передачи данных через модемное соединение – 28 Кбит/с. Передача текстового файла заняла 10 с. Определите, сколько символов содержал переданный текст, если известно, что он был представлен в кодировке UNICODE.

**Решение.** 1 Кбит = 1024 бит, поэтому за 1 сбудет передано  $28 \cdot 1024 = 28\,672$  бита, за 10 с –  $28\,672 \cdot 10 = 286\,720$  битов. Так как переданный текст представлен в кодировке UNICODE, на кодирование 1 символа отводится 16 бит, поэтому количество символов равно  $286\,720 : 16 = 17\,920$ .

**Пример 1.13.** Для хранения растрового изображения размером  $128 \times 128$  пикселей отвели 4 Кбайта памяти. Каково максимально возможное число цветов в палитре изображения?

**Решение.** Количество записанной информации равно 4 Кбайта =  $4 \cdot 2^{10}$  байт =  $2^2 \cdot 2^{10}$  байт =  $2^{12}$  байт =  $8 \cdot 2^{12}$  бит =  $2^3 \cdot 2^{12}$  бит =  $2^{15}$  бит. Разрешающая способность изображения:  $128 \cdot 128 = 2^7 \cdot 2^7 = 2^{14}$  (пикселей). Тогда глубина цвета одного пикселя (число битов для описания цвета)  $I = 2^{15} / 2^{14} = 2$ .

Количество возможных цветов определяется формулой (1.4):  $N = 2^I = 2^2 = 4$ .

### 1.3.5. Задачи для самостоятельной работы

1. С помощью одного байта при двоичном кодировании можно представить целое неотрицательное число от 0 до...
2. Количество целых байтов<sup>1</sup> для кодирования числа 25710 равно ...
3. Максимальное целое число без знака, которое может быть записано с помощью кода, состоящего из шести двоичных цифр, равно...
4. Вещественное число имеет значение 123,456. Его нормализованная запись в форме с плавающей десятичной точкой имеет вид ....

---

<sup>1</sup> Если количество информации в технических приложениях, как говорилось выше, должно составлять целое число **битов**, то объем памяти при записи порции информации должен составлять целое число **байтов**. Это связано с тем, что собственным адресом в памяти ЭВМ может обладать только байт.

## *Глава 1. Математические основы информатики*

5. Вещественное число имеет значение 123,456. Его мантисса имеет вид ....
6. Вещественное число имеет значение 123,456. Его порядок при записи в нормализованном виде равен ... .
7. Какое количество информации (в байтах) соответствует одному символу из таблицы UNICODE?
8. Рассматривается файл в простом текстовом формате. Используется 8-разрядная кодовая таблица CP-1251 (Windows Cyrillic). В тексте 200 страниц, на странице 32 строки, а в строке в среднем 48 символов. Сколько килобайт памяти будет занимать файл?
9. В кодировке КОИ-8 ("Код информационного обмена восьмиразрядный") код русской буквы "и" в десятичной записи равен 201. Цифровой код каждой следующей буквы отличается от кода предыдущей на 1. Как будет выглядеть код слова "лимон" в шестнадцатеричной записи?
10. Текст **I LOVE** в соответствии с кодовой таблицей ASCII кодируется как 73 32 76 79 86 69 (запись шестнадцатеричная). Тогда текст **I LIVE** будет иметь кодировку ....
11. Буква "i" в таблице ASCII имеет десятичный код 105. Что зашифровано в сообщении: 108 105 110 107?
12. Оцените информационный объем высказывания Жан-Жака Руссо в кодировке ASCII:

**Тысячи путей ведут к заблуждению, к истине – только один.**

13. Оцените информационный объем высказывания Алексея Толстого в кодировке ASCII:

**Не ошибается тот, кто ничего не делает, хотя это и есть его основная ошибка.**

14. Оцените информационный объем высказывания Рене Декарта в кодировке ASCII:

**Я мыслю, следовательно, существую.**

15. Оцените информационный объем высказывания Дени Дидро в кодировке ASCII:

## *Глава 1. Математические основы информатики*

**Что такое истина? Соответствие наших суждений созданиям природы.**

16. Оцените информационный объем высказывания Блеза Паскаля в кодировке ASCII:

**Красноречие – это живопись мысли.**

17. Оцените информационный объем следующего предложения в кодировке UNICODE:

**Один пуд – около 16,4 килограмм.**

18. Оцените информационный объем высказывания Оноре де Бальзака в кодировке UNICODE:

**Ключом ко всякой науке является вопросительный знак.**

19. Определите информационный объем слова из двадцати четырех символов в кодировке UNICODE.

20. Оцените информационный объем фрагмента стихотворения А.С.Пушкина в кодировке UNICODE:

**Привычка свыше нам дана:**

**Замена счастию она.**

21. Определить информационный объем следующего сообщения в кодировке UNICODE?

**$2+12=14$ , а  $5+15=20$ .**

22. Минимальная длина возможного двоичного кода для букв английского алфавита (26 букв) равна ...

23. Сообщение из 50 символов было записано в 8-битной кодировке Windows-1251. После вставки в текстовый редактор сообщение было перекодировано в 16-битный код UNICODE. Количество памяти, занимаемое сообщением, увеличилось на ... байт.

24. Аналоговый звуковой сигнал был дискретизирован сначала с использованием 65536 уровней интенсивности сигнала (качество звучания аудио-CD), а затем с использованием 256 уровней интенсивности сигнала (качество звучания радио-

## *Глава 1. Математические основы информатики*

- трансляции). Во сколько раз будут различаться информационные объемы кодов.
25. Для хранения неупакованного растрового изображения размером  $32 \times 32$  пикселя потребовалось 512 байт памяти. Максимально возможное число цветов в палитре изображения равно ...
  26. Сколько бит памяти занимает информация об 1 пикселе в черно-белом изображении (без полутонов)?
  27. Растровое изображение размером  $64 \times 64$  пикселя занимает 4 килобайта памяти. Максимальное количество цветов, используемых в изображении, равно ...
  28. На экране с разрешающей способностью  $640 \times 200$  высвечиваются только двухцветные изображения. Какой минимальный объем видеопамати необходим для хранения изображения?
  29. Какой объем видеопамати необходим для хранения двух страниц рисунка при условии, что разрешающая способность дисплея равна  $640 \times 350$  пикселей, а количество используемых цветов – 16?
  30. Какой объем видеопамати необходим для хранения четырех страниц изображения при условии, что разрешающая способность дисплея равна  $800 \times 600$  пикселей, а глубина цвета – 24 бита?
  31. Объем памяти, занятой изображением – 1 Мбайт. Разрешающая способность дисплея –  $800 \times 600$  пикселей. Какое максимальное количество цветов можно использовать при условии, что рисунок делится на 2 страницы?
  32. Располагаемый объем памяти, равен 2Мбайтам, разрешающая способность дисплея- $640 \times 480$ , глубина цвета – 24 бита. Какое максимальное количество страниц изображения можно сохранить?
  33. Разрешающая способность дисплея –  $640 \times 200$ . Для размещения одного символа в текстовом режиме используется матрица  $8 \times 8$  пикселей, которая называется знакоместом. Какое максимальное количество: 1) текстовых строк, 2) знакомест в строке – может быть размещено на экране? Какой

## Глава 1. Математические основы информатики

- объем памяти займет код одной экранной страницы текста, если использовать кодировку ASCII?
34. Глубина цвета – 32, изображение делится на 2 страницы, разрешающая способность –  $800 \times 600$ . Вычислите объем памяти для хранения изображения.
  35. Объем задействованной памяти позволяет хранить четырехцветное изображение размером  $300 \times 200$ . Изображение какого размера можно сохранить в том же объеме памяти, если использовать 16-цветную палитру? Предполагается, что пропорции изображения остаются неизменными.
  36. Для хранения изображения экрана монитора, работающего при разрешении  $1366 \times 768$  в 256-цветном режиме, необходимо затратить ... Кбайт памяти.
  37. Для кодирования цвета используются шестнадцатеричные значения интенсивности цветовых компонент в 24-битной системе RGB. Какой цвет будет задавать код **00FF00**?
  38. Код Бодо позволял передавать символы текста по телеграфной связи путем передачи комбинаций из пяти двоичных символов. Сколько различных текстовых символов можно закодировать, используя код Бодо?
  39. Автоматическое устройство осуществило перекодировку информационного сообщения на русском языке, первоначально записанного в 16-битном коде UNICODE, в 8-битную кодировку КОИ-8. При этом информационное сообщение уменьшилось на 480 бит. Сколько символов содержит сообщение?
  40. Шестнадцатеричный код заглавной русской буквы "Б" ("мягкий знак") в таблице UNICODE равен **042C**. Каков код заглавной буквы "Я", если буквы в таблице UNICODE размещены по порядку их следования в алфавите?
  41. Какой цвет в системе CMYK соответствует коду **000000FF**?
  42. Для создания изображений на дисплее мобильного телефона используется система цветового кодирования RGB. На кодирование величины уровня каждого базового цвета отводится 4 бита. Сколько возможных цветов могут иметь точки данного изображения?

## *Глава 1. Математические основы информатики*

43. Дисплей цифрового фотоаппарата имеет разрешение  $960 \times 240$ . Сколько цветов может на нем создаваться, если изображение на дисплее занимает 450 Кбайт памяти?
44. Дисплей сотового телефона имеет разрешение  $480 \times 960$ . Количество цветов – 16 777 216. Какой объем памяти (в Кбайтах) необходим для хранения изображения?
45. Объем 3,5-дюймовой дискеты для хранения информации составлял 1,43 Мбайт. Сколько книжных страниц в кодировке ASCII можно было записать на одну дискету, если в среднем на одной странице помещается 45 строк по 50 символов? Ответ округлить до целого числа страниц.
46. Сколько изображений разрешением  $800 \times 600$  в стандартной кодировке RGB (256 уровней каждого базового цвета) можно сохранить на диске CD емкостью 700 Мбайт? Ответ округлить до целого числа.
47. Для хранения вещественного числа используется ячейка памяти размером 4 байта. Размер поля для хранения абсолютной величины мантиссы числа составляет 20 битов. Какова максимальная абсолютная величина порядка числа, которое можно сохранить в такой ячейке?
48. Для хранения вещественного числа используется ячейка памяти размером 4 байта. Размер поля для хранения абсолютной величины мантиссы числа составляет 20 битов. Каково максимальное число, которое можно сохранить в такой ячейке?
49. Для хранения целого числа использована ячейка памяти размером 2 байта. При этом диапазон возможных значений чисел, которые можно сохранить в такой ячейке, разделен пополам между отрицательными и неотрицательными числами. Каково минимальное число, которое можно сохранить в такой ячейке.
50. Сколько битов должно составлять поле для сохранения абсолютной величины порядка в ячейке памяти для хранения вещественного числа, чтобы в этой ячейке можно было записать величину массы электрона ( $9,11 \cdot 10^{-31}$  кг) ?

### 1.4. Основы математической логики

**Цель работы:** знакомство с основными понятиями математической логики, используемыми в схемотехнике вычислительных машин, программировании и др.

#### 1.4.1. Высказывания.

Основной элемент в логике – высказывание. Это любое утверждение, про которое можно сказать, что оно либо верно (имеет значение "Истина"), либо неверно (значение "Ложь"), и не может иметь третьего значения – правило исключенного третьего.

**Чтобы проверить, является ли некое предложение  $A$  высказыванием, можно подставить его внутрь вопроса: "Верно ли, что  $A$ ?" Если получится вопрос, имеющий смысл, то  $A$  – высказывание.**

Если рассмотреть некоторое множество объектов (обозначим его  $M$ ), то высказывание может выполняться для одних элементов  $M$  и не выполняться для других. Это можно проиллюстрировать диаграммой Эйлера-Венна (коротко говорят просто "диаграмма Венна")

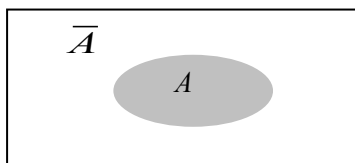


Рис. 1.1. Диаграмма Эйлера-Венна

Прямоугольник показывает множество  $M$  всех рассматриваемых элементов (фундаментальное). Заливкой выделено множество элементов, для которых некое высказывание  $A$  имеет значение "Истина". Тогда для остальных элементов высказывание имеет значение "Ложь". Любой элемент фундаментального множества либо принадлежит подмножеству истинности высказывания  $A$  (точка лежит внутри заливной области), либо не при-

## Глава 1. Математические основы информатики

надлежит ему, что соответствует правилу исключенного третьего

Для простоты будем обозначать значение "Истина" цифрой 1, а значение "Ложь" – 0. Таким образом, логические значения оказываются связаны с двоичными числами. То, что высказывание истинно, запишется в этом случае равенством

$$A = 1.$$

Для элементов, для которых  $A = 0$ , имеет место утверждение, противоположное  $A$ . Оно обозначается  $\bar{A}$ , или  $\neg A$  и называется отрицанием высказывания  $A$ . Например, высказывания  $(x \leq a)$  и  $(x > a)$  являются отрицаниями друг друга.

### 1.4.2. Бинарные логические операции.

Рассмотрим два высказывания:  $A$  и  $B$ . Их различные сочетания могут давать новые сложные высказывания. Способы этих сочетаний соответствуют различным логическим операциям:

I. Логическое сложение (дизъюнкция, "операция ИЛИ").

Дизъюнкция соответствует выражению "**A или B**" и обозначается символом  $\vee$ :

$$A \vee B$$

Такое высказывание истинно, если истинно хотя бы одно из образующих его простых высказываний. Это можно проиллюстрировать таблицей истинности (табл. 1.8).

Таблица 1.8

Таблица истинности дизъюнкции

| A | B | $A \vee B$ |
|---|---|------------|
| 0 | 0 | 0          |
| 0 | 1 | 1          |
| 1 | 0 | 1          |
| 1 | 1 | 1          |

На диаграмме Венна (рис. 1.2) дизъюнкция верна для заштрихованной области (здесь и далее эллипсы показывают



## Глава 1. Математические основы информатики

множества элементов, для которых верны соответственно высказывания А или В).

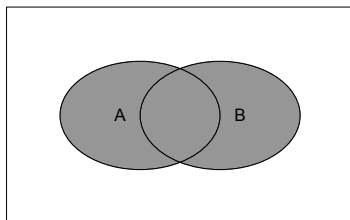


Рис. 1.2. Диаграмма Венна для дизъюнкции

### II. Логическое умножение (конъюнкция, "операция И")

Конъюнкция соответствует выражению "А и В" и обозначается символами  $\wedge$  или  $\&$  ("амперсанд"):

$$A \wedge B, \text{ или } A \& B$$

Такое высказывание истинно, если истинны оба образующих его простых высказывания. Таблица истинности и диаграмма Венна приведены в табл. 1.9. и на рис. 1.3.

Таблица 1.9

Таблица истинности конъюнкции

| A | B | A & B |
|---|---|-------|
| 0 | 0 | 0     |
| 0 | 1 | 0     |
| 1 | 0 | 0     |
| 1 | 1 | 1     |

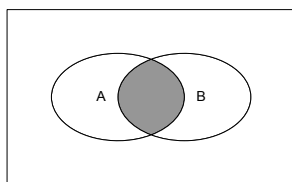


Рис. 1.3. Диаграмма Венна для конъюнкции

Для того, чтобы лучше запомнить термины "дизъюнкция" и "конъюнкция", можно применить mnemonic правило:

**Количество букв "И" в словах "дизъюнкция" и "конъюнкция" такое же, как в соответствующих им союзах "Или" и "И".**

III. Строгая дизъюнкция:  $A \text{ xor } B$ , или  $A \oplus B$  ("исключающее ИЛИ"; в литературе имеет разные символические обозначения и названия: дизъюнктивная сумма, исключающая альтернатива и др.).

Такое высказывание истинно, если истинно только одно из образующих его простых высказываний. Таблица истинности строгой дизъюнкции и диаграмма Венна приведены соответственно в табл. 1.10 и на рис. 1.4.

Таблица 1.10

Таблица истинности строгой дизъюнкции

| A | B | $A \oplus B$ |
|---|---|--------------|
| 0 | 0 | 0            |
| 0 | 1 | 1            |
| 1 | 0 | 1            |
| 1 | 1 | 0            |

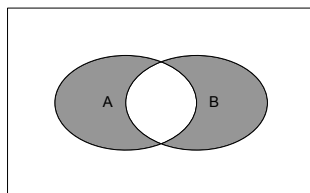


Рис. 1.4. Диаграмма Венна для строгой дизъюнкции

IV. Импликация, или следование.

Оно соответствует выражениям:

"Если A, то B"

"Так как A, то B"

"A, поэтому B"

и обозначается:

$$A \Rightarrow B, \quad A \rightarrow B.$$

Таблица. 1.11

Таблица истинности импликации:

| A | B | $A \Rightarrow B$ |
|---|---|-------------------|
| 0 | 0 | 1                 |
| 0 | 1 | 1                 |
| 1 | 0 | 0                 |
| 1 | 1 | 1                 |

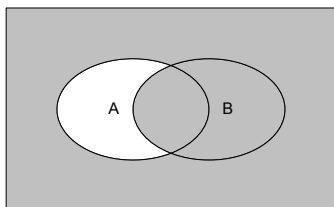


Рис. 1.5. Диаграмма Венна для импликации

V. Эквивалентность:  $A \sim B$ .

Эквивалентность соответствует высказыванию:

"A тогда и только тогда, когда B"

Таблица 1.12

Таблица истинности эквивалентности

| A | B | $A \sim B$ |
|---|---|------------|
| 0 | 0 | 1          |
| 0 | 1 | 0          |
| 1 | 0 | 0          |
| 1 | 1 | 1          |

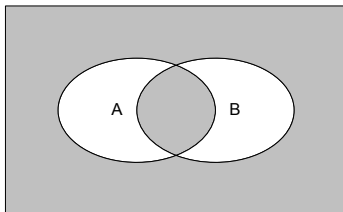


Рис. 1.6. Диаграмма Венна для эквивалентности

## Глава 1. Математические основы информатики

Анализируя диаграммы Венна можно показать, что операции строгой дизъюнкции, импликации и эквивалентности можно заменить комбинациями операций отрицания, конъюнкции и дизъюнкции.

### 1.4.3. Законы математической логики.

а) Законы ассоциативности (группировки):

$$A \& (B \& C) = (A \& B) \& C \quad (1.7)$$

$$A \vee (B \vee C) = (A \vee B) \vee C \quad (1.8)$$

б) Законы коммутативности (перестановочности):

$$A \vee B = B \vee A \quad (1.9)$$

$$A \& B = B \& A \quad (1.10)$$

в) Законы идемпотентности<sup>1</sup>:

$$A \vee A = A \quad (1.11)$$

$$A \& A = A \quad (1.12)$$

г) Свойства логических констант:

$$A \vee 0 = A \quad (1.13)$$

$$A \& 1 = A \quad (1.14)$$

д) Правило исключенного третьего:

$$A \vee \bar{A} \equiv 1 \quad (1.15)$$

$$A \& \bar{A} \equiv 0 \quad (1.16)$$

е) Законы дистрибутивности (распределительные законы):

$$A \vee (B \& C) = (A \vee B) \& (A \vee C) \quad (1.17)$$

$$A \& (B \vee C) = (A \& B) \vee (A \& C) \quad (1.18)$$

ж) Законы поглощения

---

<sup>1</sup> Слово "идемпотентность" имеет латинские корни ("idem" – "тот же самый", "potens" – "сильный") и может быть переведено, как "равносильность".

$$A \& (A \vee B) = A \quad (1.19)$$

$$A \vee (A \& B) = A \quad (1.20)$$

з) Правила де Моргана:

$$\overline{A \& B} = \bar{A} \vee \bar{B} \quad (1.21)$$

$$\overline{A \vee B} = \bar{A} \& \bar{B} \quad (1.22)$$

Примером применения правил де Моргана является запись утверждения, что некоторое число  $x$  не принадлежит отрезку  $[a, b]$ . Это высказывание можно записать так:  $\overline{a \leq x \leq b}$ , т.е.,  $\overline{(x \geq a) \& (x \leq b)}$ . А можно записать так:  $(x < a) \vee (x > b)$ .

Можно отметить, что законы ассоциативности и коммутативности, а также свойства логических констант полностью аналогичны соответствующим законам обычной (числовой) алгебры, если высказывания заменить числами, а дизъюнкцию и конъюнкцию – алгебраическими сложением и умножением. То же можно сказать о втором из законов дистрибутивности. Однако первый закон дистрибутивности (1.17) не имеет аналога в числовой алгебре.

Порядок выполнения основных операций соответствует обычной алгебре: сначала выполняется умножение, потом сложение, в последнюю очередь – отрицание. Для изменения порядка выполнения действий используются скобки.

**Пример 1.14.** Используя диаграммы Венна и законы математической логики, построить высказывание, в котором операция строгой дизъюнкции заменена комбинацией операций отрицания, конъюнкции и дизъюнкции.

**Решение.** Анализируя рис. 1.2 – 1.4, можно заметить, что залитая фигура на рис. 1.4 (строгая дизъюнкция) получается, если из заливки фигуры рис. 1.2 (дизъюнкция) "вырезать" фигуру рис. 1.3 (конъюнкция). Т.о., точки, попавшие в заливку на рис. 1.4, попадают в заливку и на рис. 1.2 (удовлетворяют условию дизъюнкции), но одновременно не должны попасть в заливку на рис. 1.3 (не удовлетворяют условию конъюнкции). Это означает, что

$$A \oplus B = (A \vee B) \& \overline{(A \& B)}$$

Построенное логическое выражение является сложным для анализа. Упростим его. Сначала избавимся от отрицания во втором логическом множителе с помощью правила де Моргана:

$$A \oplus B = (A \vee B) \& (\bar{A} \vee \bar{B})$$

Теперь на основе закона дистрибутивности внесем первый логический множитель внутрь вторых скобок:

$$A \oplus B = [(A \vee B) \& \bar{A}] \vee [(A \vee B) \& \bar{B}]$$

Внутри каждой квадратной скобки также внесем логический множитель внутрь круглых скобок:

$$A \oplus B = [(A \& \bar{A}) \vee (B \& \bar{A})] \vee [(A \& \bar{B}) \vee (B \& \bar{B})]$$

Но из правила исключенного третьего следует:

$$A \& \bar{A} \equiv 0,$$

$$B \& \bar{B} \equiv 0.$$

Таким образом:

$$A \oplus B = [0 \vee (B \& \bar{A})] \vee [(A \& \bar{B}) \vee 0]$$

Используя свойства логических констант, получаем окончательное выражение:

$$A \oplus B = (B \& \bar{A}) \vee (A \& \bar{B})$$

Вернувшись к диаграмме Венна (рис. 1.4), мы можем видеть, что первое логическое слагаемое  $(B \& \bar{A})$  определяет на диаграмме правый "полумесяц" заливки, а второе – левый "полумесяц".

## Глава 1. Математические основы информатики

### 1.4.4. Задания для самостоятельной работы

1. Какие из приведенных ниже предложений являются высказываниями:

а) Который час?

б) Целое число  $x=1$  есть наименьшее положительное целое число.

в) Если  $x = 3$ , то  $x^2 = 6$ .

г) Берегись автомобиля!

2. Пусть буквы  $a, b$ , обозначают высказывания:

$a$ : **Поездка в Италию стоит дорого.**

$b$ : **Летом я поеду в Италию.**

$c$ : **У меня нет денег.**

Запишите в символической форме высказывание:

**У меня нет денег и летом я не поеду в Италию.**

3. Для условий задачи 2 запишите в символической форме высказывание:

**Так как у меня нет денег, то летом я не поеду в Италию**

4. Для условий задачи 2 запишите в символической форме высказывание:

**Неверно, что у меня есть деньги и летом я поеду в Италию.**

5. Для условий задачи 2 запишите в символической форме высказывание:

**Неверно, что поездка в Италию стоит дорого и летом я не поеду в Италию.**

6. Для условий задачи 2 запишите в символической форме высказывание:

**Если поездка в Италию стоит недорого, то летом я поеду в Италию.**

7. Пусть буквы  $a, b$ , обозначают высказывания:

## Глава 1. Математические основы информатики

- a: **У Васи есть хороший компьютер.**
- b: **Вася защитит курсовой проект вовремя.**
- c: **Вася хорошо знает предмет..**

Сформулируйте словами высказывание, записанное в символической форме:

$$(c \& a) \rightarrow b$$

8. Для условий задачи 7 сформулируйте словами высказывание, записанное в символической форме:

$$(c \vee a) \rightarrow b$$

9. Для условий задачи 7 сформулируйте словами высказывание, записанное в символической форме:

$$(\bar{a} \& c) \rightarrow b$$

10. Для условий задачи 7 сформулируйте словами высказывание, записанное в символической форме:

$$(a \& \bar{c}) \rightarrow \bar{b}$$

11. Для условий задачи 7 сформулируйте словами высказывание, записанное в символической форме:

$$(\bar{a} \vee \bar{c}) \rightarrow \bar{b}$$

12. Для условий задачи 7 сформулируйте словами высказывание, записанное в символической форме:

$$c \sqcap b$$

13. Для условий задачи 7 сформулируйте словами высказывание, записанное в символической форме:

$$(a \& b) \vee (c \& b)$$

14. Постройте таблицу истинности высказывания из задачи 7.

15. Постройте таблицу истинности высказывания из задачи 8.



## Глава 1. Математические основы информатики

16. Постройте таблицу истинности высказывания из задачи 9.

17. Постройте таблицу истинности высказывания из задачи 10.

18. Постройте таблицу истинности высказывания из задачи 11.

19. Постройте таблицу истинности высказывания

$$(c \& a) \sim b$$

20. Постройте таблицу истинности высказывания

$$(c \& a) \vee b$$

21. Постройте таблицу истинности высказывания

$$(c \vee a) \& b$$

22. Постройте таблицу истинности высказывания

$$(\bar{a} \& \bar{b}) \rightarrow c$$

23. Постройте таблицу истинности высказывания

$$(a \oplus b) \vee c$$

24. Постройте таблицу истинности высказывания

$$c \rightarrow (\bar{a} \& \bar{b})$$

25. Постройте таблицу истинности высказывания

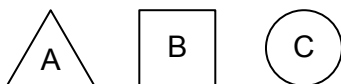
$$c \sqcap (\bar{a} \& b)$$

26. Постройте таблицу истинности высказывания:

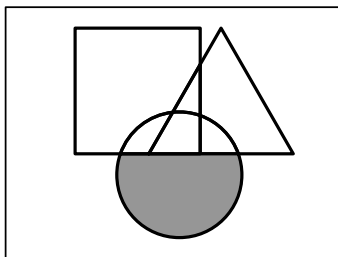
$$(a \oplus b) \& c$$

## Глава 1. Математические основы информатики

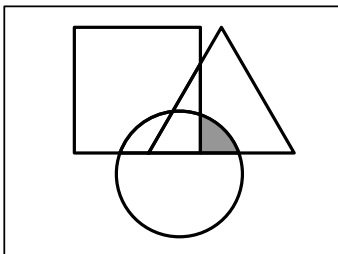
27. Пусть отдельные геометрические фигуры на диаграмме Венна соответствуют зонам истинности высказываний А, В, С:



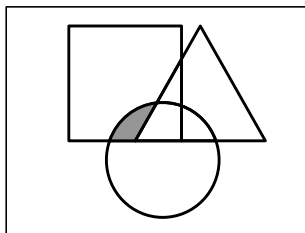
Какому логическому выражению соответствует заштрихованная часть плоскости:



28. Какому логическому выражению в условиях задачи 25 соответствует заштрихованная часть плоскости:

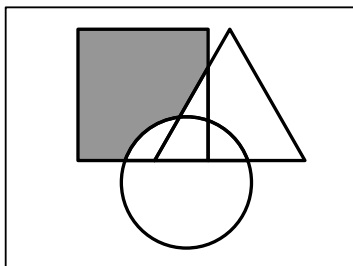


29. Какому логическому выражению в условиях задачи 25 соответствует заштрихованная часть плоскости:

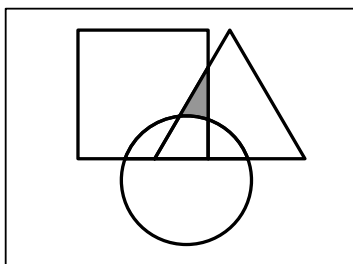


## Глава 1. Математические основы информатики

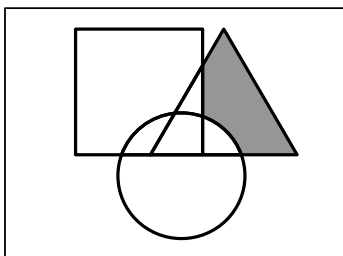
30. Какому логическому выражению в условиях задачи 25 соответствует заштрихованная часть плоскости:



31. Какому логическому выражению в условиях задачи 25 соответствует заштрихованная часть плоскости:

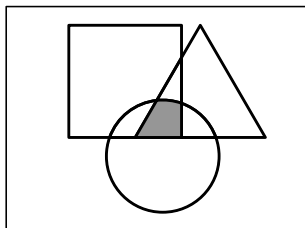


32. Какому логическому выражению в условиях задачи 25 соответствует заштрихованная часть плоскости:

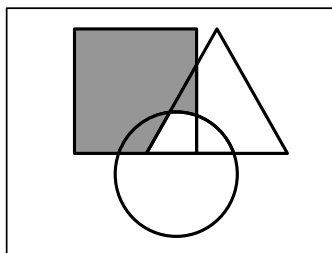


## Глава 1. Математические основы информатики

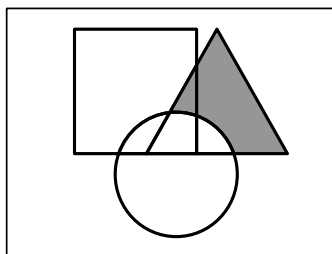
33. Какому логическому выражению в условиях задачи 25 соответствует заштрихованная часть плоскости:



34. Какому логическому выражению в условиях задачи 25 соответствует заштрихованная часть плоскости:

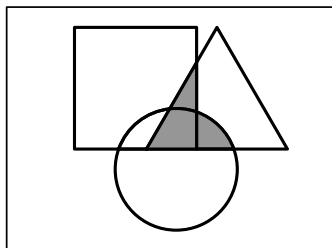


35. Какому логическому выражению в условиях задачи 25 соответствует заштрихованная часть плоскости:

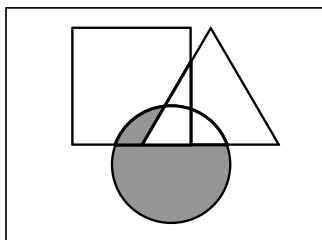


## Глава 1. Математические основы информатики

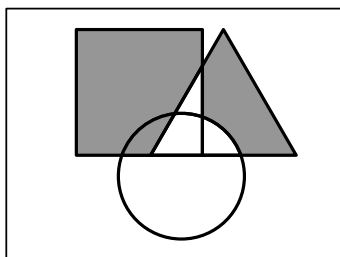
36. Какому логическому выражению в условиях задачи 25 соответствует заштрихованная часть плоскости:



37. Какому логическому выражению в условиях задачи 25 соответствует заштрихованная часть плоскости:

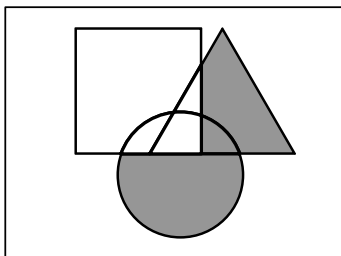


38. Какому логическому выражению в условиях задачи 25 соответствует заштрихованная часть плоскости:



## Глава 1. Математические основы информатики

39. Какому логическому выражению в условиях задачи 25 соответствует заштрихованная часть плоскости:



40. Используя законы математической логики, упростите высказывание, записанное в символической форме:

$$(A \vee B) \& (B \vee \bar{C}) \& \bar{B}$$

41. Используя законы математической логики, упростите высказывание, записанное в символической форме:

$$(B \& \bar{A}) \vee \overline{(A \vee B \vee C)}$$

42. Используя законы математической логики, упростите высказывание, записанное в символической форме:

$$\overline{(\bar{C} \& \bar{A})} \& (C \vee \bar{A})$$

43. Используя законы математической логики, упростите высказывание, записанное в символической форме:

$$[(A \vee B) \& C] \vee [\overline{(\bar{A} \& \bar{B})} \& \bar{C}]$$

44. Используя законы математической логики, упростите высказывание, записанное в символической форме:

$$[A \& \overline{(B \vee \bar{C})}] \vee [\overline{(\bar{A} \vee \bar{B})} \& \bar{C}]$$

45. Используя законы математической логики, упростите высказывание, записанное в символической форме:

$$(A \vee B) \& (B \vee C) \vee \bar{C}$$

46. Используя законы математической логики, упростите высказывание, записанное в символической форме:

$$(A \& B \& C) \vee (\bar{B} \vee \bar{C})$$

47. Используя законы математической логики, упростите высказывание, записанное в символической форме:

$$(A \& B) \vee (A \& \bar{B}) \vee (\bar{A} \& B) \vee \overline{(A \vee B)}$$

48. Используя законы математической логики, упростите высказывание, записанное в символической форме:

$$(A \& B) \vee (\bar{B} \& C \& A)$$

49. Используя законы математической логики, упростите высказывание, записанное в символической форме:

$$(A \& \bar{B}) \vee (B \& C) \vee \bar{B}$$

50. Используя законы математической логики, упростите высказывание, записанное в символической форме:

$$(A \& B) \vee \overline{(\bar{A} \& B)}$$

51. Используя законы математической логики, упростите высказывание, записанное в символической форме:

$$(A \& B \& C) \vee \left[ \overline{(\bar{B} \vee \bar{C})} \& \bar{A} \right]$$

52. Используя законы математической логики, упростите высказывание, записанное в символической форме:

$$(A \vee C) \& (B \vee C) \& (A \vee \bar{C}) \& (B \vee \bar{C})$$

### 1.5. Логические схемы

Цель работы: знакомство с логическими схемами, используемыми при проектировании узлов вычислительных устройств.

#### 1.5.1. Связь логических операций и двоичной арифметики

Сопоставление логического значения "Истина" с двоичной единицей, а значения "Ложь" с двоичным нулем позволяет связать арифметические операции с двоичными числами с операциями математической логики. Рассмотрим простейшее действие – сложение двух однозначных двоичных чисел. В общем виде оно может быть записано так:

$$a + b = cd,$$

где  $a, b, c, d$  – двоичные цифры 0 или 1. Существует четыре варианта такого действия:

$$0 + 0 = 00$$

$$0 + 1 = 01$$

$$1 + 0 = 01$$

$$1 + 1 = 10$$

Соответствующие значения двоичных цифр находятся в таблице 1.13.

Таблица 1.13

Сложение однозначных двоичных чисел

| <b>a</b> | <b>b</b> | <b>c</b> | <b>d</b> |
|----------|----------|----------|----------|
| 0        | 0        | 0        | 0        |
| 0        | 1        | 0        | 1        |
| 1        | 0        | 0        | 1        |
| 1        | 1        | 1        | 0        |

Сравнивая таблицу 1.13 с таблицами 1.9 и 1.10, можно видеть, что последняя цифра (**d**) образуется по правилу операции строгой дизъюнкции значений **a** и **b**, а предпоследняя (**c**) – по правилу конъюнкции тех же значений.



## Глава 1. Математические основы информатики

Аналогичным образом можно построить логические правила для других арифметических действий. Таким образом, двоичные арифметические операции можно заменить комбинациями операций математической логики.

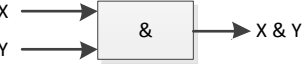
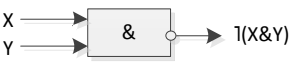
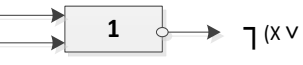
### 1.5.2. Логические элементы и логические схемы

Основой устройств вычислительной техники являются логические элементы – устройства, осуществляющие отдельные логические операции. Такие устройства легко реализуются в виде простых электронных узлов.

Всего насчитывается более 20 логических элементов. На логической схеме каждый из них обозначается определенным графическим блоком. Некоторые элементы показаны в таблице 1.14.

Таблица 1.14

Логические элементы

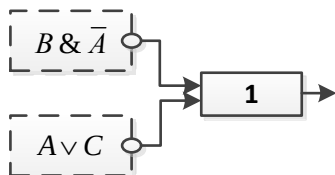
| Изображение элемента                                                                | Описание элемента                                                                                                  |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
|    | Элемент отрицания (НЕ) – инвертор                                                                                  |
|    | Элемент конъюнкции (И) – конъюнктор                                                                                |
|   | Элемент дизъюнкции (ИЛИ) – дизъюнктор                                                                              |
|  | Конъюнктор с отрицанием (И-НЕ). Изображение кружка на выходе означает дополнительную операцию отрицания результата |
|  | Дизъюнктор с отрицанием (ИЛИ-НЕ).                                                                                  |

При проектировании вычислительного устройства сначала создается его логическая схема – схема соединения и взаимодействия отдельных логических элементов – а уже затем на ее основе проектируется соответствующая электронная схема.

**Пример 1.15.** Построить логическую схему, реализующую выражение

$$F = \overline{(B \& \bar{A}) \vee (A \vee C)} \quad (1.23)$$

**Решение.** Из выражения (1.23) видно, что последним выполняемым действием является дизъюнкция двух выражений в скобках, к каждому из которых применена операция отрицания. Значит, последним элементом схемы должен быть дизъюнктор, соединяющий два выражения (в каждом из которых последней операций является отрицание результата вычисления выражения в скобках). Это можно представить следующей схемой:



В первом логическом слагаемом  $\overline{(B \& \bar{A})}$  выполняется отрицание конъюнкции. На схеме это должен представить элемент-конъюнктор с отрицанием. На входы этого конъюнктора подаются сигналы (0 или 1), соответствующие значениям выражений A или B, причем сигнал A должен быть предварительно пропущен через инвертор для получения сигнала  $\bar{A}$ . Второе выражение  $\overline{(A \vee C)}$  создается с помощью дизъюнктора с отрицанием, на входы которого подаются сигналы A и C. Окончательно получается схема, показанная на рис. 1.7.

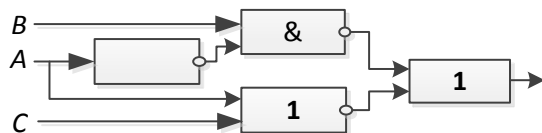


Рис. 1.7. Решение примера 1.14.

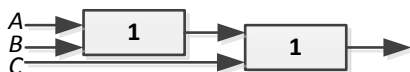
**Пример 1.16.** Построить логическую схему, реализующую выражение

$$A \vee B \vee C$$

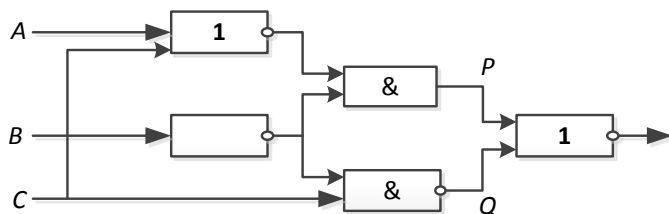
**Решение.** Рассматриваемое выражение представляет собой дизъюнкцию трех входных сигналов A, B, C. Однако логический элемент-дизъюнктор имеет только два входа. Используя закон ассоциативности (1.8), запишем выражение так:

$$(A \vee B) \vee C$$

Отсюда видно, что для реализации данного выражения необходимо использовать два дизъюнктора:



**Пример 1.17.** Записать символически выражение, реализуемое следующей логической схемой:



**Решение.** Как видно из рисунка, последним действием является отрицание, которое применяется к дизъюнкции двух выражений, обозначенных на схеме как P и Q. Поэтому общая структура искомого выражения выглядит так:

$$\overline{P \vee Q} \quad (1.24)$$

Выражение P является результатом конъюнкции, причем на вход конъюнктора подаются, с одной стороны, отрицание выражения B (т.к. входной сигнал B пропущен через инвертор), с другой – результат дизъюнкции A и C с последующим отрицанием. Тогда выражение P имеет вид:

$$P = \overline{(A \vee C)} \& \bar{B} \quad (1.25)$$

Выражение Q – это результат конъюнкции сигнала C и отрицания сигнала B:

$$Q = \overline{(\bar{B} \& C)} \quad (1.26)$$

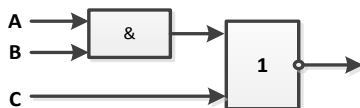
Подставив (1.25) и (1.26) в (1.24), получаем окончательно искомое выражение

$$\overline{\overline{[(A \vee C) \& \bar{B}] \vee (\bar{B} \& C)}}. \quad (1.27)$$

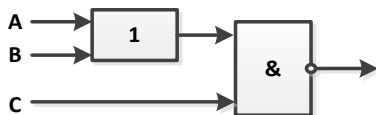
**Замечание.** Полученное выражение (1.27) можно упростить, используя законы математической логики. В результате получается выражение  $\bar{B} \& C$ . Следовательно, исходную логическую схему можно заменить эквивалентной более простой. Упрощение выражения (1.27) читателю предлагается выполнить самостоятельно.

### 1.5.3. Задания для самостоятельной работы

1. Запишите в символической форме выражение, соответствующее логической схеме:

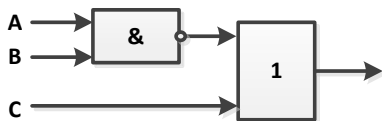


2. Запишите в символической форме выражение, соответствующее логической схеме:

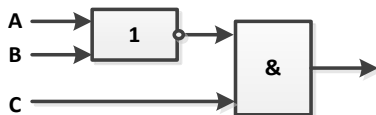


## Глава 1. Математические основы информатики

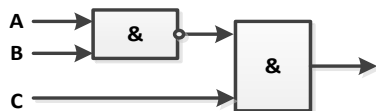
3. Запишите в символической форме выражение, соответствующее логической схеме:



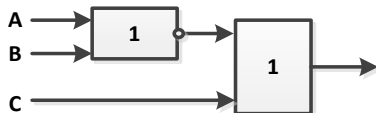
4. Запишите в символической форме выражение, соответствующее логической схеме:



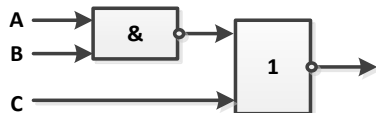
5. Запишите в символической форме выражение, соответствующее логической схеме:



6. Запишите в символической форме выражение, соответствующее логической схеме:

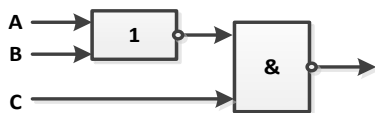


7. Запишите в символической форме выражение, соответствующее логической схеме:

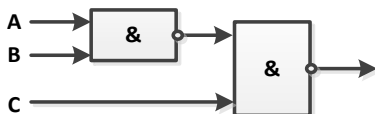


## Глава 1. Математические основы информатики

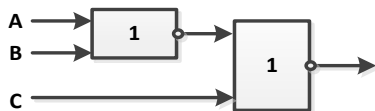
8. Запишите в символической форме выражение, соответствующее логической схеме:



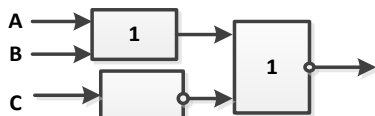
9. Запишите в символической форме выражение, соответствующее логической схеме:



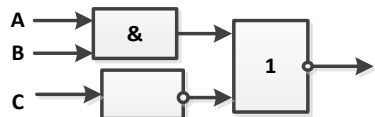
10. Запишите в символической форме выражение, соответствующее логической схеме:



11. Запишите в символической форме выражение, соответствующее логической схеме:

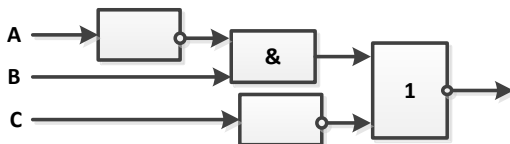


12. Запишите в символической форме выражение, соответствующее логической схеме:



## Глава 1. Математические основы информатики

13. Запишите в символической форме выражение, соответствующее логической схеме:



14. Постройте логическую схему, соответствующую выражению

$$\bar{A} \vee \bar{B} \& \bar{C}$$

15. Постройте логическую схему, соответствующую выражению

$$A \& \bar{B} \vee \bar{C}$$

16. Постройте логическую схему, соответствующую выражению

$$A \& B \& \bar{C}$$

17. Постройте логическую схему, соответствующую выражению

$$\bar{A} \vee B \vee \bar{C}$$

18. Постройте логическую схему, соответствующую выражению

$$A \& \bar{B} \& C$$

19. Постройте логическую схему, соответствующую выражению

$$A \vee \overline{(B \& C)}$$

20. Постройте логическую схему, соответствующую выражению

$$\bar{A} \& \bar{B} \& \bar{C}$$

## *Глава 1. Математические основы информатики*

21. Постройте логическую схему, соответствующую выражению

$$A \vee \overline{(A \vee B \vee C)}$$

22. Постройте логическую схему, соответствующую выражению

$$B \& \overline{(A \& B \& C)}$$

23. Постройте логическую схему, соответствующую выражению

$$(A \vee B) \& \bar{C}$$

24. Постройте логическую схему, соответствующую выражению

$$(A \& B) \vee (A \& C)$$

25. Постройте логическую схему, соответствующую выражению

$$(A \& B) \vee (B \& C)$$

26. Постройте логическую схему, соответствующую выражению

$$C \& ((A \vee B) \vee (A \& B))$$



### Глава 2. ВЫЧИСЛИТЕЛЬНЫЙ ПАКЕТ MathCAD

Пакет MathCAD является одним из наиболее популярных прикладных программных комплексов, предназначенных для проведения математических расчетов разной направленности и содержания. Популярность MathCAD обусловлена простотой интерфейса и достаточно широкими возможностями:

- вычисление значений сложных алгебраических выражений;
- наличие большого числа встроенных стандартных функций;
- выполнение операций с векторами и матрицами;
- выполнение операций математического анализа;
- возможность проведения расчетов в символьной форме;
- построение и оформление графиков функций одной и двух переменных;
- возможность создания встроенных программ;
- – и др.

Существует множество версий пакета MathCAD. Все они имеют схожий интерфейс, поэтому пользователь, освоивший любую из версий, легко может перейти к работе с другой версией.

#### 2.1. Знакомство с вычислительным пакетом MathCAD

Цель работы: освоение основных принципов работы с пакетом MathCAD.

##### 2.1.1. Интерфейс и структура документов MathCAD

На рисунке, изображающем окно MathCAD (рис. 2.1), можно увидеть стандартные элементы окна приложения ОС Windows: заголовок с кнопками управления, строку меню, стандартную панель инструментов и панель форматирования. Специфическим элементом является третья панель инструментов – панель вызова палитр. Палитра (Palette)– это особая тематическая панель, содержащая кнопки для создания элементов

математических формул и выполнения отдельных действий. Палитра вызывается на экран соответствующей ей кнопкой третьей панели инструментов и убирается управляющей кнопкой [×] (Закреть), имеющейся на палитре.

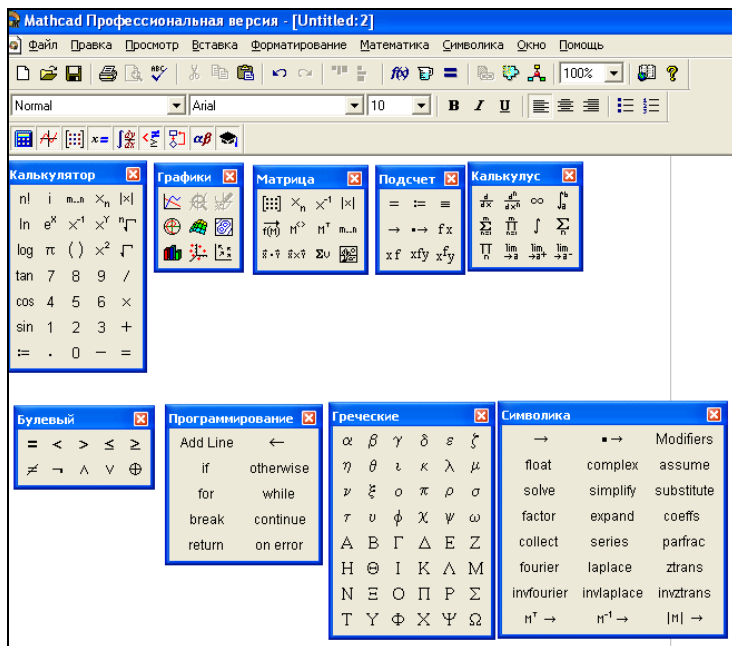


Рис. 2.1. Окно приложения MathCAD

На рисунке показаны 9 имеющихся в MathCAD палитр (в порядке следования вызывающих их кнопок):

1. Палитра организации вычислений "Калькулятор";
2. Палитра построения графиков;
3. Палитра матричных вычислений;
4. Палитра "Подсчет" ("Evaluation" в англоязычной версии). Операторы этой панели в настоящем практикуме не рассматриваются;
5. Палитра математического анализа;
6. Палитра знаков отношения (равенств и неравенств) и логических операций;
7. Палитра программирования;
8. Палитра ввода греческих букв;

### 9. Палитра организации символьных расчетов.

Окно документа представляет собой сплошное белое ("бумажное") поле, на котором пунктиром намечены границы отдельных бумажных страниц. Документ MathCad состоит из отдельных фрагментов, размещаемых в окне произвольным образом (без образования фиксированных строк или столбцов). Ввод очередного фрагмента начинается с позиции, отмеченной указателем в виде красного крестика, который устанавливается щелчком мыши в любую точку документа. Общим правилом является следующее:

**Фрагмент, расположенный ниже или правее, является последующим по отношению к любому фрагменту, расположенному выше или левее (такой фрагмент считается предшествующим). Информация, содержащаяся в любом предшествующем, по отношению к текущему, фрагменте, автоматически используется в текущем фрагменте для вычислений.**

Фрагмент может, в частности, содержать:

- Текстовый комментарий; ввод в такой фрагмент следует начать с кавычки (").
- Определение переменной, создаваемое так: с клавиатуры набирается имя переменной, затем знак '=' или ':' (любой из них автоматически преобразуется в оператор присваивания ': =' ) и числовое значение или выражение, значение которого определено (может быть вычислено). Имя должно состоять из букв и цифр и начинаться с буквы. При наборе чисел десятичным разделителем является точка. Новую переменную можно определить в любой момент. Желательно избегать повторного определения переменных с тем же именем и другим значением.

Если после определения переменной набрать ее имя и знак '=' (после определения переменной он уже не преобразуется в оператор присваивания), то после знака выводится численное значение переменной.

- Выражение – какую-либо математическую формулу. Выражения создаются с помощью клавиатуры и палитр по правилам, сходным с правилами набора математических формул в текстовом редакторе Word. Верхний и нижний индексы вводят-

ся кнопками палитры вычислений  $[x^y]$  и  $[x_n]$  и автоматически воспринимаются как показатель степени и индекс элемента вектора (одномерного массива) или матрицы (если индексов два, и они разделены запятой). Показатель степени можно также создать с помощью символа ' ^ ' на клавиатуре. Само по себе выражение – это просто некий текст. Если после ввода выражения набрать на клавиатуре знак '=', то значение выражения автоматически вычисляется, если оно содержит только числа или переменные, определенные в предшествующих фрагментах. При наличии неопределенных переменных выдается сигнал ошибки. С выражениями можно также проводить символьные (алгебраические) преобразования. При составлении выражений можно использовать большое число стандартных функций MathCAD. Они вызываются кнопкой  $[f(x)]$  на стандартной панели инструментов.

- Последовательность значений некоторой переменной в виде арифметической прогрессии. Для создания прогрессии надо набрать последовательно:

- имя переменной;
- знак '=' или ':' (автоматически преобразуется в ': =');
- значение первого элемента прогрессии;
- запятую;
- значение второго элемента<sup>1</sup>;
- точку с запятой ( ';' автоматически преобразуется в две точки<sup>2</sup>: '.. ')
- значение последнего элемента прогрессии.
- Определение пользовательской функции (см. далее лабораторную работу 2.2).

Возможны также другие варианты содержания фрагментов.

При вводе содержимого фрагмента с клавиатуры курсор имеет вид синего уголка, причем направление горизонтальной

---

<sup>1</sup> Распространенной ошибкой является набор величины шага прогрессии вместо второго значения последовательности. С точки зрения MathCAD это не воспринимается, как синтаксическая ошибка, но последовательность значений получается неверной.

<sup>2</sup> Нельзя просто набрать две точки на клавиатуре – только так, как описано!

черты указывает, с какой стороны от вертикальной стороны уголка будет происходить ввод (или удаление клавишей <Delete>) очередного символа.

Фрагменты можно передвигать мышью по окну документа, при этом может изменяться их логическая последовательность, что отражается на ходе вычислений. Если фрагмент содержит ошибку, его соответствующий элемент после ввода фрагмента выделяется красным цветом, а если навести на выделенный элемент мышь, появляется комментарий, содержащий описание ошибки.

**Замечание.** Стремление разработчиков пакета сделать его интерфейс максимально простым и понятным приводит в некоторых случаях к прямо противоположному результату. При какой-либо опечатке или незначительной ошибке в ходе набора сложных выражений MathCAD иногда начинает самостоятельно устанавливать структуру выражения, которая в итоге оказывается не такой, какая требовалась. Исправить такую ошибку бывает довольно сложно. Иногда проще оказывается стереть выражение полностью и аккуратно набрать его заново.

### 2.1.2. Практическая работа

Прежде всего установите указатель-крестик в верхней части поля документа и введите комментарий-заголовок **"Practical work 1."**

#### Вычисление простейших арифметических выражений.

а) Вычислите выражение:

$$\frac{(25 - 16/9)}{(5 - 4/3)}.$$

Порядок выполнения:

1) Установите щелчком мыши указатель-крестик в левом верхнем углу поля документа.

2) С клавиатуры наберите последовательность символов: **(25+16/9** . По окончании набора этой части выражения курсор редактирования (синий уголок) установится в знаменателе дроби сразу за цифрой 9.

3) Нажмите на клавиатуре клавишу  $\leftarrow \rightarrow$ . Курсор переместится из знаменателя на уровень основной строки набираемого выражения. Завершите ввод числителя, набрав на клавиатуре символ закрывающей скобки, затем наберите символ деления  $</>$ .

4) Аналогичным образом введите с клавиатуры знаменатель выражения. При этом не перемещайте курсор в основную строку аналогично п. 3, а сразу наберите символ закрывающей скобки. Убедитесь, что при этом скобка автоматически закрывается на уровне основной строки знаменателя (т.е., соответствует ближайшей к ней открывающейся скобке).

5) Нажмите клавишу  $<=>$ . В окне фрагмента появляется результат вычисления: число **6.333**.

### Замечания:

- Перемещение курсора для редактирования отдельных частей выражения производится клавишами со стрелками.

- Одиночные символы удаляются из выражения клавиатурными клавишами  $<Delete>$  и  $<Backspace>$ . Для удаления какой-либо части выражения целиком следует сначала выделить ее мышью.

- Если после набора выражения нажать не клавишу  $<=>$ , а клавишу  $\leftarrow \rightarrow$ , произойдет выход из окна фрагмента без вычисления выражения. В этом случае для вычисления надо установить указатель-крестик вблизи от фрагмента с правой стороны, затем клавишей  $\leftarrow \rightarrow$  перемещать его влево, пока он не окажется внутри окна фрагмента. При этом в окне появляется курсор (синий уголок) на уровне основной строки выражения. Нажатием клавиши  $<=>$  производится вычисление выражения.

б) Вычислите выражение

$$\frac{at^2}{2},$$

при значении переменных  $a=9.81$ ,  $t=5$ .

Порядок выполнения:

1) Щелчком установите указатель в начале новой строки поля документа и наберите на клавиатуре: **a=9.81**. Убедитесь,

что при наборе символ равенства преобразуется в символ присваивания автоматически.

2) Аналогично определите переменную **t**.

3) На клавиатуре наберите последовательность символов:  $a * t^2 / 2 =$ . Убедитесь, что произошло вычисление выражения.

4) Установите щелчком мыши курсор в одном из фрагментов, определяющих переменную (**a** или **t**). Введите с клавиатуры другое значение переменной. Убедитесь, что вычисленное значение выражения изменяется автоматически.

5) Щелкните мышью один из фрагментов, определяющих переменную (**a** или **t**). Наведите указатель мыши на рамку фрагмента так, чтобы он принял вид раскрытой ладони. Такой вид указателя показывает, что фрагмент можно перемещать мышью. Передвиньте фрагмент определения переменной, чтобы он оказался ниже фрагмента вычисления выражения. Убедитесь, что после нажатия клавиши ввода выражение становится неопределенным, а переменная, определение которой было передвинуто, выделилась красным цветом. Щелкните по ее обозначению и запомните появившееся описание ошибки: **"This variable or function is not defined above"** ("Эта переменная или функция не определена выше"). Верните фрагмент с определением переменной на прежнее место.

### Замечания:

- При вводе выражений обязательно вводить с клавиатуры символ операции умножения клавишей  $< * >$ .

- Обратите внимание, что отдельные множители, стоящие в числителе дробного выражения, автоматически записываются в основную строку. В состав числителя дроби записывается только один множитель. Это отличается от общепринятой формы записи математических выражений и является особенностью программы MathCad. Чтобы избежать этого, при формировании операндов сложной структуры следует использовать скобки. Открыв новый фрагмент, наберите на клавиатуре последовательность:  $(a * t^2 \rightarrow) / 2 =$ . Оцените, как изменился вид получившегося выражения; убедитесь, что вычисленное значение не изменилось.

в) Вычислите значение выражения

$$e^{-t^2}$$

с использованием:

- только ввода с клавиатуры;
- вставки стандартной функции.

1) Введите с клавиатуры последовательность символов:  **$e^{-t^2}=$** . Поскольку переменная **t** была определена ранее и ее значение равно 5, вычисленное значение выражения оказывается очень маленьким, поэтому оно будет записано в экспоненциальной форме:  $1.389 \times 10^{-11}$ . Чтобы получить более привычную запись ("с фиксированной точкой"), выполните команду "Форматирование" – "Результат" – в окне ввода "Format" мышью выберите из списка форматов: "Decimal", затем в окне "Число десятичных знаков" ("Number of decimal places") установите значение 13 и щелкните [OK].

2) Установив указатель ниже для начала нового фрагмента, щелкните на стандартной панели инструментов виртуальную кнопку [ **f(x)** ]. Выберите функцию "exp". Ее можно найти в категории "Log and exponential", либо в общем алфавитном списке (категория "All"). Категория выбирается в левой части появившегося окна диалога. После выбора функции введите в скобки с помощью клавиатуры аргумент, набрав:  **$-t^2=$** .

3) В следующем фрагменте введите с клавиатуры последовательность символов:  **$\exp(-t^2)=$** .

Сравните полученные выражения и результаты вычислений. Вы убедились:

- для создания выражений можно использовать различные способы ввода функций;
- в MathCad можно использовать общепринятые обозначения математических констант (таких, как числа "e" и "π").

### 2.1.3. Задания для самостоятельной работы

Вычислите приведенные в таблице 2.1 выражения согласно индивидуальному варианту (значения переменных задайте произвольно). Сохраните результаты.



### Предварительные замечания:

- Символы, отсутствующие на клавиатуре, вставляются с помощью панели организации вычислений "Калькулятор", которая вызывается на экран первой виртуальной кнопкой панели вызова палитр (рис. 2.1).
- В заданиях тригонометрические и гиперболические функции (**sin**, **cos**, **tg**, **sh**, **ch**, **th**, **arcsin**, **arctg**) записаны с помощью их традиционных математических обозначений. В системах программирования, в т.ч., MathCAD, они обозначаются соответственно **sin**, **cos**, **tan**, **sinh**, **cosh**, **tanh**, **asin**, **atan**. Для их ввода можно также использовать кнопку вызова функций на стандартной панели инструментов (категории Trigonometric и Hyperbolic).
- Функции **J<sub>0</sub>(x)**, **J<sub>1</sub>(x)** – функции Бесселя (специальные функции математической физики). Их обозначения можно ввести с клавиатуры (**J0**, **J1**) или с помощью кнопки вызова функций (категория Bessel).

Таблица 2.1

Варианты заданий работы 2.1

| Вариант | Выражение 1                                      | Выражение 2                                                        |
|---------|--------------------------------------------------|--------------------------------------------------------------------|
| 1       | $\frac{\sqrt[3]{125x+5!}}{(2-\sqrt{x}) \cdot 5}$ | $5 \ln x + \sin\left(\frac{3\pi}{2}x\right) + J_0(x)$              |
| 2       | $\frac{5!+6!}{2x^3} - 2\sqrt[3]{27x}$            | $10 \ln(1+x) - e^{\sin \pi x}$                                     |
| 3       | $\frac{2(3^2 + \sqrt{x^3})}{\pi^2}$              | $\sqrt{\exp(2x)} - \frac{\operatorname{th} x}{\exp(\cos(2\pi x))}$ |
| 4       | $\frac{ 5 + -5 }{\sqrt[8]{256}} + x^3$           | $\frac{\operatorname{th} x}{2\sqrt{\exp(x)}} - \sqrt{ \sin x }$    |

| Вариант | Выражение 1                                                                       | Выражение 2                                                                                 |
|---------|-----------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| 5       | $\frac{\left((2 + \sqrt{3x}) \cdot 3 + 5!\right)}{8^{0.25} + 2x}$                 | $ \sin x  +  \cos x  + 2 \operatorname{th} x$                                               |
| 6       | $\frac{2 + 3^{0.5}}{2 - 3^{0.5}} - x^{10} \sqrt{1024}$                            | $(\sin x + \cos x) \cdot (1 - \ln x)$                                                       |
| 7       | $2 \frac{3 - \sqrt{5x}}{2 + 3x^{0.67}} - 5^{1.5} \pi$                             | $\frac{\operatorname{sh} x}{\operatorname{th} x} - \frac{\operatorname{ch} x}{2}$           |
| 8       | $\left(2 - \frac{2x + \sqrt{3^2 - 2}}{2x \cdot \sqrt[3]{5x}}\right) \sqrt{10\pi}$ | $\sin\left(\frac{3\pi x}{2}\right) - J_0(x) + \operatorname{th} x$                          |
| 9       | $3^{0.5} + 3x^{0.75} / (\sqrt[3]{3} - 2/3)$                                       | $\ln(\operatorname{ch} x) \cdot \operatorname{th} x + \sqrt[3]{\sin x}$                     |
| 10      | $\frac{(3! + 4! + 5!)(x + y)}{\sqrt{x} + \sqrt{y}}$                               | $\operatorname{tg} x - \operatorname{th} x + \frac{J_1(x)}{\operatorname{ch} x}$            |
| 11      | $\frac{(\sqrt{x} + \sqrt{\pi})}{(\sqrt{x} - \sqrt{\pi})} \sqrt[3]{5 + xy}$        | $\frac{\sqrt{\operatorname{ch} x}}{2 \exp(x)} - \arcsin x \quad ( x  \leq 1)$               |
| 12      | $\sqrt{x} + \sqrt[5]{15y} + \frac{3!}{5!}$                                        | $\frac{\operatorname{arctg} x}{\exp(x)} + \frac{5}{2} \ln\left(1 + \frac{\sin x}{2}\right)$ |
| 13      | $\frac{x^{0.125} + \sqrt{y}}{(-1)^n 5!}$<br>(n - натуральное число)               | $\frac{5J_1(x)}{2} + x^{0.25} \ln x \quad (x > 0)$                                          |
| 14      | $\frac{n! + \frac{1}{n!}}{-\sqrt[9]{512}}$<br>(n - натуральное число не более 8)  | $\frac{5 \operatorname{tg} x}{\sqrt{\operatorname{ch} x}} + \frac{\ln x}{e^x}$              |
| 15      | $\sqrt{x} + 3^{2/3} \sqrt{x + y} + \sqrt[3]{\frac{x}{y}}$                         | $\sin\left(\frac{\pi x}{10}\right) + J_0(x) \cdot e^{-\frac{x^2}{2}}$                       |

| Вариант | Выражение 1                                              | Выражение 2                                                                                              |
|---------|----------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| 16      | $\sqrt{x+y} - \frac{2x}{\sqrt{y}} + 0.25(xy)^{3/5}$      | $\frac{\operatorname{arctg}(x^2 - 2y)}{\sin 3\pi x + e^{-y}}$                                            |
| 17      | $\sqrt[3]{x^2 + \sqrt{\frac{x}{y} + 10} + 10$            | $\sin(x^3) \operatorname{sh} \frac{y^2 + 1}{x} - J_0(xy^2)$                                              |
| 18      | $\frac{\sqrt[3]{8 + 3y - x^2} - 2}{\sqrt[3]{x^2 + y^3}}$ | $\frac{\ln(2x^2 - 5)}{e^{\sin \pi y} - 1}$                                                               |
| 19      | $\frac{y^e}{\sqrt{x^{3.2} + e^{3.2}}}$                   | $x \ln \left( \operatorname{sh} y + \frac{1}{2x+1} \right)$                                              |
| 20      | $\frac{10x^2 - 9}{\sqrt{4x^2 - 1}} + (y-1)^{2.5}$        | $\frac{n!}{\sqrt{n}} \sin \frac{\sqrt{n}}{(n+2)!}$<br>( $n$ – целое число от 1 до 5)                     |
| 21      | $\frac{3\sqrt[3]{6(x-5)^2}}{(x-3)^2 + 8y}$               | $\frac{1}{2} \frac{e^{\left(-\frac{x+1}{2}\right)}}{\left(1 + e^{\left(-\frac{x+1}{2}\right)}\right)^2}$ |
| 22      | $-2 + \sqrt[3]{2( x +1)^2(5-y^3)}$                       | $\frac{\sin x}{(1 + \operatorname{ch} x + \operatorname{sh} y)^2}$                                       |
| 23      | $\sqrt[3]{2(x-2^2)\left(8 - \frac{1}{ y }\right)} - 1$   | $\frac{e^{x \sin 5x} - 1}{x + \arcsin \left( x^2 \operatorname{th} \frac{6}{x} \right)}$                 |
| 24      | $\frac{2x(2^x + 3)}{x^2 + \frac{x}{4} + 5}$              | $\sin \left( \exp \left( x^2 \sin \frac{5}{x} \right) - 1 \right) +  y $                                 |
| 25      | $\sqrt[3]{(x-1)^2} - \sqrt[4]{ x-3 ^3}$                  | $\operatorname{arctg}(x J_0( x )) - y \operatorname{sh} \frac{1}{y}$                                     |

### 2.2. Табулирование функций и построение графиков

Цель работы: освоение приемов графического отображения результатов вычислений.

#### 2.2.1. Табулирование функций одной переменной и построение их графиков.

Задача табулирования функции требует задания последовательности значений аргумента и определения табулируемой функции. Последовательность значений аргумента задается в виде арифметической прогрессии.

Для определения пользовательской (нестандартной) функции надо набрать имя функции, список аргументов в круглых скобках (элементы списка разделяются запятыми), знак присваивания ':=' (в этом случае обязательно надо полностью набрать его на клавиатуре!) и выражение, которое может содержать числа, ранее определенные переменные, стандартные и ранее определенные пользовательские функции и имеющиеся в списке аргументы.

При табулировании стандартных функций MathCAD их определение не требуется.

Если теперь набрать в новом фрагменте имя функции и ее аргумента (в скобках), знак равенства и нажать клавишу <Enter>, то автоматически строится таблица значений функции при указанных в заданной прогрессии значениях аргумента. Следует иметь в виду, что при таком способе табулирования функции сами значения аргумента в таблицу не вставляются. Для них рядом надо построить отдельную таблицу, набрав на клавиатуре имя переменной, знак '=' и нажав клавишу ввода. При необходимости можно выровнять две полученные таблицы, перемещая содержащие их фрагменты документа мышью.

**Пример 2.1.** Построение таблицы значений функции Бесселя  $J_0(x)$  на отрезке  $[0,1]$  с шагом 0.25.

|                  |            |
|------------------|------------|
| x := 0, 0.25.. 1 |            |
| x =              | $J_0(x)$ = |
| 0                | 1          |
| 0.25             | 0.984      |
| 0.5              | 0.938      |
| 0.75             | 0.864      |
| 1                | 0.765      |

Рис. 2.2. Табулирование функции

Видно, что представленные в таблице значения не отформатированы и содержат разное число знаков после десятичной точки. Для форматирования чисел следует выделить фрагмент с таблицей и выполнить команду "Format" – "Result" – в диалоговом окне в списке числовых форматов выбрать "Decimal", задать с помощью мыши требуемое число знаков после точки ("Number of decimal places") и установить флажок "Show trailing zeros" ("Показывать незначащие нули").

Для построения графика функции надо на палитре построения графиков щелкнуть кнопку, соответствующую типу выбранного графика; например, для построения графика в прямоугольных (декартовых) координатах выбирается кнопка [X-YPlot]. Появляется окно, показанное на рис. 2.3. Последовательно устанавливая курсор в позиции ввода для описания осей, надо указать в них источники значений переменных – обозначения величин, определяющих координаты точек для построения, т.е.  $x$  и  $F(x)$  (для примера 2.1 вместо  $F(x)$  следует указать имя стандартной функции  $J_0(x)$ ). После указания каждого обозначения у концов соответствующего края рамки графика появляются две дополнительных позиции для ввода границ диапазона значений соответствующих величин, который будет отражен на графике. После заполнения всех позиций ввода надо нажать

<Enter> или щелкнуть мышью любую точку документа за пределами окна графика.

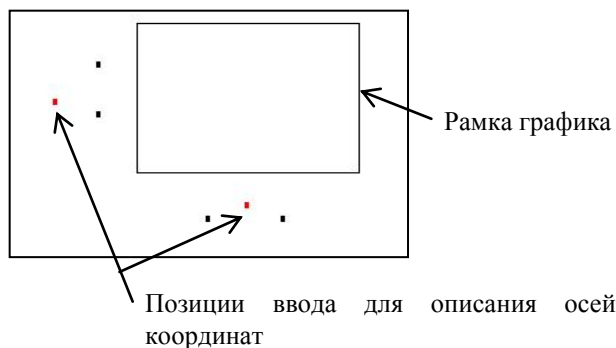


Рис 2.3. Окно построения графика в прямоугольных координатах.

Можно на одной координатной сетке строить несколько графиков. В этом случае после ввода обозначения в позицию описания координатной оси следует набрать запятую, и автоматически появится дополнительная позиция ввода для обозначения следующей величины.

Следует быть внимательным при выборе границ диапазонов значений для координатных осей. Если они не соответствуют реальным значениям соответствующих величин, график не будет виден, причем компьютер не воспринимает такую ситуацию, как ошибочную. Например, на рис. 2.4 показано построение графика функции  $f(x)=2 \cdot \sin(x)$ , причем заданы границы диапазона значений вертикальной координаты  $-1$  и  $2$ . В результате части графика, расположенные ниже уровня  $y = -1$ , оказались не видны.

Построенный график можно отредактировать с помощью команды меню "Format" – "Graph" – "X-Y Plot". При выполнении указанной команды форматирования на экране возникает окно диалога, показанное на рис. 2.5. На рисунке показана вкладка "Traces", предназначенная для форматирования линий графика.

Для каждой линии (Trace) графика можно установить следующие параметры:

1. "Legend label" (метка легенды). Легенда – поясняющая надпись, которая служит для распознавания отдельных графиков в случае, если их построено несколько на одной сетке. Отдельная метка содержит фрагмент линии и соответствующий этой линии текстовый комментарий (обычно – обозначение соответствующей функции). Для создания комментария надо с помощью мыши выделить (полосой цветного фона) строку с описанием нужной линии, затем щелчком установить курсор в окне ввода под столбцом "Legend label" и ввести с клавиатуры текст комментария.

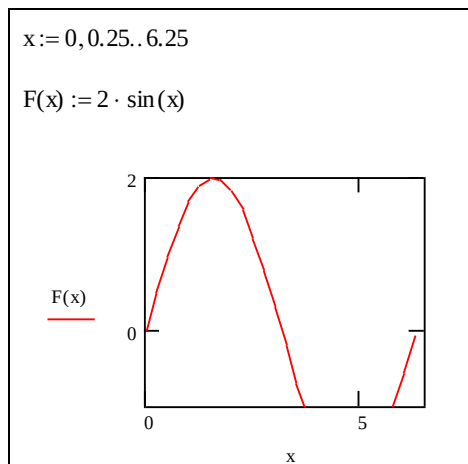


Рис. 2.4. Пример неправильного построения графика функции (неудачно задан диапазон изменения вертикальной координаты)

Следует иметь в виду, что по умолчанию легенда на график не выводится. Чтобы она появилась в поле графика, надо снять флажок "Hide legend" (Скрыть легенду).

2. "Symbol" – обозначение линии характерными маркерами. Маркеры выставляются в точках, соответствующих отдельным значениям аргумента из последовательности, по которой построен график. Возможные виды маркеров: "x" – крест, "+" – прямой крест, "box" – квадрат, "dmnd" ("diamond" – "алмаз") – ромб, "o" – кружок. Выбор символа маркера, как и

остальных параметров, производится мышью в списке, раскрываемом в находящемся под соответствующим столбцом окне ввода.

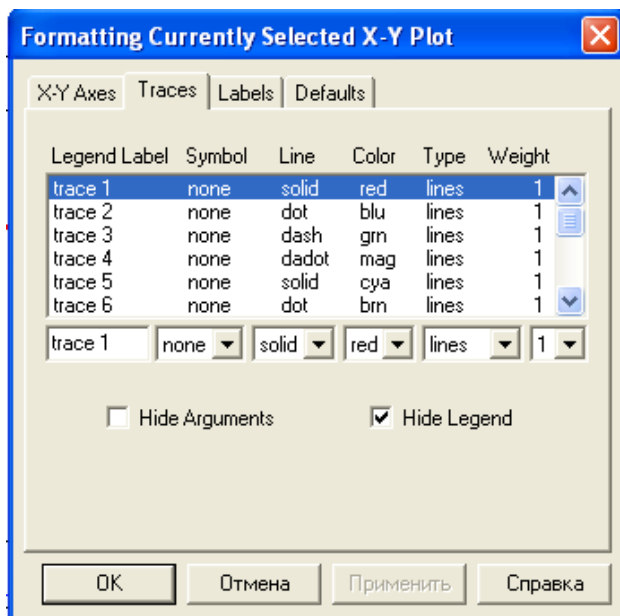


Рис. 2.5 Диалоговое окно форматирования графика.

3. "Line" – тип линии. Возможные варианты: "solid" – сплошная, "dot" ("точка") – пунктирная с короткими штрихами, "dash" ("тире") – пунктирная с длинными штрихами, "dadot" – штрих-пунктирная.

4. "Color" – цвет линии. Возможные варианты: "red" – красный, "blu" – синий, "grn" – зеленый, "mag" – розовый, "cya" – голубой, "brn" – коричневый, "blk" – черный, "wht" – белый (т.е., сокращения от английских названий соответствующих цветов).

5. "Type" – тип графика. В научных и технических документах в основном применяются типы "lines" (график изображается линией) и "points" (график изображается с помощью отдельных точек-маркеров, вид которых выбран в столбце "Symbol"). Тип "points" применяется для отображения на диаграммах результатов экспериментальных исследований. Тип "bar" обозначает гистограмму (график, в котором значение каж-



дого элемента исходной последовательности изображается отдельным прямоугольником, высота которого пропорциональна численному значению функции; гистограммы часто используются в экономической литературе). Прочие типы диаграмм в прямоугольных координатах встречаются редко.

6. "Weight" – толщина линии в условных единицах.

На рис. 2.6 показано построение на одной сетке двух графиков – функции  $\sin(2x)$  и функции Бесселя 1 рода 1 порядка  $J_1(2x)$ .

График функции  $\sin(2x)$  построен с помощью сплошной линии толщиной в 1 единицу и снабжен круглыми маркерами. График функции Бесселя построен пунктирной линией толщиной в 3 единицы. Внизу под полем графика видна легенда.

Вкладка диалогового окна "X-Y axes" служит для форматирования осей координат и координатной сетки. Параметры форматирования выбираются установкой или снятием соответствующих флажков на вкладке. В частности:

- флажок "Grid lines" позволяет нанести на график сплошные линии координатной сетки;
- флажок "Numbered" служит для нанесения на координатные оси разметки с подписями соответствующих значений;

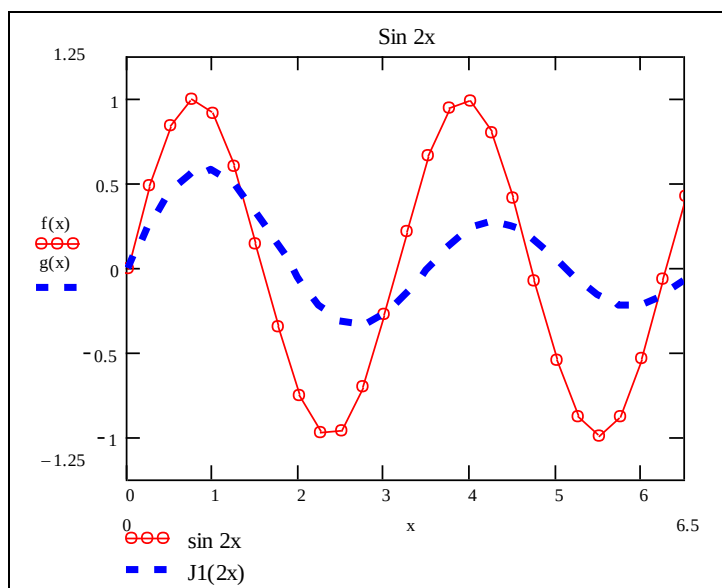


Рис. 2.6. Пример построения графиков функций.

- флажок "Auto Grid" включает автоматическое масштабирование разметки осей координат. При его снятии активизируется окно ввода "Number of Grids" для указания количества линий координатной сетки соответствующего направления, которые надо нанести на график. При этом автоматически будет определен масштаб разметки.

- флажок "Log Scale" служит для установки по соответствующей оси логарифмической шкалы разметки.

В нижней части вкладки находится переключатель "Axes Style" для выбора расположения координатных осей:

- "Boxed" – вдоль границ поля графика, независимо от расположения точки начала координат (так расположены оси на рис. 2.6);

- "Crossed" – оси пересекаются в начале координат и могут при этом проходить через поле графика.

Вкладка "Labels" служит для нанесения на график текстового заголовка и установки, при необходимости, дополнительных текстовых заголовков координатных осей. Заголовки вводятся с клавиатуры в предназначенные для них окна ввода и наносятся на график при установке соответствующих флажков. Например, в верхней части рис. 5 находится заголовок "Sin 2x".

### 2.2.2. Построение графика функции двух переменных.

Для построения графика функции двух переменных надо сначала определить функцию и задать последовательность значений каждого из аргументов (впрочем, последовательность можно и не задавать – тогда получится т.н. "график быстрого ввода", в котором диапазон изменения значений каждой переменной будет задан компьютером самостоятельно). Затем на палитре построения графиков следует найти и щелкнуть кнопку



В единственную позицию ввода в нижней части окна построения графика вводится имя функции. В результате по умол-

чанию строится так называемый "проволочный" график, в котором поверхность, представляемая функцией двух переменных, изображается сеткой линий, соответствующих различным конкретным значениям каждой из переменных (рис. 2.7).

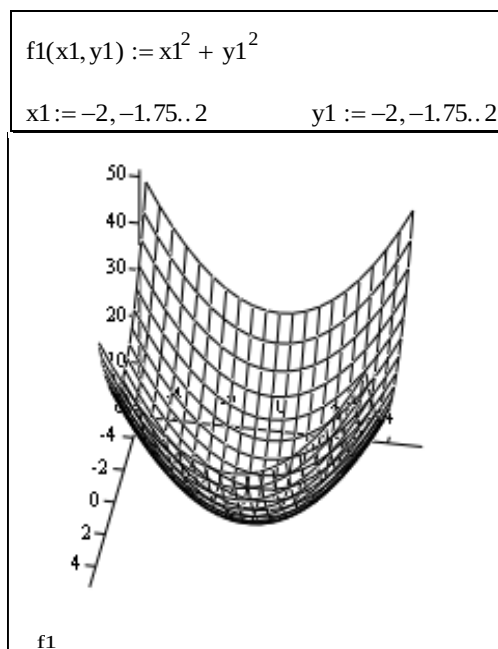


Рис. 2.7. График функции двух переменных

Если теперь дважды щелкнуть по полю графика, на экран вызывается окно форматирования графика (рис. 2.8). Его основные элементы:

Вкладка "General" ("Общие") содержит наиболее важные и часто используемые настройки.

Раздел "View" ("Вид") позволяет установить углы поворота графика вокруг трех координатных осей. Они называются вращение ("Rotation"), наклон ("Tilt") и поворот ("Twist"). Величины углов задаются в градусах. График можно также повернуть мышью, захватив конец одной из координатных осей.

Переключатель "Display as:" позволяет уточнить тип графика. Вместо проволочной диаграммы, изображающей поверхность ("Surface"), можно задать тип графика в виде набора кон-

турных линий уровня равных значений функции ("Contour plot"; такие линии соединяют на плоскости XY точки с одинаковыми значениями функции  $F(X,Y)$ ), или в виде отдельных точек, построенных в трехмерном пространстве ("Data points"). Возможны и другие варианты (не всегда доступные в конкретном случае).

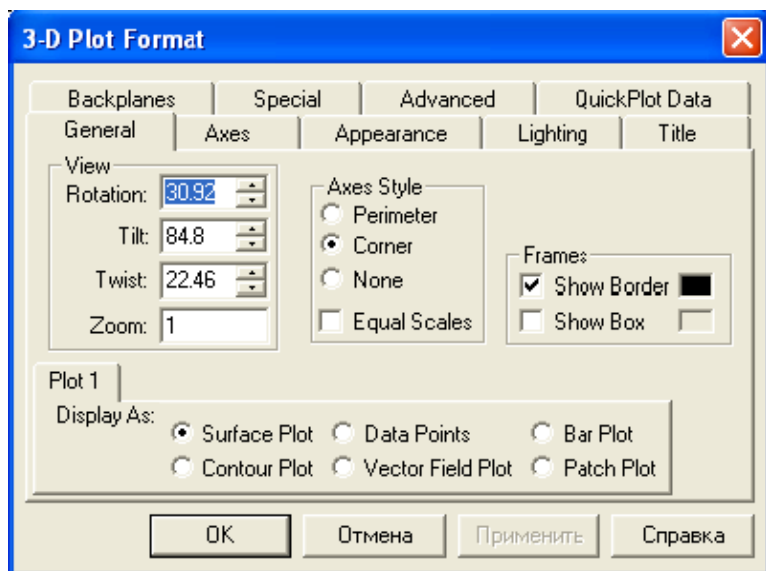


Рис. 2.8. Окно диалога для форматирования графика функции двух переменных.

Остальные элементы вкладки служат для оформления границ рисунка и указания расположения осей координат.

Результат применения выбранной настройки можно увидеть, не закрывая окно диалога, если щелкнуть клавишу [Применить]. Это позволяет пробовать разные варианты настроек и вовремя отказываться от неудачных вариантов, возвращаясь к предыдущим.

Вкладка "Backplanes" ("Задний план") служит для оформления задних по отношению к наблюдателю плоскостей воображаемого куба, в который помещен график (ребрами куба являются отрезки координатных осей). Таких плоскостей 3: X-Y, Y-Z, X-Z. Соответственно этому, вкладка, при ее выборе, распа-

дается на три, имеющих идентичное устройство. Плоскости можно подкрасить ("Fill Backplane "), обвести рамкой ("Backplane border"). С помощью разделов "Grids" и "Sub-Grids" можно задать для них наличие и оформление основных и вспомогательных линий координатной сетки, поставить метки ("Ticks") на осях координат.

Вкладка "Axes" ("Оси") позволяет выполнить настройки осей координат (она также распадается, при ее выборе, на три, соответственно трем осям). С помощью этой вкладки можно также задать наличие и настройку линий координатной сетки, аналогично предыдущей вкладке (но уже не для одной плоскости, а для двух, пересекающихся на выбранной оси). Разметка осей может быть автоматической (при установке флажков "Auto Grid" для выбора количества линий сетки и "Auto Scale" для выбора диапазона изменения значений координаты) или ее можно указать вручную, сняв флажки и указав самостоятельно количество линий ("Number"), максимальное и минимальное значения координаты ("Maximum Value", "Minimum Value").

Вкладка "Appearance" ("Вид") служит для художественного оформления поверхности графика – рисования и раскраски (заливки, "Fill") различными способами линий, ячеек, узловых точек.

Вкладка "Title" позволяет задать и разместить нужным образом заголовок диаграммы.

Остальные вкладки содержат сравнительно редко применяемые специализированные настройки. Более подробную информацию о них можно найти в руководствах по программе MathCad. Можно также поэкспериментировать с ними самостоятельно (интересные эффекты можно получить, меняя настройки вкладок "Advanced" - "Дополнительные" и "Lighting" - "Освещение")

### 2.2.3. Задания для самостоятельной работы.

1. Определите 2 различных функции, выражения которых выберите из таблицы 2.2 согласно индивидуальному варианту задания (табл. 2.3). Постройте на одной координатной сетке графики этих функций в указанных диапазонах аргументов[a,b] с указанным шагом (используйте для разных функций разные обозначения аргументов, иначе может возникнуть путаница и

искажение результатов!). Отформатируйте график, используя различные варианты настройки по собственному выбору или указанию преподавателя.

Таблица 2.2

Функции для табулирования и построения графиков

| №  | Функция $y = F(x)$            | a   | b  | шаг  |
|----|-------------------------------|-----|----|------|
| 1  | $x^2 - 3x + 2$                | 0   | 4  | 0,25 |
| 2  | $x \cdot \sin x$              | 0   | 6  | 0,25 |
| 3  | $5x^3 e^{-x}$                 | 0   | 5  | 0,25 |
| 4  | $(e^x - e^{-x}) / 2$          | -2  | 3  | 0.25 |
| 5  | $3x^{-2} \cdot \ln x$         | 0.5 | 5  | 0.25 |
| 6  | $2e^{-x/5} \sin (x / 2)$      | 0   | 10 | 0.5  |
| 7  | $2 \cdot \sin x \cdot \cos x$ | 0   | 6  | 0.25 |
| 8  | $x^2 \sqrt{4 - x^2}$          | -2  | 2  | 0.25 |
| 9  | $\sqrt{x} - \ln (x + 1)$      | 0   | 4  | 0.25 |
| 10 | $ \sin x $                    | -2  | 2  | 0.25 |
| 11 | $1 / \operatorname{ch} x$     | -3  | 3  | 0.25 |
| 12 | $x^2 / e^x$                   | 0   | 10 | 0.5  |
| 13 | $x^4 - x^3 + 2$               | -2  | 2  | 0.25 |
| 14 | $e^{ x } \cdot \cos x$        | -2  | 2  | 0.2  |
| 15 | $\sqrt{x^2 + 5x + 3}$         | 0   | 4  | 0.25 |
| 16 | $e^{-x} \sqrt{x^2 + 5x + 3}$  | 0   | 6  | 0.25 |

| №  | Функция $y = F(x)$        | a   | b | шаг  |
|----|---------------------------|-----|---|------|
| 17 | $\ln x \cdot e^{-x^2}$    | 0.5 | 5 | 0.25 |
| 18 | $x + \sin x$              | 0   | 5 | 0.25 |
| 19 | $x + e^{-x} \cos$         | -1  | 3 | 0.25 |
| 20 | $5(\sqrt{x} - \ln x - 1)$ | 1   | 3 | 0.1  |
| 21 | $5e^{-x^2/2}$             | -5  | 5 | 0.5  |
| 22 | $0.5 \sin x \cdot \ln x$  | 1   | 5 | 0.25 |
| 23 | $sh x \cdot ch x$         | -2  | 2 | 0.25 |
| 24 | $th x + 0.25 \sin 2x$     | -2  | 2 | 0.2  |
| 25 | $ch x - x^4$              | -2  | 2 | 0.25 |

Таблица 2.3

Индивидуальные варианты выбора функций

| № варианта | Номера функций из табл. 2.2 | № варианта | Номера функций из табл. 2.2 |
|------------|-----------------------------|------------|-----------------------------|
| 1          | 1, 14                       | 14         | 14, 5                       |
| 2          | 2, 15                       | 15         | 15, 6                       |
| 3          | 3, 16                       | 16         | 16, 7                       |
| 4          | 4, 17                       | 17         | 17, 8                       |
| 5          | 5, 18                       | 18         | 18, 9                       |
| 6          | 6, 19                       | 19         | 19, 10                      |
| 7          | 7, 20                       | 20         | 20, 11                      |
| 8          | 8, 21                       | 21         | 21, 12                      |
| 9          | 9, 22                       | 22         | 22, 13                      |
| 10         | 10, 23                      | 23         | 23, 14                      |
| 11         | 11, 24                      | 24         | 24, 15                      |
| 12         | 12, 25                      | 25         | 25, 16                      |
| 13         | 13, 26                      |            |                             |

2. Определите функцию двух переменных по варианту, заданному преподавателем. Постройте и отформатируйте график функции. Диапазоны изменения аргументов:  $[-2; +2]$ , шаг изменения 0.25 для обоих аргументов.

Таблица 2.4

Варианты функций двух переменных

| №  | $F(x, y)$                               | №  | $F(x, y)$                                         |
|----|-----------------------------------------|----|---------------------------------------------------|
| 1  | $3x^2 - 4y^2$                           | 14 | $\arctg (x + y)$                                  |
| 2  | $x^2 \sin (\pi y)$                      | 15 | $\arctg ( x  +  y )$                              |
| 3  | $0.125 \cdot x^2 y^3$                   | 16 | $\sin (xy)$                                       |
| 4  | $5 \sin (\pi x / 2) \cdot \cos (\pi y)$ | 17 | $\sin ( xy )$                                     |
| 5  | $1 - x^2 - y^2$                         | 18 | $\sin (\sqrt{x^2 + y^2})$                         |
| 6  | $\cos x \cdot \sin y$                   | 19 | $5 \cdot e^{-\sqrt{x^2 + y^2}} \cdot \arctg (xy)$ |
| 7  | $e^x \cdot \cos y$                      | 20 | $e^{x - \sqrt{x^2 + y^2}}$                        |
| 8  | $e^{-xy} \cdot xy$                      | 21 | $x^2 - 3 \cdot xy + y^2$                          |
| 9  | $ch x - ch y$                           | 22 | $x^2 - (x + y) - y^2$                             |
| 10 | $x^2 \cdot \sqrt{4 - y^2}$              | 23 | $x \cdot (x + 1) - y \cdot (y + 1)$               |
| 11 | $\ln(1 +  x  +  y )$                    | 24 | $ch ( y ) / ch ( x )$                             |
| 12 | $5 \cdot e^{-\sqrt{x^2 + y^2}}$         | 25 | $-x^2 \cdot th y$                                 |
| 13 | $\arctg(xy)$                            |    |                                                   |



### 2.3. Действия с векторами и матрицами

Цель работы: знакомство с основными средствами выполнения действий над векторами и матрицами.

#### 2.3.1. Палитра матричных операций

Для операций с матрицами служит отдельная палитра, показанная на рис. 2.9.

Назначение кнопок палитры (в порядке следования):

1. Создание матрицы;
2. Создание нижнего индекса. Эти кнопки уже упоминались в предыдущей работе;
3. Построение обратной матрицы (эта операция применима только к квадратным матрицам!);
4. Вычисление определителя (детерминанта) квадратной матрицы;

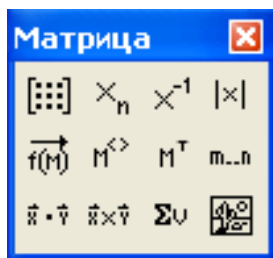


Рис. 2.9. Палитра матричных операций

5. Векторизация матриц. С помощью операции векторизации можно производить над каждым элементом матрицы однотипные операции. Например, для получения матрицы  $B$ , каждый элемент которой равен квадрату соответствующего элемента матрицы  $A$ , достаточно определить матрицу  $B$  выражением

$$B := \overline{A \cdot A} \quad (2.1)$$

Для этого надо набрать на клавиатуре  $B :=$ , затем щелкнуть на палитре кнопку векторизации и в позиции ввода, появившейся под стрелкой, набрать  $A * A$ .

6. Выделение столбца (именно столбца, не строки!).

7. Транспонирование матрицы;
8. Создание шаблона для указания диапазона индексов. Например, для указания диапазона индексов  $i$  на рис. 11 можно набрать на клавиатуре  $i:=$ , затем щелкнуть указанную кнопку и автоматически возникнет шаблон для указания целочисленной (с шагом 1) последовательности (1..5).
9. Операция скалярного умножения векторов. При осуществлении этой и двух следующих операций вектора должны быть представлены в виде вектор-столбцов.
10. Операция векторного умножения векторов.
11. Операция сложения векторов.
12. Кнопка создания изображений (в данном пособии не рассматривается).

Для производства матричных вычислений можно использовать также стандартные функции MathCad (категория "Vector and Matrix").

### 2.3.2. Создание матрицы

Для создания матрицы надо набрать на клавиатуре ее имя, символы присваивания:  $=$ , затем щелкнуть первую кнопку палитры. В появившемся диалоговом окне надо указать число строк ("Rows") и столбцов ("Columns"), затем щелкнуть [OK]. Во фрагменте появится заготовка матрицы нужной структуры.

Значения элементов матрицы можно вводить с клавиатуры по очереди, перемещая курсор по заготовке матрицы клавишами со стрелками, или переставляя его мышью. В качестве элементов матрицы могут использоваться числа или выражения.

Если в качестве элементов заданы выражения, для вывода на экран матрицы с числовыми значениями элементов надо набрать ее имя и знак  $=$ .

Можно также задать формулы для вычисления элементов матрицы в зависимости от индексов элементов. Для этого сначала задаются последовательности индексов для нумерации строк и столбцов. Последовательности индексов можно задать, как обычные арифметические прогрессии, или воспользоваться кнопкой создания шаблона для указания диапазона индексов на палитре матричных операций. Например, для указания диапазона индексов  $i$  надо набрать на клавиатуре  $i:=$ , затем щелкнуть

указанную кнопку и автоматически возникнет шаблон для указания целочисленной (с шагом 1) последовательности (1..5)

По умолчанию принято, что нумерация строк и столбцов начинается с 0. Если необходимо, чтобы нумерация строк и столбцов начиналась с 1, следует в начале документа вставить фрагмент с указанием

$$\text{ORIGIN} := 1. \quad (2.2)$$

После задания индексов в следующих фрагментах определяются выражения для вычисления элементов матрицы. Элемент обозначается указанием имени матрицы и индексов элемента. Для добавления к имени матрицы индексов элемента используются кнопки  $[X_n]$ , имеющиеся на палитрах организации вычислений и работы с матрицами. Индексы вводятся с клавиатуры через запятую.

После этого достаточно набрать имя матрицы и знак равенства. На экране появляется сформированная матрица. Она может быть выведена в традиционном виде (в скобках), а также в виде разлинованной таблицы с указанием номеров строк и столбцов. С такой таблицей можно проводить все обычные операции матричной алгебры (транспонирование, умножение, вычисление определителя квадратной матрицы и др.). Вид представления матрицы можно изменить, дважды щелкнув по ней. В возникшем окне диалога "Result Format" на вкладке "Display Options" ("Настройки представления") следует раскрыть список вариантов для окна ввода "Matrix Display Style" и выбрать вариант "Matrix" или "Table".

На рис. 2.10 показано создание матрицы из 5 строк и 2 столбцов. Диапазон значений индекса  $i$  определен, как обычная арифметическая прогрессия. Диапазон значений индекса  $j$  определен с помощью кнопки задания шаблона диапазона на палитре матричных операций. Можно видеть, что структура описания диапазона в этом случае выглядит проще.

Первый столбец матрицы содержит последовательность значений от 0 до 0.4 с шагом 0.1. Эти значения вычислены с помощью формулы в зависимости от значения индекса  $i$ . Для вычисления значений элементов второго столбца использованы операции векторизации и выделения столбца. Ссылка на определенный столбец матрицы создается с помощью кнопки выде-

ления столбца на палитре матричных операций. Для этого надо набрать имя матрицы и щелкнуть кнопку  $[M^{<}]$ . Рядом с именем матрицы в позиции верхнего индекса появятся угловые скобки. В них надо указать номер нужного столбца. Так, на рис. 2.10 указаны столбцы  $A^{<1>}$  и  $A^{<2>}$  вместо указаний элементов столбцов  $A_{i,1}$  и  $A_{i,2}$ .

```

ORIGIN := 1

i := 1..5      j := 1..5

Ai,1 := 0.1 · (i - 1)

A<2> :=  $\overrightarrow{\sin(A^{<1>})}$ 

A =
(
0.000  0.000
0.100  0.100
0.200  0.199
0.300  0.296
0.400  0.389
)
    
```

Рис. 2.10. Табулирование функции с помощью векторизации и выделения столбцов

В результате все значения во втором столбце вычисляются, как синусы значений соответствующих элементов первого столбца.

Построенная таким образом матрица представляет собой таблицу значений функции  $\sin(x)$ . В отличие от примера, показанного на рис. 2.2, таблица в виде матрицы представляет собой единый объект, а не два отдельных фрагмента, однако она не содержит заголовков столбцов, что делает ее менее наглядной.

### 2.3.3. Построение графика функции на основе матрицы

На основе созданной матрицы можно построить график функции, указывая в качестве источников значений переменных обозначения элементов матрицы.

На рис. 2.11 показано построение графика функции на основе матрицы.

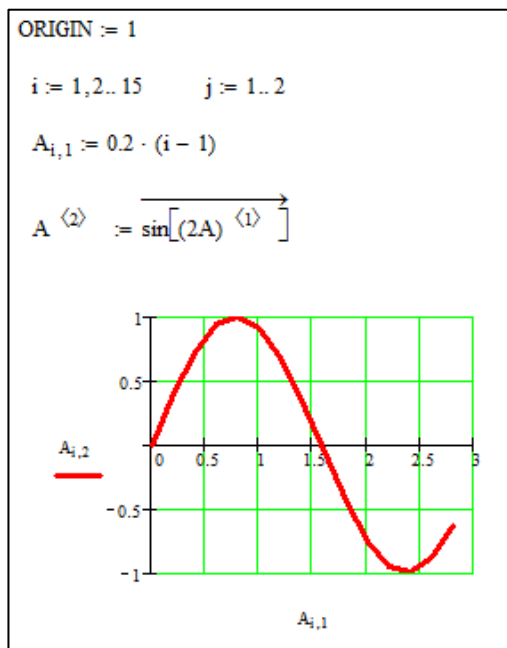


Рис. 2.11. Построение графика на основе матрицы

Матрица A задана путем определения формул для вычисления элементов, аналогично предыдущему примеру. Первый столбец содержит последовательность 15 значений арифметической прогрессии с шагом 0.2 на отрезке от 0 до 2.8. Второй столбец содержит значения функции  $\sin(2x)$ , аргументами для которой служат значения из первого столбца. Таким образом, матрица представляет собой таблицу значений функции.

По этим значениям построен график функции. При его построении в качестве источников значений для координатных осей были указаны выражения  $A_{i,1}$  и  $A_{i,2}$ .

### 2.3.4. Примеры действий с матрицами

На рис. 2.12 показаны примеры действий с матрицами. Сначала произведено создание двух матриц: A (3 строки и 4 столбца) и B (4 строки, 3 столбца). Обратите внимание на отсут-

ствие указания **ORIGIN:=1**; из-за этого в примерах на рис. 2.12 нумерация строк и столбцов начинается с 0.

Для матрицы В получена транспонированная матрица  $B^T$ . Затем матрицы А и В умножены друг на друга (это можно сделать, так как число строк матрицы А равно числу столбцов матрицы В). Результатом является матрица С.

$$\begin{aligned}
 A &:= \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{pmatrix} & B &:= \begin{pmatrix} 12 & 11 & 10 \\ 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{pmatrix} \\
 B^T &= \begin{pmatrix} 12 & 9 & 6 & 3 \\ 11 & 8 & 5 & 2 \\ 10 & 7 & 4 & 1 \end{pmatrix} \\
 C &:= A \cdot B \\
 C &= \begin{pmatrix} 60 & 50 & 40 \\ 180 & 154 & 128 \\ 300 & 258 & 216 \end{pmatrix} & D &:= \text{submatrix}(C, 0, 1, 0, 1) \\
 D &= \begin{pmatrix} 60 & 50 \\ 180 & 154 \end{pmatrix} \\
 D^{-1} &= \begin{pmatrix} 0.642 & -0.208 \\ -0.75 & 0.25 \end{pmatrix} & |D| &= 240 \\
 D^{-1} \cdot D &= \begin{pmatrix} 1 & -4.885 \times 10^{-15} \\ 0 & 1 \end{pmatrix}
 \end{aligned}$$

Рис. 2.12. Примеры действий с матрицами.

С помощью встроенной функции `submatrix(...)` из матрицы  $C$  вырезана подматрица  $D$  меньшего размера. Структура этой функции выглядит так:

**submatrix(имя обрезаемой матрицы, номер первой вырезаемой строки матрицы, номер последней вырезаемой строки, номер первого вырезаемого столбца, номер последнего вырезаемого столбца).**

В результате в матрицу  $D$  вошли элементы строк матрицы  $C$  с номерами от 0 до 1 и столбцов также с номерами от 0 до 1. Так как матрица  $D$  получилась квадратной, для нее вычислены определитель  $|D|$  и обратная матрица  $D^{-1}$ .

Отметим, что используя функцию `submatrix(...)`, можно вырезать из матрицы отдельную строку (тогда как кнопка  $[M^{\wedge}]$  на палитре позволяет вырезать только столбец). Для этого надо указать один и тот же номер нужной строки в качестве как второго, так и третьего аргументов функции `submatrix(...)`.

Далее вычислено произведение обратной матрицы  $D^{-1}$  и исходной матрицы  $D$ . Из курса высшей математики известно, что такое произведение дает единичную матрицу

$$E = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}. \quad (2.3)$$

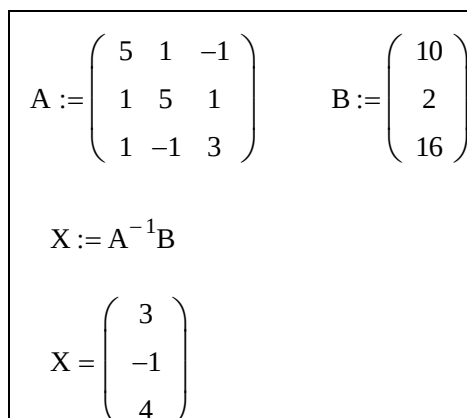
Видно, однако, что результат вычисления получился несколько иным – второй элемент первой строки отличается от нуля, хотя и является крайне малым. Это является результатом накапливания вычислительных погрешностей в ходе компьютерных вычислений. Обычно эта погрешность мала, но иногда она способна оказывать существенное влияние на ход и результаты расчетов.

### 2.3.5. Решение систем линейных уравнений с помощью матричных операций

На рис. 2.13 показано, как с помощью матричных вычислений в MathCAD можно решать системы линейных алгебраических уравнений. Такая система в общем виде может быть представлена следующим образом:

$$Ax = B, \quad (2.4)$$

где  $A$  – матрица коэффициентов,  $x$  – вектор-столбец неизвестных,  $B$  – вектор-столбец правых частей.



$$A := \begin{pmatrix} 5 & 1 & -1 \\ 1 & 5 & 1 \\ 1 & -1 & 3 \end{pmatrix} \quad B := \begin{pmatrix} 10 \\ 2 \\ 16 \end{pmatrix}$$

$$X := A^{-1}B$$

$$X = \begin{pmatrix} 3 \\ -1 \\ 4 \end{pmatrix}$$

Рис. 2.13. Решение системы линейных алгебраических уравнений

Например, для системы

$$\begin{cases} 5x_1 + x_2 - x_3 = 10 \\ x_1 + 5x_2 + x_3 = 2 \\ x_1 - x_2 + 3x_3 = 16 \end{cases} \quad (2.5)$$

такое представление имеет вид

$$\begin{pmatrix} 5 & 1 & -1 \\ 1 & 5 & 1 \\ 1 & -1 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 10 \\ 2 \\ 16 \end{pmatrix} \quad (2.6)$$

Умножив обе части уравнения (2.4) слева на матрицу, обратную по отношению к матрице  $A$ , получим

$$A^{-1}Ax = A^{-1}B \quad (2.7)$$

где  $A^{-1}$  – матрица, обратная по отношению к матрице  $A$ . Поскольку произведение матриц, обратных по отношению друг к другу, дает единичную матрицу, аналогичную (2.3), а произве-



деление единичной матрицы на вектор-столбец равно этому вектору, получаем из (2.7):

$$x = A^{-1}b \quad (2.8)$$

- то есть, находим вектор решения системы (рис. 2.13).

### 2.3.6. Примеры действий с векторами

На рис. 2.14 представлены примеры выполнения операций с векторами. Сначала определены два произвольных вектора **V** и **W** (в следующих формулах для обозначения векторов применяется стрелка над буквенным обозначением вектора или выделение этого обозначения жирным шрифтом). Затем найдены их сумма, скалярное и векторное произведения.

$$\begin{aligned} \mathbf{V} &:= \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} & \mathbf{W} &:= \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix} \\ \\ \mathbf{V} + \mathbf{W} &= \begin{pmatrix} 5 \\ 7 \\ 9 \end{pmatrix} & \mathbf{V} \cdot \mathbf{W} &= 32 & \mathbf{V} \times \mathbf{W} &= \begin{pmatrix} -3 \\ 6 \\ -3 \end{pmatrix} \\ \\ \mathbf{i} &:= \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} & \mathbf{j} &:= \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} & \mathbf{k} &:= \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \\ \\ \mathbf{i} \times \mathbf{j} &= \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} & \mathbf{j} \times \mathbf{k} &= \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} & \mathbf{i} \times \mathbf{k} &= \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix} \end{aligned}$$

Рис. 2.14. Примеры операций векторной алгебры.

Можно видеть, что эти величины вычислены в соответствии с правилами векторной алгебры

$$\begin{aligned}
 (\vec{V} + \vec{W})_i &= V_i + W_i, \\
 \vec{V} \cdot \vec{W} &= V_x W_x + V_y W_y + V_z W_z, \\
 \vec{V} \times \vec{W} &= \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ V_x & V_y & V_z \\ W_x & W_y & W_z \end{vmatrix} = \\
 &= (V_y W_z - V_z W_y) \vec{i} + (V_z W_x - V_x W_z) \vec{j} + (V_x W_y - V_y W_x) \vec{k}
 \end{aligned} \tag{2.9}$$

После этого определены три орта (вектора единичной длины, направления которых совпадают с направлениями координатных осей): **i**, **j**, **k**. Каждый из них, соответственно, имеет только одну проекцию, равную 1, а остальные две равны 0. После этого проверено выполнение известных тождеств векторной алгебры

$$\vec{i} \times \vec{j} = \vec{k}, \quad \vec{j} \times \vec{k} = \vec{i}, \quad \vec{i} \times \vec{k} = -\vec{j}. \tag{2.10}$$

### 2.3.7. Задания для самостоятельной работы.

1. Определите матрицу A, имеющую 3 строки и 4 столбца, и матрицу B, имеющую 4 строки и 4 столбца. Значения компонентов матриц задайте произвольно. Вычислите:

- а) Произведение этих матриц.
- б) Транспонированную матрицу  $A^T$ .
- в) Определитель матрицы B.
- г) Вектор-столбец, компонентами которого являются элементы 3 столбца матрицы A.

д) Вектор-строку, элементами которой являются элементы 2 столбца произведения матриц A и B. Для этого необходимо сначала выделить нужный столбец матрицы-произведения, а затем транспонировать полученный вектор-столбец для получения строки. Отдельные промежуточные результаты – матрицу-произведение, вектор-столбец – можно получить в виде отдельных объектов, дав им собственные имена. Можно использовать сложные выражения; например, выражение:  $((A \cdot B)^T)^{<2>}$  – дает в результате вычисления вектор-столбец, компонентами которого

являются элементы 2 столбца транспонированного произведения матриц **A** и **B**.

е) Обратную матрицу  $B^{-1}$ .

ж) Используя операцию векторизации, определите матрицу **A2**, элементами которой являются квадраты соответствующих элементов матрицы **A**.

2. Определите два произвольных вектора **C** и **D**.

а) Вычислите их сумму, скалярное и векторное произведения.

б) Определите третий вектор: **E**. Используя возможности MathCad, вычислите двойное векторное и смешанное произведения

$$\mathbf{C} \times (\mathbf{D} \times \mathbf{E}), \mathbf{C} \cdot (\mathbf{D} \times \mathbf{E})$$

– и проверьте выполнение тождеств:

$$\mathbf{C} \times (\mathbf{D} \times \mathbf{E}) = (\mathbf{C} \cdot \mathbf{E})\mathbf{D} - (\mathbf{C} \cdot \mathbf{D})\mathbf{E} \quad (2.11)$$

$$\mathbf{C} \cdot (\mathbf{D} \times \mathbf{E}) = \mathbf{D} \cdot (\mathbf{E} \times \mathbf{C}) = \mathbf{E} \cdot (\mathbf{C} \times \mathbf{D}), \quad (2.12)$$

а также формулу

$$\mathbf{C} \cdot (\mathbf{D} \times \mathbf{E}) = \begin{vmatrix} C_x & C_y & C_z \\ D_x & D_y & D_z \\ E_x & E_y & E_z \end{vmatrix}. \quad (2.13)$$

Для этого вычислите по отдельности выражения, стоящие в разных частях тождеств (2.11), (2.12) и формулы (2.13).

3. Используя средства MathCAD для работы с матрицами, решите систему линейных уравнений. Варианты заданий представлены в таблице 2.5 (**A** – матрица коэффициентов уравнений, **B** – вектор свободных членов).

Таблица 2.5

Варианты заданий по решению системы линейных уравнений

| А |  |  | В |
|---|--|--|---|
|---|--|--|---|

Вариант 1

|      |      |     |        |
|------|------|-----|--------|
| 4,6  | 1,5  | 8,6 | -10,07 |
| 6,8  | 2,5  | 1,1 | -52,78 |
| -6,9 | -9,2 | 4,7 | 116,23 |

| А |  |  | В |
|---|--|--|---|
|---|--|--|---|

Вариант 2

|      |      |      |        |
|------|------|------|--------|
| -1,1 | 2,3  | -5,6 | 66,76  |
| -0,2 | 2,6  | 8,8  | -73,28 |
| -0,3 | -5,2 | 1,8  | -34,38 |

Вариант 3

|     |      |      |        |
|-----|------|------|--------|
| -5  | 0,3  | 2    | -58,54 |
| 5,5 | -2,8 | 0,4  | 29,06  |
| 5,5 | 3,4  | -8,7 | 137,57 |

Вариант 4

|      |      |      |        |
|------|------|------|--------|
| -0,6 | 4,2  | -6,8 | 56,74  |
| -2,1 | -2   | -9,6 | 48,76  |
| -6,6 | -0,7 | 0,9  | -18,73 |

Вариант 5

|      |     |      |      |
|------|-----|------|------|
| -8,2 | 8,4 | 8,5  | 23,5 |
| 7,3  | 3,7 | -5,6 | -8,4 |
| -9,4 | 5,1 | 4,8  | 42,6 |

Вариант 6

|      |      |      |        |
|------|------|------|--------|
| 9,8  | -0,4 | -4,1 | -63,69 |
| -3,6 | 4,7  | -1,5 | 52,13  |
| -2,2 | -2,5 | 2,8  | -2,64  |

Вариант 7

|     |      |      |       |
|-----|------|------|-------|
| 2,8 | -0,2 | 7,6  | 77,48 |
| 0,1 | 2,9  | 1,1  | 17,87 |
| 3,9 | 8    | -8,6 | 0,11  |

Вариант 8

|      |      |     |        |
|------|------|-----|--------|
| 1    | -9,3 | 8,6 | 37,67  |
| -1,9 | 8,4  | 2,4 | -73,16 |
| -9,6 | 3,8  | 4,9 | -94,63 |

Вариант 9

|      |      |      |        |
|------|------|------|--------|
| -2,6 | 0,3  | -1,7 | 8,57   |
| 3,2  | -9   | -7   | -83,54 |
| 8,1  | -5,4 | -6,1 | -59,11 |

Вариант 10

|      |      |      |       |
|------|------|------|-------|
| 1,8  | 8,5  | -3,9 | 8,5   |
| -2,4 | -5   | 8,9  | 37,78 |
| 1,6  | -2,6 | 9,1  | 28,7  |

Вариант 11

|      |      |      |        |
|------|------|------|--------|
| 0,4  | -8,6 | -3,8 | -31,82 |
| -1,1 | -5,4 | 3,9  | -15,68 |
| 1,1  | -1,2 | -8,3 | -4,56  |

Вариант 12

|     |      |     |        |
|-----|------|-----|--------|
| 5,1 | 9,7  | 6,2 | 34,78  |
| 8,8 | -9,5 | 7,5 | 175,15 |
| 6,7 | -1,9 | 8,2 | 125,22 |

Вариант 13

|      |      |     |       |
|------|------|-----|-------|
| 3,7  | -2,2 | 6   | 17,74 |
| 0,8  | 6,5  | 5,6 | 84,89 |
| -8,5 | -5,6 | -8  | -62   |

Вариант 14

|     |     |      |       |
|-----|-----|------|-------|
| -1  | 1,3 | -5   | 10,33 |
| 3,1 | 5,8 | 5,8  | 18,11 |
| -9  | 7,7 | -8,3 | 72,57 |

Вариант 15

|      |     |      |        |
|------|-----|------|--------|
| 4,8  | 5,1 | 4,5  | -11,67 |
| -7,3 | 9,7 | -7,5 | 117,8  |
| 7,1  | 9   | 1    | 10,69  |

Вариант 16

|      |     |      |        |
|------|-----|------|--------|
| 0,9  | 5,7 | 3,7  | 43,81  |
| 4,2  | 4,7 | 6,4  | 64,97  |
| -1,9 | 4,4 | -4,7 | -29,18 |

Вариант 17

|      |      |      |       |
|------|------|------|-------|
| -2,2 | -9,7 | 8,1  | 51,18 |
| -5,2 | -5,1 | -7,5 | 59,4  |
| -6,9 | 9,2  | -0,8 | 0,56  |

Вариант 18

|     |      |      |        |
|-----|------|------|--------|
| 6,4 | -3,4 | -1,9 | -16,36 |
| 9,7 | -4,2 | -7,5 | -61,65 |
| 5   | 8,4  | -0,5 | 72,76  |

Вариант 19

|      |      |      |        |
|------|------|------|--------|
| 7    | 4    | -9,1 | 130,96 |
| -3,5 | -0,6 | -6,1 | 49,08  |
| 7    | 8,3  | 5,8  | 25,76  |

Вариант 20

|      |     |      |        |
|------|-----|------|--------|
| 7    | 3,6 | 0,8  | -4,04  |
| -5,5 | 2,2 | -7,2 | 40,02  |
| 4,2  | 2,1 | 7,7  | -31,43 |

Вариант 21

|     |      |      |        |
|-----|------|------|--------|
| -7  | 0,5  | 10   | -46,75 |
| 9,9 | -1,4 | -5,5 | 88,87  |
| 7,3 | -9,9 | -4,8 | 89,46  |

Вариант 22

|      |      |      |        |
|------|------|------|--------|
| 7,5  | 9,2  | 8,5  | -27,75 |
| 1,3  | -5,8 | -2,1 | 31,93  |
| -1,2 | -5   | 3,4  | 29,08  |

Вариант 23

|      |      |      |       |
|------|------|------|-------|
| -8,8 | -4,1 | -4,1 | 19,18 |
| 3,6  | 8,8  | -6,2 | 24,2  |
| -2,2 | 2,5  | 0,1  | -12,5 |

Вариант 24

|      |      |     |       |
|------|------|-----|-------|
| -1,9 | -7,8 | 7,4 | -65,8 |
| 6,6  | -5,8 | 2   | -9,68 |
| 4,3  | -4,8 | 8,6 | -44,6 |

Вариант 25

|      |      |      |        |
|------|------|------|--------|
| 0,9  | -8,2 | 4,5  | -12,57 |
| -6,5 | -8,3 | 7,1  | 38,24  |
| -6,6 | 5,1  | -9,5 | 11,62  |

### 2.4. Решение задач математического анализа

Цель работы: знакомство с основами применения операций математического анализа в MathCAD.

### 2.4.1. Палитра операций математического анализа

Для решения задач математического анализа используются инструментальные средства MathCAD, сосредоточенные в палитре "Calculus" ("Исчисление", имеется в виду дифференциальное и интегральное исчисление; в русскоязычном варианте эта панель так и называется – "Калькулус"). Внешний вид панели показан на рис. 2.15.

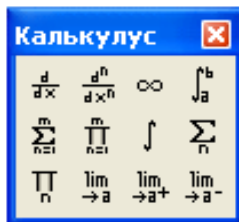


Рис. 2.15. Палитра операций математического анализа.

Назначение кнопок палитры (в порядке следования):

1. Ввод в выражение оператора первой производной.
2. Ввод оператора n-й производной
3. Ввод символа бесконечности "∞".
4. Оператор вычисления определенного интеграла.
5. Оператор вычисления суммы членов последовательности.
6. Оператор вычисления произведения членов последовательности.
7. Оператор вычисления неопределенного интеграла.
8. "Ранжированная" сумма.
9. "Ранжированное" произведение.
10. Предел функции в точке **a** (т.н. двусторонний предел).
11. Правосторонний предел функции в точке **a** (при стремлении  $x \rightarrow a$  с правой стороны).
12. Левосторонний предел функции в точке **a** (при стремлении  $x \rightarrow a$  с левой стороны).

Операторы неопределенного интеграла и пределов применяются только при символьных вычислениях. Остальные операторы могут производить как символьные, так и численные расчеты.

Вычисление ранжированных суммы и произведения обобщают операции вычисления суммы и произведения членов последовательности. Если при обычном суммировании или перемножении членов действие проводится по индексу, меняющемуся с шагом 1, то при ранжированном вычислении индекс задается в виде прогрессии с любым шагом изменения. Эта прогрессия должна быть определена прежде, чем производится вычисление. На рис. 2.16 приведен пример вычисления ранжированной суммы. Вычисляется сумма четных степеней числа 2, от  $2^0$  до  $2^8$ . Сначала определяется последовательность изменения индекса суммирования  $i$  (для наглядности она также выведена на рисунок), затем производится суммирование.

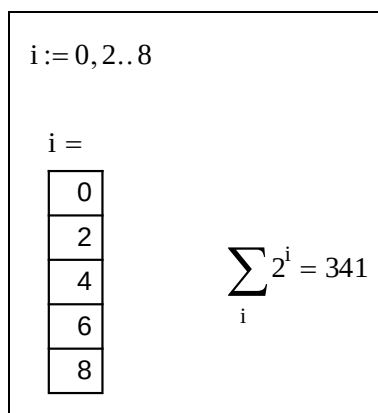


Рис. 2.16. Вычисление ранжированной суммы.

На рис. 2.17 приведены примеры численных расчетов с применением операторов математического анализа.

В первой паре фрагментов вычислено значение производной  $(\sin x)' = \cos x$  в точке  $x = \pi$ . Обратите внимание на то, что для определения численного значения производной сначала потребовалось задать численное значение аргумента, а уже потом вычислить значение производной. Для вычисления производной в другой точке необходимо задать новое численное значение аргумента и заново записать оператор вычисления производной. Это не всегда удобно, особенно, если вычисление значений одного и того же выражения приходится проводить несколько раз в разных точках.

$$\begin{array}{ll}
 1) & x := \pi \qquad \frac{d}{dx} \sin(x) = -1 \\
 2) & f(y) := \frac{d^2}{dy^2} (y^4) \qquad f(2) = 48 \\
 3) & \int_0^{\pi} \sin(z) \, dz = 2 \\
 4) & f1(x1) := \int_0^{x1} \sin(p) \, dp \\
 & f1(\pi) = 2 \\
 5) & \sum_{k=0}^{\infty} \frac{1}{k!} \qquad 2.718281828 \\
 & \sum_{k=1}^{\infty} (-1)^{k-1} \cdot \frac{\left(\frac{\pi}{4}\right)^{2k-1}}{(2k-1)!} \qquad .70710 \\
 6) & x3 := \frac{\pi}{4} \\
 & \sum_{k=1}^{\infty} (-1)^{k-1} \cdot \frac{(x3)^{2k-1}}{(2k-1)!} \qquad \sin(x3)
 \end{array}$$

Рис. 2.17. Примеры выполнения операций математического анализа



В этих случаях имеет смысл сначала определять вычисляемые выражения, как пользовательские функции, а уже потом производить их вычисление. Именно так оформлены вычисления в остальных фрагментах на рис. 2.17. В примерах 3,4 одно и то же вычисление определенного интеграла сначала произведено путем непосредственной записи вычисляемого выражения, а затем – через определение пользовательской функции.

### 2.4.2. Задания для самостоятельной работы.

1. Для одной из функций, которые табулировались в работе 2.2 (задание 1), определите пользовательские функции, выражающие:

- а) ее первую производную;
- б) ее вторую производную;

Используя эти определения, вычислите значения производных в трех точках, соответствующих расстояниям от левого конца отрезка табулирования, равным 0.25, 0.5 и 0.75 длины отрезка.

2. Вычислите значение определенного интеграла согласно приведенному индивидуальному варианту (табл. 2.6). Проверьте правильность вычисления, подсчитав точное значение интеграла по приведенному в таблице алгебраическому выражению.

Таблица 2.6

Варианты заданий по вычислению определенного интеграла

| № | Интеграл                             | Точное значение                                                                                                                      |
|---|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| 1 | $\int_2^3 \frac{dx}{\sqrt{x}(4-9x)}$ | $\frac{1}{6} \left[ \ln \left  \frac{2+3\sqrt{3}}{2-3\sqrt{3}} \right  - \ln \left  \frac{2+3\sqrt{2}}{2-3\sqrt{2}} \right  \right]$ |
| 2 | $\int_0^2 \frac{x \cdot dx}{16+x^4}$ | $\frac{\pi}{32}$                                                                                                                     |

| №  | Интеграл                                        | Точное значение                                                                             |
|----|-------------------------------------------------|---------------------------------------------------------------------------------------------|
| 3  | $\int_0^1 \frac{x^2 \cdot dx}{(x^2 + 1)^2}$     | $\frac{1}{4} \left( \frac{\pi}{2} - 1 \right)$                                              |
| 4  | $\int_1^2 \frac{dx}{(4+x)\sqrt{x}}$             | $\operatorname{arctg} \left( \frac{\sqrt{2}}{2} \right) - \operatorname{arctg} \frac{1}{2}$ |
| 5  | $\int_1^2 \sqrt{3+4x} \cdot dx$                 | $\frac{1}{6} \left( \sqrt{11^3} - \sqrt{7^3} \right)$                                       |
| 6  | $\int_0^1 \frac{x^3 \cdot dx}{(4-x^2)^{2.5}}$   | $\frac{1}{3} - \frac{5}{9\sqrt{3}}$                                                         |
| 7  | $\int_0^1 \frac{\sin x \cdot dx}{1 - \sin x}$   | $\operatorname{tg} \left( \frac{\pi}{4} + \frac{1}{2} \right) - 2$                          |
| 8  | $\int_0^1 \frac{x \cdot dx}{\cos^2 x}$          | $\operatorname{tg} 1 + \ln(\cos 1)$                                                         |
| 9  | $\int_0^1 \frac{\sin^3 x}{\cos^2 x} \cdot dx$   | $\cos 1 + \frac{1}{\cos 1} - 2$                                                             |
| 10 | $\int_0^2 \frac{x \cdot e^x \cdot dx}{(1+x)^2}$ | $\frac{e^2}{3} - 1$                                                                         |

| №  | Интеграл                                                | Точное значение                                                                                                                                                                                                  |
|----|---------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 11 | $\int_0^1 \ln(x + \sqrt{1+x^2}) \cdot dx$               | $\ln(1 + \sqrt{2}) + (1 - \sqrt{2})$                                                                                                                                                                             |
| 12 | $\int_2^3 \frac{\ln(x + \sqrt{x^2 - 4})}{x^3} \cdot dx$ | $\frac{\ln 2}{8} + \frac{\sqrt{5}}{24} - \frac{\ln(3 + \sqrt{5})}{18}$                                                                                                                                           |
| 13 | $\int_1^2 \frac{dx}{\operatorname{sh}^3 x}$             | $\frac{\operatorname{ch} 1}{2 \operatorname{sh}^2 1} + \frac{1}{2} \ln\left(\operatorname{th} \frac{1}{2}\right) - \frac{\operatorname{ch}^2 2}{2 \operatorname{sh}^2 2} - \frac{1}{2} \ln(\operatorname{th} 1)$ |
| 14 | $\int_0^1 \operatorname{th}^2 x \cdot dx$               | $1 - \operatorname{th} 1$                                                                                                                                                                                        |
| 15 | $\int_3^4 \frac{dx}{x(8 - x^3)}$                        | $\frac{1}{24} \ln \frac{152}{189}$                                                                                                                                                                               |
| 16 | $\int_0^1 \frac{x^2 \cdot dx}{4 - x^2}$                 | $\ln 3 - 1$                                                                                                                                                                                                      |
| 17 | $\int_0^1 \frac{x \cdot dx}{(\sqrt{x^2 + 4})^3}$        | $\frac{1}{2} - \frac{1}{\sqrt{5}}$                                                                                                                                                                               |
| 18 | $\int_1^2 \frac{\sqrt{4 - x^2}}{x^4} \cdot dx$          | $\frac{\sqrt{3}}{4}$                                                                                                                                                                                             |

| №  | Интеграл                                      | Точное значение                                        |
|----|-----------------------------------------------|--------------------------------------------------------|
| 19 | $\int_0^1 \frac{\cos x}{1 + \cos x} \cdot dx$ | $1 - \operatorname{tg} \frac{1}{2}$                    |
| 20 | $\int_0^1 \frac{dx}{1 + \cos x + \sin x}$     | $\ln \left  1 + \operatorname{tg} \frac{1}{2} \right $ |
| 21 | $\int_0^1 \operatorname{tg}^2 x \cdot dx$     | $\operatorname{tg} 1 - 1$                              |
| 22 | $\int_0^1 e^x \sin x \cdot dx$                | $\frac{e}{2} (\sin 1 - \cos 1) + \frac{1}{2}$          |
| 23 | $\int_0^1 \frac{dx}{1 + e^x}$                 | $\ln \frac{2e}{e+1}$                                   |
| 24 | $\int_1^2 x^3 \ln x \cdot dx$                 | $\frac{64 \ln 2 - 15}{16}$                             |
| 25 | $\int_1^2 \sin (\ln x) \cdot dx$              | $\sin (\ln 2) - \cos (\ln 2)$                          |

### 2.5. Символьные вычисления в MathCad.

Цель работы: знакомство с возможностями символьных вычислений в MathCAD.

#### 2.5.1. Палитра символьных операций

Палитра символьных операций показана на рис. 2.18.

Ее основные кнопки:

[  $\rightarrow$  ] – оператор символьного вычисления. Если использовать его вместо знака равенства после записи некоторого выражения, оно вычисляется в символьном виде (в виде формулы) вместо числового значения.

[simplify] – "Упростить". Упростить выражение (выполнить арифметические операции, привести подобные, сократить дроби, использовать для упрощения основные тождества и т.п.).

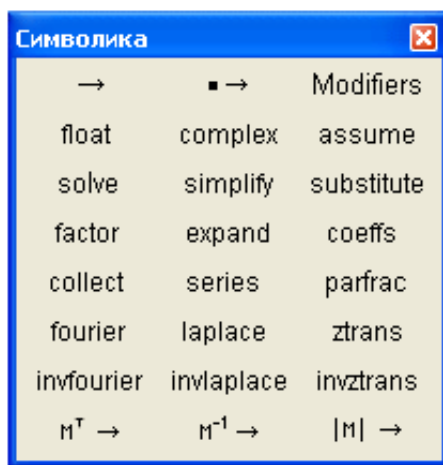


Рис. 2.18. Палитра символьных вычислений.

[expand] – "Развернуть". Раскрыть скобки, перемножить и привести подобные.

[factor] – "Разложить на множители".

[substitute] – "Подставить". Заменить в алгебраическом выражении букву или выражение другим выражением.

[parfrac] – Разложить рациональную дробь на простейшие дроби.

[series] – "Разложить в ряд". Разложить функцию в ряд Тейлора.

[solve] – "Решить". Разрешить выражение относительно указанной переменной; если выражение представляет собой функцию переменной, то найти корни этой функции (значения аргумента, при которых функция принимает нулевое значение).

Кнопки нижнего ряда палитры служат для производства матричных операций: транспонирования, вычисления обратной матрицы и определителя. Их действие незначительно отличается от действия соответствующих кнопок палитры матричных вычислений (см. ниже примеры вычислений).

Сведения об остальных кнопках данной палитры можно получить в более подробных руководствах по пакету MathCad.

### 2.5.2. Порядок выполнения символьных вычислений

Общий порядок символьных вычислений таков. С клавиатуры вводится некое символьное выражение, затем с помощью палитры в этот же фрагмент вводится оператор символьного вычисления [  $\rightarrow$  ]. После щелчка мышью по полю документа в стороне от фрагмента производится символьное вычисление.

В случае использования специальных символьных операторов ("simplify", "expand" и т.п.) следует сначала ввести с помощью палитры сам оператор а затем заполнить готовые позиции ввода.

Если позиция всего одна (оператор "simplify"), в нее вводится выражение, которое надо преобразовать.

Операторы "solve", "factor", "expand", "parfrac" содержат по две позиции ввода. В первую (слева от имени оператора) вводится выражение, во вторую (справа) – имя переменной, относительно которой производится символьное вычисление (см. ниже примеры использования оператора "solve", рис. 2.20). Если указание переменной не требуется, вторую позицию ввода можно удалить, установив курсор справа от нее и воспользовавшись клавиатурной клавишей <Backspace> (см. примеры применения оператора "factor").

Оператор "series" содержит три позиции ввода (рис. 2.21). В первую вводится выражение функции, которую следует раз-

ложить в ряд. Во второй указывается имя переменной, относительно которой производится разложение. В третьей позиции задается точность разложения в виде целого числа, которое определяет порядок первого отброшенного члена разложения (см. ниже примеры). Если во второй позиции просто указать имя переменной, то разложение в ряд производится в окрестности нулевой точки. Для разложения в ряд в окрестности другой точки следует задать значение переменной, по которой ведется разложение, с помощью символа логического равенства (на панели логических операций "Boolean" – "Булевский").

### 2.5.3. Алгебраические преобразования

На рис. 2.19, 2.20 приведены примеры применения символьных операторов при проведении преобразований алгебраических выражений.

На рис. 2.19 показан пример преобразования алгебраического выражения путем замены одной из переменных в нем ( $z$ ) на выражение  $(p+q)$ . При этом во второй позиции ввода с помощью символа логического равенства указано, какая переменная каким выражением заменяется.

$$z^2 + b \cdot z + c \text{ substitute, } z = p + q \rightarrow (p + q)^2 + b \cdot (p + q) + c$$

Рис. 2.19. Операция подстановки.

Более сложные действия показаны на рис. 2.20. Из примеров 3, 4 видно, что операция "factor" может производиться и без заполнения второй позиции ввода (указания переменной)<sup>1</sup>. В то же время, некоторые операции требуют обязательного указания переменной, даже если выбор переменной очевиден (т.е., других переменных в выражении просто нет)<sup>2</sup>. Поэтому успешное использование символьных вычислений требует некоторого опыта, который вырабатывается путем проб и ошибок. Если позиция ввода переменной была удалена ошибочно, ее можно вос-

---

<sup>1</sup> Более того, без указания переменной операция "factor" проходит успешнее.

<sup>2</sup> В частности, обязательное указание переменной требуется для операции разложения на простые дроби ("partfrac").

становить, просто набрав с клавиатуры имя переменной (через запятую справа от имени оператора).

Примеры 6-8 показывает ограниченность возможностей MathCad. Известно, что квадратный трехчлен  $z^2+bz+c$  можно разложить на множители и представить в виде  $(z-z_1)(z-z_2)$ , где  $z_1, z_2$  – корни трехчлена. Однако в примере 6 MathCad не справляется с этой задачей, представляя в качестве результата исходный трехчлен. В то же время, он позволяет найти корни трехчлена с помощью операции "solve" (пример 7), а затем с помощью операции "expand" решить обратную задачу – преобразовать выражение  $(z-z_1)(z-z_2)$  в трехчлен  $z^2+bz+c$  (пример 8).

$$\begin{array}{ll}
 1) \quad \frac{2+y-3y^2}{9y^2-4} \text{ simplify} \rightarrow \frac{-(y-1)}{(3 \cdot y-2)} & 2) \quad \frac{(a+b)^3 - (a-b)^3}{2 \cdot b \cdot (3a^2 + b^2)} + 1 \text{ simplify} \rightarrow 2 \\
 3) \quad 6a^2b + 2 \cdot b^3 \text{ factor} \rightarrow 2 \cdot b \cdot (3 \cdot a^2 + b^2) & 4) \quad a^2 - z^2 \text{ factor} \rightarrow (a-z) \cdot (a+z) \\
 5) \quad (y-3) \cdot (y+1) \text{ expand} \rightarrow y^2 - 2 \cdot y - 3 \\
 6) \quad z^2 + 4 \cdot z - 3 \text{ factor} \rightarrow z^2 + 4 \cdot z - 3 & 7) \quad z^2 + 4 \cdot z - 3 \text{ solve, } z \rightarrow \begin{pmatrix} -2 + \sqrt{7} \\ -2 - \sqrt{7} \end{pmatrix} \\
 8) \quad (z+2-\sqrt{7}) \cdot (z+2+\sqrt{7}) \text{ expand, } z \rightarrow z^2 + 4 \cdot z - 3 \\
 9) \quad z^2 + b \cdot z + c \text{ solve, } z \rightarrow \begin{bmatrix} \frac{-1}{2} \cdot b + \frac{1}{2} \cdot (b^2 - 4 \cdot c)^{\left(\frac{1}{2}\right)} \\ \frac{-1}{2} \cdot b - \frac{1}{2} \cdot (b^2 - 4 \cdot c)^{\left(\frac{1}{2}\right)} \end{bmatrix} \\
 10) \quad z^2 + b \cdot z + c \text{ solve, } b \rightarrow \frac{-(z^2 + c)}{z} \\
 11) \quad \frac{1}{a^2 - 2 \cdot a - 15} \text{ convert, parfrac, } a \rightarrow \frac{-1}{[8 \cdot (a+3)]} + \frac{1}{[8 \cdot (a-5)]}
 \end{array}$$

Рис. 2.20. Символьные алгебраические действия в MathCAD



На примерах 9, 10 можно видеть роль указания переменной, относительно которой производится вычисление. Для выражения  $z^2+bz+c$  можно указать в качестве переменной  $z$  – тогда оно воспринимается, как квадратный трехчлен с параметрами  $a$ ,  $b$  и оператор "solve" находит его корни в символьной форме. Если же указать для того же самого выражения в качестве переменной величину  $b$ , то оператор "solve" просто разрешает выражение относительно  $b$ .

#### 2.5.4. Символьные операции математического анализа

Примеры выполнения отдельных операций математического анализа показаны на рис. 2.21.

1)  $\lim_{z \rightarrow 0} \frac{\sin(z)}{z} \rightarrow 1$       2)  $F(x,y) := 2 \cdot \sin(x) \cdot \cos(y)$

$\lim_{n \rightarrow \infty} \operatorname{atan}(n^2) \rightarrow \frac{1}{2} \cdot \pi$        $\frac{d}{dx} F(x,y) \rightarrow 2 \cdot \cos(x) \cdot \cos(y)$

$\lim_{x \rightarrow 1} \frac{(x^2 - 1) \cdot (x + 3)}{(x - 1)} \rightarrow 8$        $\frac{d}{dy} F(x,y) \rightarrow -2 \cdot \sin(x) \cdot \sin(y)$

3)  $\int \frac{\cos(x)}{1 + \cos(x) + \sin(x)} dx \rightarrow \frac{-1}{2} \cdot \ln\left(2 + 2 \cdot \tan\left(\frac{1}{2} \cdot x\right)^2\right) + \operatorname{atan}\left(\tan\left(\frac{1}{2} \cdot x\right)\right)$

4)  $\cos(x) \text{ series}, x, 10 \rightarrow 1 - \frac{1}{2} \cdot x^2 + \frac{1}{24} \cdot x^4 - \frac{1}{720} \cdot x^6 + \frac{1}{40320} \cdot x^8$

$\sin(x) \text{ series}, x = \frac{\pi}{4}, 3 \rightarrow \frac{1}{2} \cdot \sqrt{2} + \frac{1}{2} \cdot \sqrt{2} \cdot \left(x - \frac{1}{4} \cdot \pi\right) - \frac{1}{4} \cdot \sqrt{2} \cdot \left(x - \frac{1}{4} \cdot \pi\right)^2$

Рис. 2.21. Символьные операции математического анализа

Пример 4 показывает варианты разложений функции в ряд Тейлора. В первом случае во второй позиции ввода указано просто имя переменной  $x$  и разложение проводится в окрестности точки  $x=0$ . Порядок разложения задан равным 10, поэтому последний сохраненный член разложения имеет порядок  $x^8$ . Следующий член имел бы порядок  $x^{10}$  (функция  $\cos x$  – четная), но

указание порядка разложения 10 говорит о том, что члены этого порядка должны быть отброшены.

Во втором случае вторая позиция ввода содержит равенство  $x=\pi/4$ . В результате разложение проводится в окрестности этой точки до членов 2-го порядка (члены 3-го и более высоких порядков отброшены).

### 2.5.5. Задания для самостоятельной работы.

1. Упростить алгебраические выражения:

$$1. \quad \frac{4x^2 - 5x + 1}{4x - 1} - \frac{x^2 - 1}{1 - x}$$

$$2. \quad \frac{9a^2 - 4}{2 - 3a} - \frac{6a^2 - 5a - 6}{3 - 2a}$$

$$3. \quad \frac{x+1}{x^3 + x^2 + x} : \frac{1}{x^4 - x} - x^2$$

$$4. \quad \frac{(a+b)^3 - (a-b)^3}{2b(3a^2 + b^2)} + 1$$

$$5. \quad \left( \frac{28x}{x^2 - 49} + \frac{x-7}{x+7} \right) \cdot \frac{x}{x+7} - \frac{x}{x-7}$$

$$6. \quad 5 \frac{(a^2 + 5a + 6)(a^2 - 2a + 4)}{(a+3)(a^3 + 8)}$$

$$7. \quad \left( \frac{ab}{a-b} + a \right) \cdot \left( \frac{ab}{a+b} - a \right) : \frac{a^2 b^2}{b^2 - a^2}$$

$$8. \quad \left( \frac{x+5}{(x-9)(x+9)} + \frac{x+7}{(x-9)^2} \right) \cdot \left( \frac{x-9}{x+3} \right)^2 + \frac{7+x}{9+x}$$

$$9. \quad \left( \frac{1}{2-6a} + \frac{1}{27a^3 - 1} : \frac{1+3a}{1+3a+9a^2} \right) \cdot \frac{2+6a}{a}$$

$$10. \quad \frac{8-n^3}{2+n} : \left( 2 + \frac{n^2}{2+n} \right) - \frac{n^2}{n-2} \cdot \frac{4-n^2}{n^2+2n}$$

11.  $\left( \frac{12}{5a^2 + a - 4} - \frac{a + 1}{3(5a - 4)} \right) \cdot \frac{15a - 12}{a + 7}$
12.  $\left( \frac{9}{m^2 - 3m + 9} + \frac{2m}{3 + m} - \frac{m^3 - 15m^2}{m^3 + 27} \right) \cdot \left( m + 3 - \frac{9m}{m + 3} \right) \cdot \frac{1}{m + 3}$
13.  $\left( \frac{16}{x^2 - 4x + 16} + \frac{2x}{x + 4} - \frac{x^3 - 20x^2}{x^3 + 64} \right) \cdot \left( x + 4 - \frac{12x}{x + 4} \right) \cdot \frac{1}{4 + x}$
14.  $\left( \frac{8a^3 + b^3}{4a^2 - b^2} + \frac{1}{b^{-1}} \right) : \frac{a^2}{2a - b}$
15.  $a^{\frac{1}{2}} - \frac{a - a^{-2}}{a^{\frac{1}{2}} - a^{-\frac{1}{2}}} + \frac{1 - a^{-2}}{a^{\frac{1}{2}} + a^{-\frac{1}{2}}} + \frac{2}{a^{\frac{1}{2}}}$
16.  $\frac{x^3 - y^3}{(3x + y)^2 - 8x^2 - 5xy} + \frac{(x + y^2)(x^2 + y) - xy(xy + 1)}{x^2 - xy + y^2}$
17.  $\frac{m^4 - 49}{m^2 + 7} - \frac{m^6 - 343}{m^4 + 7m^2 + 49}$
18.  $\frac{x^{-6} - 64}{16 + 4x^{-2} + x^{-4}} \cdot \frac{1}{4 - 4x^{-1} + x^{-2}} - \frac{4x^2(2x + 1)}{1 - 2x}$
19.  $\left( \frac{9 - x^6}{3 - x^3} - \frac{27 - x^9}{9 - x^6} + \frac{x^6}{3 + x^3} \right)^3$
20.  $\left( \frac{2x + 1}{x + 2} - \frac{4x + 2}{4 - x^2} \right) : \frac{2x + 1}{x - 2} + \frac{2}{x + 2}$
21.  $\frac{x^3 + 4x^2 - 9x - 36}{x^2 + x - 12} \cdot \frac{x^4 - 7x^3 + 12x^2 + 6x - 18}{x^3 - 4x^2 + 6}$
22.  $\left[ \frac{1}{13(x + 5)} + \frac{1}{78(2x - 3)} - \frac{1}{6(2x + 9)} \right] (4x^3 + 32x^2 + 33x - 135)$
23.  $\frac{2}{3} \left( \frac{1}{x - 5} - \frac{1}{x + 4} \right) \frac{x^2 - x - 20}{3}$

$$24. \frac{x^2 - 4x + 3}{x^3 - 27} + \frac{x^2 + 2x + 10}{x^2 + 3x + 9}$$

$$25. \frac{x^2(x-1)}{x^3-1} \cdot \frac{x^2+x+1}{x^4-16} + -\frac{4}{x^4-16}$$

2. Раскрыть скобки и привести подобные в выражениях. Затем произвести обратную операцию – разложить на множители полученный результат. Если в результате получится выражение, отличающееся от исходного, показать их тождественность. Для этого записать разность двух выражений и применить к ней операцию упрощения ("simplify"). Если выражения тождественны, результатом упрощения должен оказаться ноль.

$$1. (x^3 + 7x^2 - 13)(x^5 + x^3 - x^2 + x + 1)$$

$$2. (x^3 + 3x^2 - 5)(x^4 + 3x - 1)$$

$$3. (x^4 - x - 1)(x^5 - 1)$$

$$4. \left(y^3 - \frac{1}{y}\right)\left(y^3 + \frac{1}{y}\right)$$

$$5. (y^5 - 1)^2(y + 3)$$

$$6. (z^3 + z + 1)(z^3 - z + 1)$$

$$7. (z^2 + 1)(z^3 - z + 1)(z^3 - 2)$$

$$8. (p - 5)(p^5 - p - 1)(2p - 3)$$

$$9. (p^3 + 8)(p^2 - 1)(p^2 + 1)$$

$$10. x^5(x^3 - 1)(x + 2)$$

$$11. (x^2 + 1)(2 - x)(x^4 + x^2 + 1)$$

$$12. x^3(x^3 + x + 1)(x^5 - x + 1)$$

$$13. (p - p^2 + 2)(5 - p)(p - 1)$$

14.  $(z-2)(z+5)(z^2+2z+4)$

15.  $(y^2+4y-2)(3y-8)(y+1)$

16.  $(y^3-3)(y^2+2y-1)$

17.  $(y-1)(y^4-1)$

18.  $(x-1)(x^2+4)(x^3-2)$

19.  $x^3(x^2+2x+8)(x-1)$

20.  $x^5(x^2-1)(x^3-6x+25)$

21.  $(y+2)^3(y-1)^2$

22.  $(y+2)(y-1)(y+8)$

23.  $(y-2)^2(y-3)^3$

24.  $(z-25)(z+25^3)$

25.  $(z+5)(z^2-25)^2$

3. Разложить выражение на простые дроби:

1.  $\frac{1}{x^3-2x^2-5x+6}$

2.  $\frac{1}{x^3+2x^2+2x+1}$

3.  $\frac{x-1}{2x^3-x^2-5x+3}$

4.  $\frac{2x+3}{x^3-27}$

5.  $\frac{x^2+x-1}{x^5+x^3-x^2-1}$

6.  $\frac{x+7}{x^4-16}$

$$7. \frac{x}{x^5 - x^3 - x^2 + 1}$$

$$8. \frac{x-11}{x^7 - x^4 + 8x^3 - 8}$$

$$9. \frac{1}{x^7 - x^4 - 16x^3 + 16}$$

$$10. \frac{25x-1}{x^5 - 2x^4 + x^3 - 8x^2 + 16x - 8}$$

$$11. \frac{x^2+1}{x^3 + 2x^2 + 2x + 1}$$

$$12. \frac{x^2-20}{x^3 - 2x^2 - 5x + 6}$$

$$13. \frac{x^2+x-3}{x^7 - x^4 + 8x^3 - 8}$$

$$14. \frac{y-16}{y^3 + 9y^2 + 6y - 16}$$

$$15. \frac{y+16}{y^5 - 13y^4 + 67y^3 - 171y^2 + 216y - 108}$$

$$16. \frac{z+16}{z^4 + 50z^3 - 31250z - 390625}$$

$$17. \frac{1}{z^5 + 5z^4 - 50z^3 - 250z^2 + 625z + 3125}$$

$$18. \frac{z-8}{z^4 + 5z^3 - 8z - 40}$$

$$19. \frac{y+16}{3y^4 + 7y^3 - 34y^2 - 22y + 16}$$

$$20. \frac{2y^2 - y - 1}{y^5 + 2y^4 - y^3 - 3y^2 - 6y + 3}$$

$$21. \frac{2y-5}{y^5 - y^4 - y + 1}$$

$$22. \frac{1}{x^6 - x^5 + 4x^4 - 6x^3 + 2x^2 - 8x + 8}$$

$$23. \frac{5x-7}{x^6 + x^5 + 6x^4 - 8x^3}$$

$$24. \frac{25x}{x^{10} - 7x^8 + 25x^7 + 6x^6 - 25x^5}$$

$$25. \frac{2p-7}{p^7 + 8p^4 - p^3 - 8}$$

4. Выполнить в символьной форме неопределенное интегрирование и дифференцирование указанного выражения по переменной  $x$ . Затем вычислить в символьной форме интеграл от найденной производной. В том случае, если найденный интеграл будет иметь иное выражение, нежели исходное, показать их тождественность, используя операцию упрощения ("simplify").

$$1. \frac{x}{a^2 + x^2}$$

$$2. \frac{x}{(a^2 + x^2)}$$

$$3. \frac{x}{(a + bx)^4}$$

$$4. \frac{1}{x(a^3 - x^3)}$$

$$5. \frac{x}{a^4 + x^4}$$

$$6. \frac{1}{x(a + bx^m)}$$

$$7. \sin^2 x \cdot \cos^3 x$$

8.  $\frac{1}{x(a+bx)^2}$

9.  $\frac{x^3}{(a^2+x^2)^2}$

10.  $\frac{1}{x^4(a^2+x^2)}$

11.  $\frac{1}{x^2(a^2+x^2)^2}$

12.  $\frac{1}{x^2(a^2-x^2)}$

13.  $\frac{1}{x(a+bx^3)}$

14.  $\frac{1}{x^2-1}$

15.  $\frac{x}{1+x^4}$

16.  $\frac{1}{x(a-x^2)}$

17.  $\frac{a+bx}{p+qx}$

18.  $\frac{1}{x(a+bx^3)^2}$

19.  $\frac{1}{x^4(a+x^6)}$

20.  $\frac{x}{\sqrt{a+bx}}$



21.  $\frac{p+qx}{\sqrt{a+bx}}$

22.  $\frac{x}{1-x^4}$

23.  $\frac{p+qx}{\sqrt{(a+bx)^3}}$

24.  $\frac{x^2}{\sqrt{a+bx}}$

25.  $\frac{1}{x(a+x^6)}$

### Глава 3. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ ПАСКАЛЬ

#### 3.1. Общие сведения

Pascal (Паскаль) – язык программирования высокого уровня, созданный швейцарским программистом Никлаусом Виртом. Первоначально Паскаль создавался, как язык для обучения студентов программированию. Будучи учебным языком, Паскаль имеет очень ясную структуру, облегчающую его изучение и написание программ. Эти качества оказались настолько привлекательными, что Паскаль постепенно получил применение в качестве языка практического программирования, хотя и имеет ряд недостатков, обусловленных первоначальной направленностью только на обучение.

Изучение языка Паскаль имеет значение не только в плане овладения понятиями и приемами программирования, но еще и потому, что на его базе построена одна из наиболее популярных сегодня систем программирования Delphi.

#### 3.2. Типы данных в Паскале

Программы, написанные на языке Паскаль, допускают операции с различными типами данных.

Тип данных определяет характер информации, которую данные содержат, и способ кодирования этой информации. Различают следующие основные типы данных:

1. Типы целых чисел, различающиеся диапазоном значений и объемом памяти, требующимся для записи одного числа. Целочисленные типы приведены в таблице 3.1. Наиболее часто используется тип **integer**. Тип **byte** используется для записи порядковых номеров элементов относительно небольших последовательностей, например, индексации массивов.

2. Вещественные числа: тип **real**. Для вещественных чисел существуют также другие типы, различающиеся точностью записи (числом сохраняемых значащих цифр), диапазоном значе-

ний и размером ячейки памяти, но они используются реже. Диапазон значений чисел типа **real** – от  $2.9 \cdot 10^{-29}$  до  $1.7 \cdot 10^{38}$  (по абсолютной величине).

Таблица 3.1

Целочисленные типы языка Pascal.

| Наименование типа | Диапазон значений          | Объем занимаемой памяти, байт |
|-------------------|----------------------------|-------------------------------|
| <b>byte</b>       | 0 .. 255                   | 1                             |
| <b>word</b>       | 0 .. 65535                 | 2                             |
| <b>shortint</b>   | -128 .. 127                | 1                             |
| <b>integer</b>    | -32768 .. 32767            | 2                             |
| <b>longint</b>    | - 2147483648 .. 2147483647 | 4                             |

3. Логический тип **Boolean** – логическое выражение (см. § 1.4). В Паскале значения логических выражений обозначаются словами **True** ("Истинно") и **False** ("Ложно"). В программах логические выражения создаются с помощью равенств, неравенств и логических операций.

4. Символы расширенной таблицы кодов ASCII: тип **char**.

5. Строки (последовательности символов): тип **string**.

6. Записи (тип **record**) – наборы значений различных типов, объединенных под общим именем. Отдельные значения (поля) внутри записи различаются с помощью дополнительных внутренних имен.

7. Массивы (тип **array**) – наборы значений любого другого типа (но только какого-то одного!), объединенных под общим именем и различаемых внутри массива по номерам (индексам). В зависимости от числа индексов массивы могут быть одномерными, двумерными и т.д.

Существуют более сложные типы данных, которые в рамках настоящего пособия не рассматриваются.

Целые и вещественные числа, логические значения и символы относятся к простым типам данных. У таких величин имя

определяет одно значение и одну ячейку памяти компьютера. У массивов, записей и строк одно имя определяет несколько значений (и несколько ячеек памяти); такие типы данных называются сложными (или составными, структурированными).

Типы целых чисел, логические значения и символы относятся к порядковым типам. Это значит, что все возможные значения любого из этих типов можно перенумеровать по порядку. Вещественные числа перенумеровать невозможно.

Данные могут играть в программе роль констант и переменных. Значения констант (**Const**) задаются в начале программы и не могут изменяться в ходе ее выполнения. Переменные (**Var**) могут изменять свое значение по ходу выполнения программы.

### 3.3. Управляющие структуры языка Паскаль

Для обработки данных в языке Паскаль используются различные управляющие структуры:

1. Операторы – отдельные команды языка.

2. Подпрограммы – наборы операторов, имеющие собственное имя и описывающие обобщенно последовательность действий по обработке каких-то условных объектов<sup>1</sup>. В программе подпрограммы обычно применяются несколько раз – в разных местах, или для одинаковой обработки разных реальных объектов. Выделение в составе программы отдельных подпрограмм позволяет также упростить структуру программы и является одним из основных принципов структурного программирования – методики создания больших программ со сложной структурой.

---

<sup>1</sup> В рамках данного пособия не рассматривается *объектно-ориентированное программирование*, в котором *объектом* называется особый элемент программы (объединение набора данных и подпрограмм их обработки), поэтому здесь и далее под "объектом" понимается любой используемый в конкретной программе элемент, обозначенный индивидуальным нестандартным ("пользовательским") именем, например, переменная или константа.

3. Модули – наборы подпрограмм и других элементов (пользовательских типов, данных и др.), записанные по особым правилам и не связанные с конкретной программой. Подпрограммы, включенные в модуль, могут применяться в различных программах, если модуль подключить к программе особой декларацией. В системах программирования на Паскале существуют стандартные модули, содержащие типовые подпрограммы (например, для создания графических элементов). В основном же модули разрабатываются программистами для разбиения особо сложных программ на отдельные более простые части.

### 3.4. Алфавит языка Паскаль

Алфавит (набор используемых символов) языка Паскаль включает в себя:

- заглавные и строчные буквы английского языка (в языке Паскаль регистр букв – заглавные или строчные – не имеет значения, программу можно целиком набрать заглавными или строчными буквами, либо чередовать их произвольным образом);
- десятичные цифры 0, 1, ..., 9;
- Знаки математических операций: +, –, \* (умножение), / (деление) – и символы отношений: =, <, >;
- Знаки препинания: ., , ; : ; символ подчеркивания: \_; апостроф: ';
- Скобки (...), [...], {...};

Буквы русского алфавита ("кириллицы") допустимо использовать только в качестве значений символов в типах **char** и **string**.

Назначение знаков препинания в текстах программ приведено в таблице 3.2.

Таблица 3.2

Знаки препинания языка Pascal.

| ЗНАК | ПРИМЕНЕНИЕ                                                                                                                                    |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| { }  | Выделение комментария                                                                                                                         |
| [ ]  | Выделение списка индексов массива, размера строковой константы                                                                                |
| ( )  | Выделение сложных операндов выражения, списка параметров подпрограммы, значений перечисляемого типа                                           |
| ,    | Апостроф для выделения символьной или строковой константы                                                                                     |
| :=   | Знак присваивания в операторе присваивания (набирается путем нажатия последовательно двух клавиш: и = на клавиатуре)                          |
| ;    | Завершение отдельного предложения программы (оператора, описания и др.)                                                                       |
| :    | Отделение имени переменной от указания ее типа, метки от оператора; также применяется при задании формата числовых данных в процедурах вывода |
| =    | Отделение имени типа от описания типа, константы от ее значения; создание логического выражения (равенства)                                   |
| ,    | Разделение элементов списка                                                                                                                   |
| .    | Обозначение конца программы; <u>отделение целой части вещественного числа от дробной</u> ; выделение имени поля в обращении к записи          |
| \$   | Обозначение директивы компилятора (не рассматривается в рамках данного пособия)                                                               |

Каждый объект, использованный в программе (переменная, константа, тип и др.), должен иметь имя (или идентификатор). Имена большинства объектов могут иметь произвольную длину (до 255 символов), но должны состоять только из букв, цифр и символов подчеркивания (пробелы не допускаются!), причем начинаться обязательно с буквы. Исключением являются имена меток, которые могут состоять просто из цифр. Имена

объектов не должны совпадать со служебными словами Паскаля, приведенными в таблице 3.3.

Таблица 3.3

Служебные слова языка Паскаль.

|                 |                       |                  |                  |              |
|-----------------|-----------------------|------------------|------------------|--------------|
| <b>absolute</b> | <b>end</b>            | <b>inline</b>    | <b>procedure</b> | <b>type</b>  |
| <b>and</b>      | <b>external</b>       | <b>interface</b> | <b>program</b>   | <b>unit</b>  |
| <b>array</b>    | <b>file</b>           | <b>interrupt</b> | <b>record</b>    | <b>until</b> |
| <b>begin</b>    | <b>for</b>            | <b>label</b>     | <b>repeat</b>    | <b>uses</b>  |
| <b>case</b>     | <b>forward</b>        | <b>mod</b>       | <b>set</b>       | <b>var</b>   |
| <b>const</b>    | <b>function</b>       | <b>nil</b>       | <b>shl</b>       | <b>while</b> |
| <b>div</b>      | <b>goto</b>           | <b>not</b>       | <b>shr</b>       | <b>with</b>  |
| <b>do</b>       | <b>if</b>             | <b>of</b>        | <b>string</b>    | <b>xor</b>   |
| <b>downto</b>   | <b>implementation</b> | <b>or</b>        | <b>then</b>      |              |
| <b>else</b>     | <b>in</b>             | <b>packed</b>    | <b>to</b>        |              |

### 3.5. Алгоритм и его описание

Написанию программы предшествует разработка алгоритма. Алгоритмом называется однозначно определенная последовательность конечного числа шагов, необходимых для достижения цели. Существует три основных вида алгоритмических структур:

- Линейная, когда шаги выполняются поочередно один за другим без пропусков или повторений;
- Разветвляющаяся, когда в зависимости от выполнения каких-то условий может выполняться та или иная последовательность шагов;
- Циклическая, когда какая-то последовательность шагов повторяется несколько раз.

Теоретически доказано, что любой сколь угодно сложный алгоритм обработки данных можно построить, комбинируя различным образом эти три структуры.

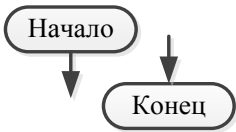
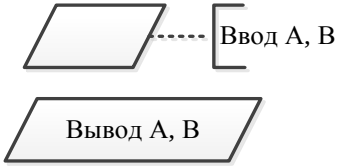
Описать алгоритм можно обычным языком, но такое описание будет длинным и неудобным. Существуют также специ-

альные приемы описания алгоритмов с помощью упрощенных и формализованных словесных конструкций ("псевдокод"). Часто для описания структуры алгоритма применяются блок-схемы. В этом случае отдельные шаги алгоритма изображаются условными графическими элементами ("блоками"), показанными в таблице 3.4. Блоки соединяются в схему горизонтальными или вертикальными линиями, либо ломаными линиями, состоящими из горизонтальных и вертикальных отрезков. Эти линии называются линиями потока.

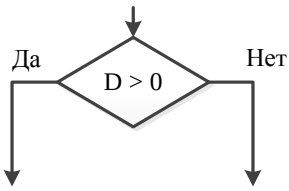
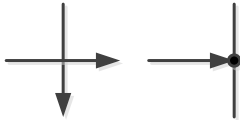
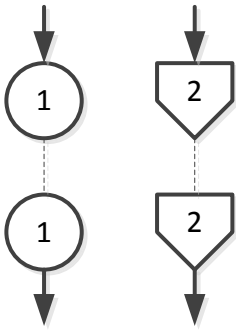
Направления линий потока слева направо и сверху вниз считаются естественными. Противоположные направления – неестественные, их окончание положено фиксировать стрелками.

Таблица 3.4

Основные элементы блок-схем.

|                                                                                     |                                                                                                                                                                                                  |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|    | <p>Элемент, определяющий начало или конец программы.</p>                                                                                                                                         |
|   | <p>Элемент ввода или вывода данных. Указание на ввод или вывод и список ввода/вывода записываются внутри блока или в боковой выноске.</p>                                                        |
|  | <p>Элемент вычисления (присваивания значения). Содержит внутри запись формулы. Может содержать только имя вычисляемой переменной; тогда рядом в боковой выноске может быть записана формула.</p> |



|                                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                               |
|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|   | <p>Элемент разветвления. Выбор дальнейшего хода программы осуществляется в зависимости от выполнения условия, записанного внутри элемента. Если условие истинно, то продолжение вычислительного процесса осуществляется по направлению <b>Да</b>. В противном случае – по направлению <b>Нет</b>.</p>                                                                                                                         |
|   | <p>Пересечение или слияние линий потока. Место слияния фиксируется точкой.</p>                                                                                                                                                                                                                                                                                                                                                |
|  | <p>Из-за пересечения линий потока блок-схема может потерять наглядность. При такой ситуации лучше разрывать линии и пользоваться соединителями, расположенными в конце разрыва линии потока и в начале ее продолжения. Соединители изображаются: внутривстраничный – кружком, междустраничный – пятиугольником. Внутри соединителей, относящихся к одной и той же линии, записываются одинаковые метки: буквы или номера.</p> |

Дополнительные элементы блок-схем будут рассмотрены впоследствии.

При создании блок-схемы рекомендуется размещать блоки в основном вдоль вертикальной линии, которая служит общей осью линейных участков схемы. Элемент начала должен располагаться выше всех прочих элементов, элемент конца – ниже всех. После окончания сложного (разветвленного или цик-

лического) участка завершающая его линия должна вновь выйти на основную ось схемы. Такое расположение упрощает чтение блок-схемы и анализ алгоритма. Одновременно оно отражает последовательность записи операторов в тексте программы. Это облегчает написание программы, реализующей построенный алгоритм.

### 3.6. Общая структура программы на языке Паскаль

Программа на Паскале имеет следующую общую структуру:

```
program <имя_программы>; {заголовок программы}
uses <список модулей>; {сообщение о включении нуж-
                           ных модулей}
...
<Раздел описаний>
...
Begin {начало раздела операторов}
...
<Раздел операторов (исполняемая часть программы)>
...
End_ {конец программы}
```

Угловыми скобками "<...>" здесь и далее выделяются места в программе или ее фрагментах (в том числе в структуре отдельных операторов), которые должны быть заполнены в зависимости от содержания конкретной программы. Внутри скобок помещается пояснение – что именно должно стоять на этом месте. Например, первая строка программы, условно записанная так:

```
program <имя_программы>;
```

– в реальной программе может иметь вид:

```
program MY_FIRST_PROGRAM;
```

Здесь **MY\_FIRST\_PROGRAM** – это имя данной конкретной программы.

Внутри фигурных скобок помещаются комментарии к программе. Комментарий не воспринимается компьютером, как часть программы, и не влияет на ее выполнение.

Три подчеркнутых строки являются обязательными в структуре любой программы. Все остальные строки и элементы программы определяются содержанием решаемой задачи.

### 3.7. Раздел описаний

Раздел описаний (или объявлений) располагается между второй строкой программы (строкой подключения модулей) и ключевым словом **Begin**, открывающим исполняемую часть программы. В разделе описаний должны быть перечислены все использованные в программе имена объектов, не относящихся к стандартным, с указанием вида объекта (метка, константа, пользовательский тип, переменная, подпрограмма), а для переменных – также тип значения каждой переменной. Тип константы определяется автоматически при задании ее значения.

Описание констант и переменных заставляет компьютер произвести распределение ячеек памяти для записи их значений, однако начальные значения переменных при этом в разных системах программирования либо вообще не задаются (остаются неопределенными), либо в ячейки вносятся нулевые значения. Задание начальных значений переменных (их инициализация) должно быть произведено в разделе операторов с помощью присваивания значений, вычисления, ввода с клавиатуры или чтения из отдельного файла данных. В отличие от переменных, значения констант задаются непосредственно при их описании.

Ниже приведены примеры описаний меток и данных (правила описания подпрограмм будут рассмотрены впоследствии):

**Label START, 1,1000;** {описание меток}

**Const E = 2.81828;** {описание вещественной константы}

**KONEC = 'Ура! Конеч!';** {описание константы - строки}  
**N=10;** {описание целочисленной константы}  
**Type MATRIX = array [1 .. 3, 1 .. 5] of real;** {описание типа двумерного массива вещественных чисел}  
**Var i, j, k :integer;** {описание целочисленных переменных}  
**A, B, R :real;** {описание вещественных переменных}  
**TABL : MATRIX;** {описание массива с помощью объявленного выше типа MATRIX}  
**TABLICA: array [1 .. 3, 1 .. 5] of real;** {непосредственное описание массива такой же структуры}  
**RJAD: array [1 .. 10] of integer;** {описание одномерного массива целых чисел}  
**BUKVA: char;** {описание символьной переменной}  
**SLOVO: string [20];** {описание строки из 20 символов}  
**FRAZA: string;** {описание строки максимальной длины – 255 символов}

Под одним стандартным словом, обозначающим вид описываемых объектов (**var, type, const, ...**), могут помещаться несколько описаний объектов разных типов, как это сделано в примерах при описании констант и переменных.

### 3.8. Простые операторы

#### 1. Оператор присваивания имеет структуру:

<имя простой переменной>:=<выражение>;

В результате выполнения в ячейку памяти, соответствующую указанной переменной, заносится вычисленное значение выражения. Выражением может служить имя другой переменной, или формула для вычисления, в том числе с использованием предыдущего значения этой же переменной, например:

**j:=j+1;**

При выполнении такого оператора в ячейку памяти, содержащую значение переменной **j**, заносится новое значение, увеличенное на 1.

Выражение состоит из операндов и операций. Операнды – это отдельные слагаемые, сомножители и т.п., операции – действия над ними.

В Паскале предусмотрены арифметические операции:

|            |                                                              |
|------------|--------------------------------------------------------------|
| +          | - сложение,                                                  |
| –          | - вычитание,                                                 |
| *          | - умножение,                                                 |
| /          | - вещественное деление (с возможностью дробного результата), |
| <b>div</b> | - целочисленное деление (см. § 1.1).                         |
| <b>mod</b> | - определение остатка от целочисленного деления (см. § 1.1). |

Выполнение операций производится в следующем порядке приоритетности:

а) в первую очередь выполняются операции **\***, **/**, **div**, **mod** в том порядке, в каком они записаны в выражении;

б) во вторую очередь выполняются операции сложения и вычитания (также в порядке записи);

Для изменения порядка выполнения операций служат круглые скобки. Выражение, заключенное в скобки, вычисляется полностью прежде, чем будут выполняться операции, стоящие за пределами скобок.

Примеры проявления приоритетности операций приведены в таблице 3.5.

Операция возведения в степень в Паскале отсутствует. Для вычисления значения вида  $X^A$  в основном можно использовать выражение  $e^{A \cdot \ln(X)}$ : **exp(A\*ln(X))**.

Основные математические функции, предусмотренные в Паскале, представлены в таблице 3.6 (здесь **x** – вещественное или целое число или выражение).

При использовании оператора присваивания следует обращать внимание на соответствие типов переменной слева и выражения справа. Не разрешается присваивать значение вещест-

венного выражения переменной целого типа (даже если в результате вычисления этого выражения получится целое число). Вещественный результат, имеющий значение целого числа, можно преобразовать в целый с помощью функции округления **round(...)**.

Таблица 3.5

Примеры записи выражений на языке Паскаль

| Алгебраическое выражение | Запись на языке Паскаль |
|--------------------------|-------------------------|
| $\frac{a+b}{c+d}$        | <b>(a+b)/(c+d)</b>      |
| $\frac{a+b}{c} + d$      | <b>(a+b)/c+d</b>        |
| $a + \frac{b}{c} + d$    | <b>a+b/c+d</b>          |
| $\frac{ab}{cd}$          | <b>a*b/(c*d)</b>        |
| $\frac{ab}{cd}$          | <b>a*b/c/d</b>          |
| $\frac{ab}{c} d$         | <b>a*b/c*d</b>          |

Однако вещественной переменной можно присваивать значение целочисленного выражения – оно автоматически преобразуется в вещественный тип.

Тип числового выражения определяется следующими правилами: выражение принадлежит к целому типу тогда и только тогда, когда выполнены следующие условия:

- все его операнды принадлежат к одному из целых типов;
- в нем использованы следующие арифметические действия: сложение, вычитание, умножение, деление нацело – **div**, определение остатка от деления нацело – **mod** (обычное деле-

ние, обозначаемое косой чертой, всегда дает результат вещественного типа, независимо от получившегося значения!);

- в нем использованы только целочисленные функции. Из перечисленных в таблице 3.6 функций целочисленный результат имеют только функции **round**, **trunc**, **abs** и **sqr** (две последних – только, если их аргумент принадлежит к одному из целых типов).

Таблица 3.6.

Математические функции языка Паскаль

| Функция          | Результат                                                                  | Тип результата             |
|------------------|----------------------------------------------------------------------------|----------------------------|
| <b>Abs(x)</b>    | Абсолютное значение аргумента                                              | Совпадает с типом <b>x</b> |
| <b>Arctan(x)</b> | Арктангенс аргумента                                                       | Вещественный               |
| <b>Cos(x)</b>    | Косинус аргумента                                                          | Вещественный               |
| <b>Exp(x)</b>    | $e^x$                                                                      | Вещественный               |
| <b>Frac(x)</b>   | Дробная часть вещественного числа                                          | Вещественный               |
| <b>Int(x)</b>    | Целая часть вещественного числа                                            | Вещественный               |
| <b>Ln(x)</b>     | Натуральный логарифм                                                       | Вещественный               |
| <b>Pi</b>        | Значение величины $\pi=3.141592653...$                                     | Вещественный               |
| <b>Sin(x)</b>    | Синус аргумента                                                            | Вещественный               |
| <b>Round(x)</b>  | Округление вещественного аргумента до целого                               | Целый                      |
| <b>Trunc(x)</b>  | Выделение целой части вещественного числа путем отбрасывания дробной части | Целый                      |
| <b>Sqr(x)</b>    | Квадрат <b>x</b>                                                           | Совпадает с типом <b>x</b> |
| <b>Sqrt(x)</b>   | Квадратный корень из <b>x</b>                                              | Вещественный               |

Также не разрешается присваивание, если присваиваемое значение может выйти за границы диапазона значений, допустимых для типа величины, стоящей слева в операторе (напри-

мер, не разрешается присваивание величине типа **byte** значения величины типа **integer**).

При нарушении соответствия типов выдается сообщение об ошибке: "**Type mismatch**" ("Несоответствие типов").

### 2. Оператор перехода

**goto** <имя метки>;

Результатом выполнения оператора является переход не к следующему по порядку оператору, а к оператору, перед которым стоит метка, например:

```
.....  
goto 1;  
.....  
1: j:=j+1;  
.....
```

Все метки, имеющиеся в программе, должны быть объявлены в разделе описаний.

Использование операторов перехода, особенно в сложных программах, запутывает структуру программы, усложняет ее понимание и увеличивает вероятность ошибок. Поэтому использование операторов перехода "в чистом виде" нежелательно. В основном они применяются как составная часть условных операторов, однако их применение в любом случае не рекомендуется и считается признаком низкой квалификации программиста. Теоретически доказано, что любой алгоритм можно реализовать с помощью программы, не использующей оператор перехода.

### 3.9. Процедуры ввода-вывода

Составными частями любого алгоритма являются ввод исходных данных с клавиатуры или из файла и вывод полученных результатов на экран монитора, в файл или на печать. В рамках данного пособия рассматриваются только ввод с клавиатуры и вывод на монитор ("консольный" ввод и вывод).

Для этого в Паскале предусмотрен ряд процедур.



### 1. Процедуры ввода:

**Read** (<список ввода>);

**Readln** (<список ввода>);

Список ввода – это перечень имен простых переменных или отдельных элементов массивов. Элементы списка перечисляются через запятую.

Дойдя до обращения к процедуре ввода, компьютер приостанавливает выполнение программы и ожидает ввода данных. Пользователь должен набрать на клавиатуре значения переменных, перечисленных в списке ввода, отделяя их пробелами, затем нажать клавишу ввода (<↵>, <Enter>). Можно также вводить элементы списка по очереди, нажимая клавишу ввода каждый раз после набора одного значения. Так или иначе, выполнение программы продолжится только после того, как будет введено необходимое количество значений.

### 2. Процедуры вывода:

**Write** (<список вывода>);

**Writeln** (<список вывода >);

Список вывода состоит из элементов, перечисляемых через запятую. Элементами списка могут быть:

- имена простых переменных, строк или отдельных элементов массивов (на экран выводятся их значения);
- строковые константы – текстовые пояснения или просто наборы символов, заключенные между двумя апострофами и служащие для придания наглядности выводимым на экран результатам;
- выражения (на экран выводится значение выражения).

Для улучшения вида выводимых числовых значений предусмотрены форматы вывода. Для целых чисел формат вывода – это указание количества позиций на экране, занимаемых данным числом (под позицией понимается место, занимаемое одним символом; на экране в строке помещается 80 позиций). Формат записывается после имени выводимой переменной в списке вывода и отделяется от имени двоеточием, например,

**A:5.** Если выводимое число окажется короче, то лишние позиции перед первой цифрой числа заполняются пробелами, если длиннее – количество позиций автоматически увеличивается.

Для вещественных чисел формат вывода состоит из двух целых чисел, также разделенных двоеточиями. Первое показывает общее число занимаемых выводимым числом позиций (надо помнить, что десятичная точка занимает отдельную позицию!), а второе – количество цифр дробной части числа, например, **A:8:3**. Если дробная часть окажется короче, свободные позиции заполняются незначащими нулями, если длиннее – производится округление. Если указанной ширины поля недостаточно из-за длинной целой части, то поле автоматически удлиняется. Если формат вывода вещественного числа не указан, оно выводится в форме с плавающей точкой.

Следует учесть, что при выводе элементов списка пробелы между ними автоматически не вставляются, их надо вводить в составе строковых фрагментов списка вывода.

Две буквы **'In'** (сокращение от английского **"line"** – "типографская строка") в конце имени процедуры означают в обоих случаях, что после выполнения ввода или вывода курсор на экране автоматически переводится в следующую строку. При использовании процедуры **write** курсор остается в строке, в которой отображены вводимые или выводимые значения. После выполнения процедуры **read** при вводе с клавиатуры курсор также переходит в новую строку из-за нажатия клавиши ввода (разница между процедурами **read** и **readln** существенна только при вводе данных из файлов).

Существуют пустые процедуры ввода-вывода, в которых отсутствуют списки:

**Readln;**

**Writeln;**

Первая процедура вызывает приостановку выполнения программы до момента, когда на клавиатуре будет нажата клавиша ввода. Вторая процедура вызывает перевод курсора на экране в начало следующей строки.

**Пример 3.1.** Вывод на экран значений переменных  $j=12$  и  $A=2.81828$ :

```
writeln ('j=' , j : 5 , 'A=' , A : 8 : 3);
```

Результат:

```
j= 12A= 2.818
```

Между последней цифрой значения  $j$  и началом текста **'A='** не оказалось пробела. Чтобы отделить на экране эти элементы, надо в списке вывода записать строку **' A='**, вставив в нее пробел перед буквой **'A'**.

### 3.10. Структурные операторы

Структурными называют операторы, которые содержат внутри себя другие операторы.

#### 3.10.1. Условный оператор и оператор выбора

Условный оператор позволяет осуществить разветвление вычислительного процесса. Различают полную и неполную форму условного оператора. Полный условный оператор имеет структуру

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

В случае истинности условия выполняется оператор 1, а если условие ложно, то выполняется оператор 2. После этого выполнение программы продолжается в порядке следования дальнейших операторов.

Неполный условный оператор имеет структуру

```
if <условие> then <оператор 1>;
```

Если условие истинно, то выполняется оператор 1, а если условие ложно, то происходит переход к следующему по порядку оператору программы.

В блок-схемах алгоритмов полному и неполному условным операторам соответствуют структуры, показанные на рисунке 3.1.

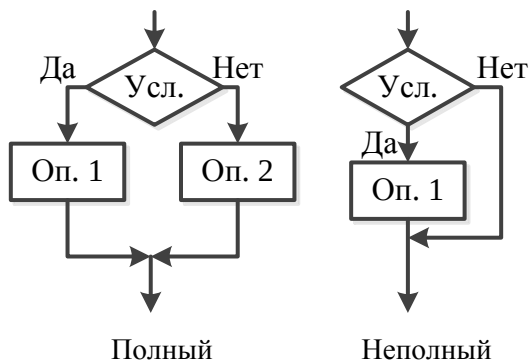


Рис. 3.1. Блок-схемы условных операторов

Любой из внутренних операторов в структуре условного оператора может также быть условным оператором. Это позволяет производить сложные разветвления вычислительного процесса.

**Пример 3.2:** вычислить "кусочное" выражение

$$y = \begin{cases} x & \text{при } x < 0 \\ \sin x & \text{при } 0 \leq x \leq \pi \\ \pi - x & \text{при } x > \pi \end{cases}$$

Отметим, что второе условие является сложным и распадается на два простых:  $x \geq 0$  и  $x \leq \pi$ . Чтобы избежать записи сложного условия, можно использовать логическое правило исключенного третьего (см. § 1.4).

Очевидно, что любое значение  $x$  либо отрицательно, либо неотрицательно. Если поставить условие  $x < 0$ , то его ложность автоматически означает, что  $x \geq 0$ . В этом случае значение  $y$  должно вычисляться либо по второй формуле, либо по третьей, других возможностей уже не существует. Поэтому можно выбрать любое из оставшихся условий – при его истинности следует использовать соответствующую ему формулу, а при ложности автоматически применить оставшуюся третью формулу, уже не ставя для нее никаких дополнительных условий. В качестве такого условия можно взять простое условие  $x \geq \pi$ , записать которое легче, нежели сложное  $0 \leq x \leq \pi$ . Соответствующий

условный оператор имеет структуру, показанную на рисунке 3.2.

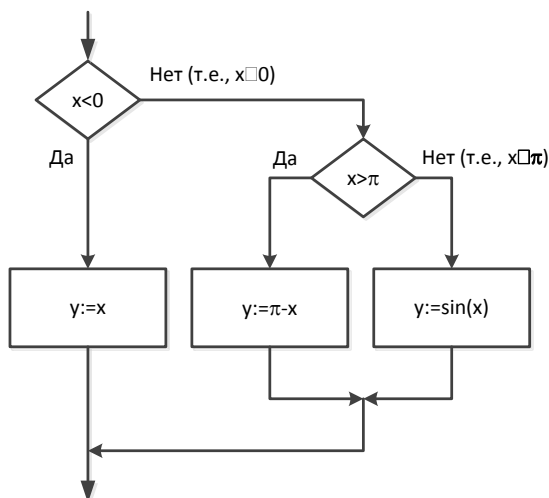


Рис. 3.2. Алгоритм вычисления кусочного выражения  
Запись оператора имеет вид:

```
if x<0
then
  y:=x
else
  if x>Pi
  then
    y:=Pi - x
  else
    y:=sin(x);
```

Из этого примера видно, что простые условия в Паскале имеют вид математических отношений (равенств или неравенств). Допускается постановка условий в виде отношения для символьных переменных, при этом фактически сравниваются

между собой их номера в таблице кодов ASCII; в частности, неравенства

**'A'<'F',       'y'<'z',       'A'<'a'.**

имеют значение **True**. Последнее неравенство истинно, так как в таблице кодов ASCII заглавные буквы стоят раньше строчных.

В качестве простого условия можно также использовать значение переменной логического типа.

Даже пользуясь правилом исключенного третьего, не всегда удастся избежать использования сложных условий (логических выражений). Для их создания в Паскале предусмотрены следующие логические операции (см. также § 1.4):

- Дизъюнкция записывается служебным словом **or** ("или"):

**if (<условие 1> or <условие 2>) then       ...**

Сложное условие считается истинным, если истинно хотя бы одно из условий: 1 или 2.

Блок-схема дизъюнкции показана на рисунке 3.3.

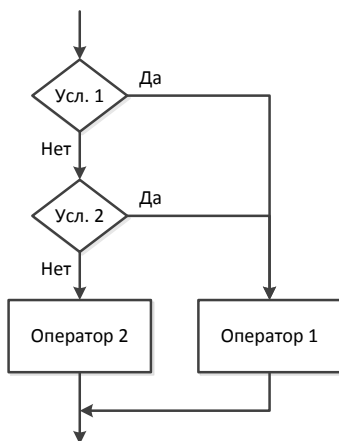


Рис. 3.3. Схема дизъюнкции (логического сложения)

- Конъюнкция записывается служебным словом **and** ("и"):

**if (<условие 1> and (<условие 2>) then     ...**

Сложное условие считается истинным, только если оба условия: 1 и 2 – одновременно являются истинными.

Блок-схема конъюнкции показана на рисунке 3.4.

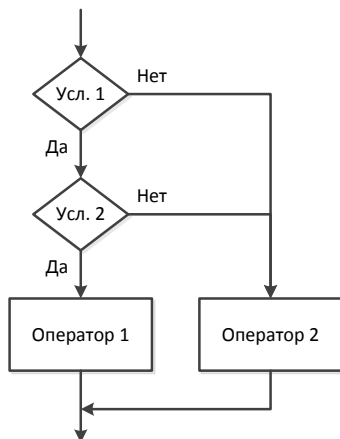


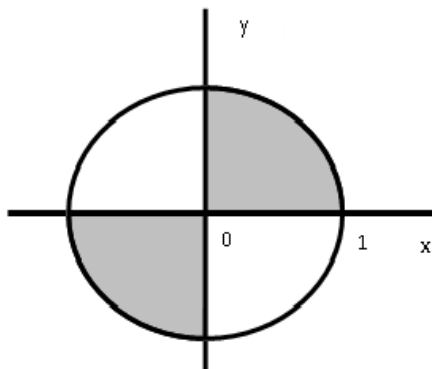
Рис. 3.4. Схема конъюнкции (логического умножения)

- Отрицание записывается служебным словом **not** ("не"), стоящим перед условием. Оно меняет значение условия на противоположное.

Еще одна логическая операция – **xor** (строгая дизъюнкция) используется реже и в данном пособии ее применение не рассматривается.

В логических выражениях Паскаля можно также использовать скобки, вынося за них общие логические множители или слагаемые в соответствии с законами дистрибутивности (см. п. 1.4.3)

**Пример 3.3.** Даны координаты точки на плоскости X и Y. Записать условие, определяющее, попала ли точка в одну из двух областей, выделенных заливкой:



Вариант 1:

**$(X * X + Y * Y \leq 1) \text{ and } (X \geq 0) \text{ and } (Y \geq 0)$**

**or**

**$(X * X + Y * Y \leq 1) \text{ and } (X \leq 0) \text{ and } (Y \leq 0)$**

Вариант 2 (общее условие-множитель вынесено за скобки):

**$(X * X + Y * Y \leq 1)$**

**and**

**$((X \geq 0) \text{ and } (Y \geq 0) \text{ or } (X \leq 0) \text{ and } (Y \leq 0))$**

Второй вариант записывается короче.

Более сложные разветвления вычислительного процесса организуются с помощью оператора выбора. Его форма:

**case <выражение> of**

**<значение 1> : <оператор 1>;**

**< значение 2> : <оператор 2>;**

**.....**

**< значение N-1> : <оператор N-1>**

**else**

**<оператор N>;**



Выражение в заголовке оператора, по которому делается выбор, должно быть целым или символьным. Значения, соответствующие отдельным операторам (**значение 1**, **значение 2**, ..., по своему типу должны совпадать с типом этого выражения. При исполнении оператора сначала вычисляется значение выражения, записанного в заголовке. Если оно совпадает с одним из перечисленных значений, то выполняется соответствующий этому значению оператор. Если значение выражения не равно ни одному из перечисленных, то выполняется оператор N.

### 3.10.2. Составной оператор (блок)

Часто бывает необходимо при выполнении какого-либо структурного оператора исполнить внутри него не один, а несколько отдельных операторов. В этом случае простые операторы объединяются в составной оператор или блок. В начале блока ставится служебное слово **begin** ("начало"), а в конце слово **end** ("конец") и точка с запятой:

```
begin  
  y:=sin(x);  
  x:=x+h;  
end;
```

Слова **begin** и **end** называются операторными скобками. Как и при использовании обычных скобок в арифметических выражениях, количество слов **begin** в программе должно строго соответствовать количеству слов **end**, и отдельные блоки не должны пересекаться (но могут помещаться один внутри другого по принципу "матрешки").

С точки зрения правил языка Паскаль весь составной оператор считается одним оператором.

### 3.10.3. Операторы цикла

Значительное количество процессов обработки информации требуют многократного повторения однотипных процессов. Для выполнения таких действий применяются операторы цикла. Общая схема работы оператора цикла:

1) Оператор, или последовательность операторов (называемые телом цикла), повторяется многократно;

2) При выполнении определенного условия повторение тела цикла прекращается и управление передается следующему по порядку оператору программы.

При программировании циклических процессов важно определить как содержание тела цикла, так и условие выхода из него. В зависимости от формирования выхода из цикла в Паскале применяются 3 типа циклов.

А) **Цикл с предусловием** (проверка условия выполнения цикла перед входом в тело цикла):

**while <условие> do <тело цикла>;**  
("Пока")            ("выполнять")

Блок-схема цикла показана на рисунке 3.5. Из нее видно, что условие проверяется перед входом в цикл и тело цикла выполняется, только если условие удовлетворяется. Если перед началом цикла условие не выполняется, то тело цикла не будет исполнено ни разу.

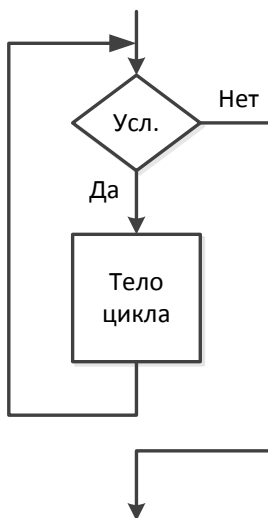


Рис. 3.5. Цикл с предусловием

Если тело цикла состоит из нескольких операторов, их необходимо объединить в блок.

Важнейшим требованием при использовании описанного цикла является необходимость изменения значения входного условия в ходе исполнения тела цикла. В противном случае цикл будет повторяться до бесконечности (такая ситуация у программистов называется "зацикливанием"). Для предотвращения зацикливания условие должно включать в себя какую-то управляющую переменную, значение которой в теле цикла обязательно должно изменяться при каждом повторении цикла.

Б) **Цикл с постусловием** (проверка условия прекращения повторений в конце тела цикла).

**repeat <операторы тела цикла> until <условие>;**  
("повторять") ("до тех пор, пока")

Блок-схема цикла показана на рисунке 3.6.

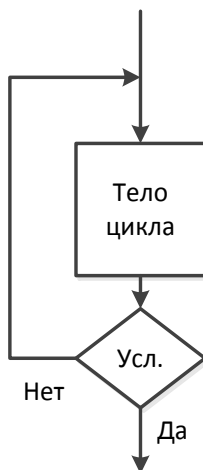


Рис. 3.6. Цикл с постусловием

Тело такого цикла всегда выполняется как минимум один раз, а условие проверяется после выполнения тела цикла. При его выполнении происходит выход из цикла. Как и в предыдущем типе цикла, исполнение тела цикла должно влиять на усло-

вие выхода за счет изменения значения управляющей переменной.

Тело цикла с постусловием может состоять из нескольких операторов, но объединять их в блок словами **begin** и **end** не обязательно, так как служебные слова **repeat** и **until** сами по себе играют роль операторных скобок.

В) **Счетный цикл (цикл со счетчиком)**. В отличие от двух вышеописанных, цикл со счетчиком выполняется строго определенное число раз. Его вид:

```
for <i>:=<S> to <F> do <тело цикла>;  
("для")      ("к")  ("выполнять")
```

Счетчик **i** – это переменная целого типа, **"S"** и **"F"** – начальное и конечное значения счетчика. Это могут быть числа, константы, переменные или выражения любого целочисленного типа; конечное значение (**F**) должно быть больше начальное (**S**).

Ход выполнения счетного цикла можно проиллюстрировать с помощью эквивалентного ему цикла с предусловием (рис. 3.7).

Вначале счетчику присваивается его начальное значение и выполняется тело цикла, затем счетчик увеличивается на 1 (это действие производится автоматически и не записывается в тексте программы!) и тело цикла повторяется. Далее значение счетчика увеличивается на 1 после каждого повторения тела цикла, пока после очередного повторения счетчик не станет больше заданного конечного значения. Последний раз тело цикла выполняется при значении счетчика, равном конечному.

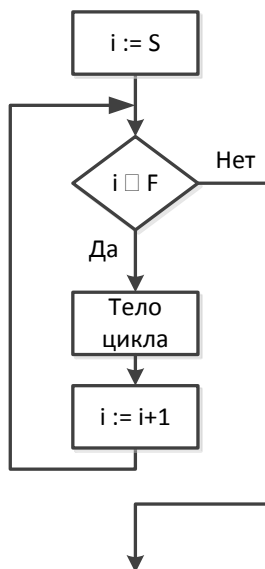


Рис. 3.7. Схема выполнения счетного цикла

Оператор цикла со счетчиком изображается на блок-схемах особой фигурой (рис. 3.8).

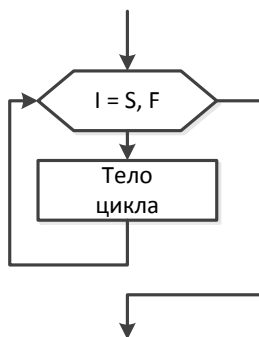


Рис. 3.8. Изображение счетного цикла на блок-схемах

Тело цикла должно быть одиночным оператором или блоком. Важное требование – выполнение тела цикла не должно влиять на значение счетчика, так как управление им происходит автоматически.

Разновидностью счетного цикла является цикл вида

```
for <i>:=<S> downto <F> do <тело цикла>;  
      ("вниз к")
```

Он отличается тем, что конечное значение счетчика должно быть меньше начального и после выполнения тела цикла счетчик каждый раз уменьшается на единицу, так что цикл проводится как бы в обратном порядке.

### 3.11. Подпрограммы. Процедуры и функции.

Подпрограмма – это часть программы, которая оформлена особым образом и может быть исполнена в программе несколько раз для одинаковой обработки разных объектов. Подпрограммы в языке Pascal делятся на процедуры и функции. Процедуры служат для выполнения различных действий, не всегда связанных с какими-то вычислениями, а функции всегда вычисляют какие-то значения. Команда на выполнение подпрограммы называется обращением к ней, или вызовом подпрограммы. Обращение к процедуре всегда является отдельным оператором программы, тогда как обращение к функции производится в ходе вычисления какого-то выражения и значение функции служит операндом при вычислении.

Выше уже упоминались стандартные процедуры и функции Паскаля. Рассмотрим основные правила работы с пользовательскими (написанными самим программистом) подпрограммами.

Каждая подпрограмма должна быть описана в разделе описаний программы. В описании подпрограммы указываются объекты, которые в ней обрабатываются или используются. На стадии описания эти объекты обозначаются условными именами, которые называются формальными параметрами подпрограммы. При обращении к подпрограмме формальные параметры заменяются фактическими – именами реальных объектов программы. Типы формальных и фактических параметров должны совпадать, тогда как совпадение их имен нежелательно (хотя и не запрещено).

Если в подпрограмме используются вспомогательные переменные, которые нужны только при ее выполнении и не используются в основной программе, они описываются непосредственно в теле подпрограммы и называются ее локальными параметрами. Кроме того, правилами Паскаля разрешается (хотя и не рекомендуется) без дополнительного описания использовать в теле подпрограммы любые величины, описанные в основной программе – тогда они называются глобальными параметрами подпрограммы.

Структура описания подпрограммы-процедуры похожа на структуру программы:

```
procedure <имя> (<список формальных параметров>);  
  <раздел описаний локальных параметров>  
begin  
  <раздел операторов процедуры>  
end;
```

Список формальных параметров напоминает раздел описаний. Он состоит из перечня имен формальных параметров, разделенных на группы. Группы отделяются друг от друга точкой с запятой. Каждая группа – это перечень имен формальных параметров одного типа, перечисленных через запятую, за которым после двоеточия следует указание типа этих параметров. Перед перечнем имен в группе может стоять ключевое слово **const** или **var**. Наличие или отсутствие такого ключевого слова указывает на один из трех видов формальных параметров:

а) параметр-константа (перед списком имен в группе стоит слово **const**). При исполнении процедуры значение фактического параметра, соответствующего формальному параметру-константе, не может изменяться (при этом в самой программе фактический параметр может быть переменной величиной);

б) параметр-значение (перед списком имен в группе не стоит никаких ключевых слов). В этом случае перед исполнением процедуры начальное значение соответствующего фактического параметра заносится во вспомогательную ячейку памяти,

а в конце выполнения процедуры это значение вновь присваивается фактическому параметру. Таким образом, в ходе выполнения процедуры значение фактического параметра может изменяться, но это изменение будет "потеряно" после окончания выполнения процедуры.

в) параметр-переменная (перед списком имен в группе стоит слово **var**). Значение соответствующего фактического параметра в ходе выполнения процедуры может изменяться и это изменение сохранится после завершения процедуры.

Параметрами-переменными обычно являются результирующие величины подпрограммы, тогда как исходные данные для работы подпрограммы являются константами или значениями.

Для обращения к процедуре в программе записывается ее имя с указанием в скобках списка фактических параметров. При этом в качестве параметров-констант и переменных могут использоваться только имена (для параметров-переменных – только имена описанных в программе переменных, для констант – имена как констант, так и переменных). В качестве фактических параметров-значений могут использоваться не только имена, но и выражения.

Обращение к процедуре изображается на блок-схеме алгоритма с помощью блока, изображенного на рис. 3.9.

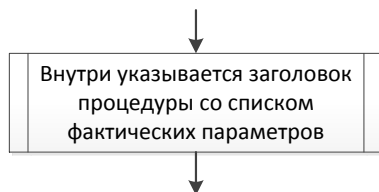


Рис. 3.9. Изображение обращения к процедуре

Для примера запишем процедуру, осуществляющую обмен значений между двумя переменными (ячейками памяти):

```
procedure OBMEN(var AF,BF : real);  
  var BUF :real;  
  begin
```



```
    BUF := AF;  
    AF := BF;  
    BF := BUF;  
end;
```

Процедура имеет два формальных параметра-переменных **AF** и **BF**. Для временного хранения одного из значений использована промежуточная (буферная) переменная **BUF**, которая является локальным параметром процедуры. С помощью этой переменной значения переменных **AF** и **BF** переставляются "по кругу".

Описание подпрограммы-функции сходно с описанием процедуры, но имеет два отличия. Первое – в структуре заголовка. У функции он имеет вид

```
function <имя> (<список формальных параметров>) :  
<тип значения функции>;
```

Второе отличие заключается в том, что последний оператор в разделе операторов функции имеет вид

```
<имя функции > := <вычисленное значение функции>;
```

Формальные параметры функций, как правило, являются параметрами-значениями. Это позволяет вычислять функции от аргументов-выражений.

В качестве примера рассмотрим пользовательскую функцию, вычисляющую степенное выражение  $x^y$ . Как уже было сказано, в большинстве случаев в Паскале для этого можно использовать конструкцию вида **exp(y\*ln(x))**, однако она непригодна, если  $x \leq 0$ , т.к. при таких значениях  $x$  не определено значение **ln(x)**. Между тем, при  $x=0$  степенное выражение существует – оно просто равно нулю. Кроме того, при  $x < 0$  степенное выражение также может быть вычислено, если показатель степени  $y$  имеет целое значение:

$$x^y = \begin{cases} \exp(y \cdot \ln(|x|)) & \text{если } y - \text{четное число} \\ -\exp(y \cdot \ln(|x|)) & \text{если } y - \text{нечетное число} \end{cases}$$

Тогда общая схема функции для вычисления степенного выражения (в тех случаях, когда оно вообще может быть вычислено) выглядит так, как показано на рисунке 3.10.

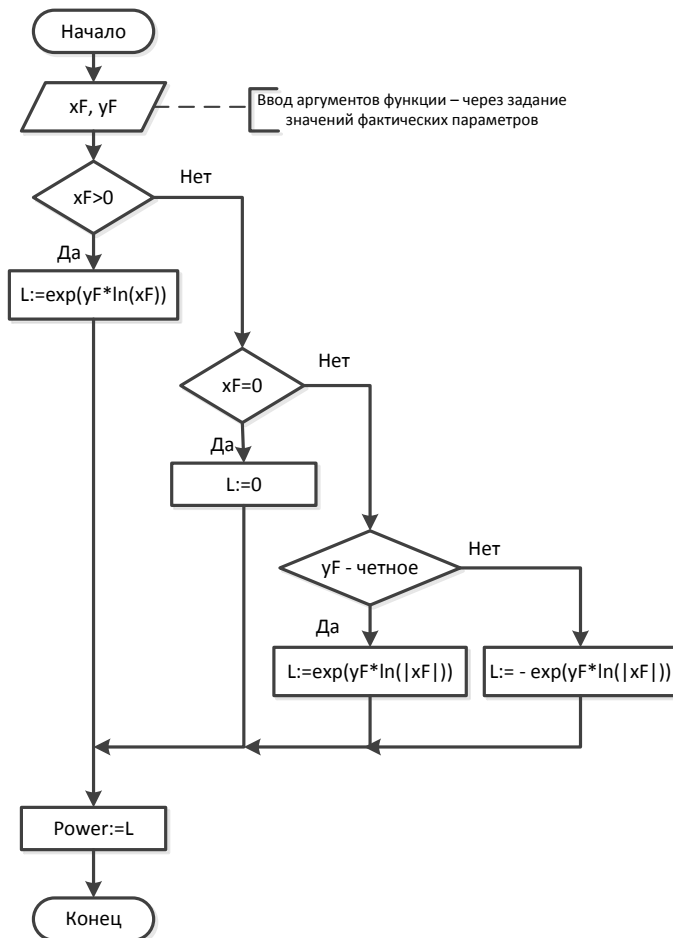


Рис. 3.9. Схема подпрограммы-функции для вычисления степенного выражения.

На этом рисунке **Power** – имя функции, **xF**, **yF** – формальные параметры, соответствующие основанию и показателю степени в вычисляемом выражении, **L** – локальный параметр, используемый в теле функции для хранения вычисленного выражения.

Поскольку при положительных **x**, **y** степенное выражение вычисляется для любых вещественных значений этих переменных, аргументы функции (фактические параметры) в общем случае должны быть вещественными. Поэтому формальные параметры **xF**, **yF** также должны быть вещественными.

В то же время при **x<0** показатель степени **y** должен быть целым и в ходе вычисления производится его проверка на четность. Для преобразования вещественного числа в целое следует использовать функцию округления **round(...)** (см. табл. 3.6), а для проверки на четность – операцию вычисления остатка от деления **mod** (число является четным, если остаток от его деления на 2 равен нулю).

Текст функции:

```
Power(xF, yF:real):real;  
var L: real;  
Begin  
  if xF>0 then  
    L :=exp(yF*ln(xF))  
  else  
    if xF=0 then  
      yF := 0  
    else  
      if round(yF) mod 2 =0 then  
        L :=exp(yF*ln(abs(xF)))  
      else  
        L := - exp(yF*ln(abs(xF)));  
      Power := L;  
End;
```

Отметим, что в теле функции не предусмотрена проверка – можно ли вообще вычислить степенное выражение. Такую проверку следует предусмотреть в тексте основной программы перед вычислением выражения, содержащего вызов данной функции.

### 3.12. Системы программирования на языке Паскаль

Долгое время для программирования на языке Паскаль использовались различные версии системы Turbo Pascal, разработанной фирмой Borland (США). Последней из версий была версия Borland Pascal 7.0 (BP 7.0). Она еще встречается на практике, хотя ее поддержка разработчиком давно прекращена, а под управлением ОС Windows Vista и более поздних версий ОС Windows эта система вообще не работает.

По аналогии с BP 7.0 была разработана свободно распространяемая система Free Pascal (она может работать под управлением любых версий ОС Windows). Интерфейс обеих систем полностью идентичен.

В 2003 г. в Южном Федеральном университете России была создана система программирования Pascal ABC. Она имеет полноценный графический интерфейс, основанный на тех же принципах, что и пользовательский интерфейс других приложений Windows.

Рассмотрим кратко системы программирования BP 7.0 / Free Pascal.

После запуска системы на экране возникает окно, в верхней части которого находится меню. Для раскрытия того или иного пункта меню надо щелкнуть его мышью.

Для начала работы над новой программой надо раскрыть пункт меню File и выбрать в нем щелчком мыши команду **New** ("Новый"). Окно принимает вид, показанный на рисунке 3.11.

Это так называемое окно редактирования программы. Оно имеет синий цвет, на котором символами разных цветов отображается набираемый с помощью клавиатуры текст программы. В верхней части окна видно имя файла. По умолчанию созданный файл называется **noname00.pas**.

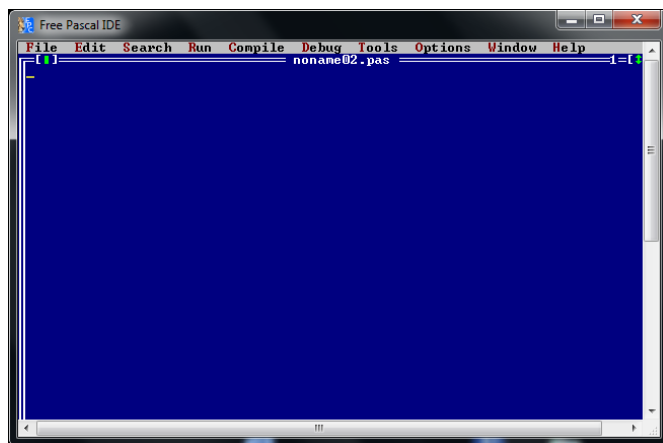


Рис. 3.11. Окно редактирования системы Free Pascal

Можно открыть несколько окон редактирования, которые располагаются одно поверх другого. Для изменения размеров активного окна надо перетащить мышью его нижний правый угол. Перетаскивание любой точки верхней границы окна приводит к перемещению окна целиком. Закрыть окно можно, щелкнув мышью прямоугольный зеленый маркер в левом конце верхней границы окна.

Другие команды раздела File:

- **Open** ("Открыть") – команда открытия уже имеющегося файла. При ее выборе возникает окно диалога (рис. 3.12).

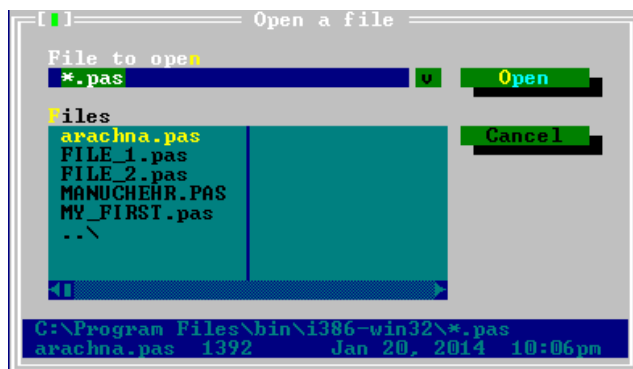


Рис. 3.12. Открытие файла

В окне ввода "File to open" можно ввести с клавиатуры имя файла, содержащего текст программы (если это имя известно). Если имя файла не известно, можно просто найти этот файл в списке, содержащемся в окне "Files" и затем щелкнуть клавишу [Open]. Для отмены команды и закрытия окна служит клавиша [Cancel].

- **Save** ("Сохранить") – команда сохранения файла. При первичном сохранении созданного файла открывается окно диалога (рис. 3.13). В окне ввода "Name" надо набрать имя файла и щелкнуть [Ok]. Список имеющихся файлов в окне "Files" помогает не дать по ошибке новому файлу уже существующее имя.

Система BP 7.0 работает с именами файлов, длина которых не превышает 8 символов (это свойство "унаследовано" от ранее применявшейся операционной системы MS DOS). Система Free Pascal поддерживает длинные имена, но злоупотреблять этим не стоит – в окне "Files" длинные имена целиком не помещаются.

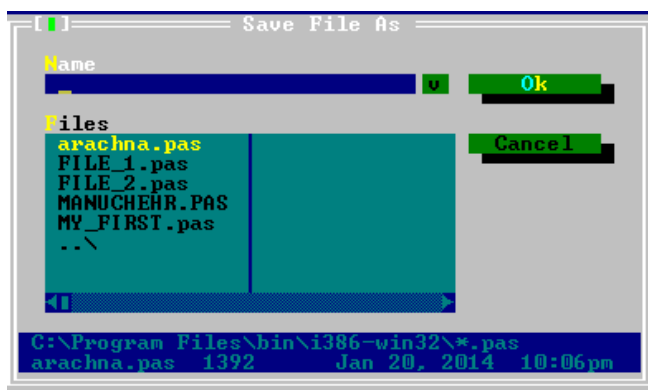


Рис. 3.13. Сохранение файла программы

При последующем выполнении команды Save новая версия файла сохраняется под тем же именем и окно диалога уже не возникает.

- **Save as...** ("Сохранить как...") – сохранение файла под новым именем. По этой команде всякий раз возникает окно диалога, показанное на рисунке 3.13.

- **Exit** – выход из системы. Если перед выходом в программу были внесены изменения и не была выполнена команда

сохранения файла, система предупреждает пользователя об этом. В ВР 7.0 возникает окно диалога с предупреждением (на английском языке), в системе Free Pascal – появляется окно сохранения (рис. 3.13; если сделанные изменения не требуется сохранять, в окне надо просто щелкнуть [Cancel]).

После набора текста программы ее запускают на выполнение командой "Run" из одноименного раздела меню. Для запуска также можно воспользоваться "горячей" комбинацией клавиш <Ctrl> + <F9>. Система производит компиляцию программы ("перевод" на язык машинных команд) и начинает ее выполнение.

Если при компиляции система обнаруживает в программе синтаксические ошибки (нарушения правил записи конструкций языка), программа не выполняется и появляется сообщение об ошибке ("**Error...**") с указанием кодового номера и содержания ошибки. Кроме синтаксических, существуют ошибки процесса выполнения ("**Runtime error...**"), возникающие уже в ходе работы программы. Перечень некоторых часто встречающихся в учебных программах ошибок и типичных причин их возникновения приведен в таблице 3.7 (разумеется, причины ошибок могут и отличаться от приведенных в таблице).

Таблица 3.7

Основные ошибки в программах (для систем ВР 7.0 / Free Pascal)

Общее замечание: если возникает сообщение об ошибке "**... expected**" (Ожидается ...), не следует сразу вставлять в указанное место якобы ожидаемый символ или слово. Необходимо проанализировать всю конструкцию (оператор, выражение, ...). Например, указание на отсутствие закрывающей скобки ("**)**" **expected** ) может означать как то, что программист забыл поставить эту скобку, так и то, что в предшествующей части выражения была поставлена лишняя открывающая скобка.

| Код                                | Сообщение об ошибке | Типичные причины возникновения |
|------------------------------------|---------------------|--------------------------------|
| I. Синтаксические и другие ошибки, |                     |                                |

| Код                                     | Сообщение об ошибке                                                   | Типичные причины возникновения                                                                                                                                                                                                                                          |
|-----------------------------------------|-----------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>выявляемые в процессе компиляции</b> |                                                                       |                                                                                                                                                                                                                                                                         |
| 3                                       | Unknown identifier<br>(Неизвестный идентификатор)                     | 1. В программе использовано имя объекта, не объявленное в разделе описаний<br>2. Опечатка в служебном слове<br>3. Забыли поставить символ умножения между двумя сомножителями в выражении (запись <b>ab</b> вместо <b>a*b</b> воспринимается, как имя новой переменной) |
| 4                                       | Duplicate identifier<br>(Двойной идентификатор)                       | Попытка дважды описать одно и то же имя                                                                                                                                                                                                                                 |
| 5                                       | Syntax error<br>(Синтаксическая ошибка)                               | Причины разнообразны (в частности, наличие символа ; внутри оператора)                                                                                                                                                                                                  |
| 8                                       | String constant exceeds line<br>(Строка превышает допустимые размеры) | Неверная расстановка апострофов в ходе записи списка вывода процедуры <b>write(...)</b>                                                                                                                                                                                 |
| 10                                      | Unexpected end of file<br>(Не найден конец файла)                     | 1. Не закрыт комментарий<br>2. Попытка скомпилировать программу, текст которой набран не до конца<br>3. Ошибка в организации блоков. В программе есть слово <b>begin</b> , к которому не имеется парного слова <b>end</b> .<br>4. Не поставлена точка в конце программы |
| 20                                      | Variable identifier expected<br>(Не найдена переменная)               | Нарушение правил записи оператора                                                                                                                                                                                                                                       |



| Код | Сообщение об ошибке                                                                                  | Типичные причины возникновения                                                                                                                                                                    |
|-----|------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 21  | Type mismatch (Несоответствие типов)                                                                 | Несоответствие типов переменной и выражения в операторе присваивания                                                                                                                              |
| 36  | BEGIN expected (Нужен BEGIN)                                                                         | 1. Сразу после описания (переменной, подпрограммы...) стоит оператор (т.е., не поставлено слово <b>Begin</b> в начале раздела операторов)<br>2. Ошибка в организации блоков                       |
| 37  | END expected (Нужен END)                                                                             | Ошибка в организации блоков                                                                                                                                                                       |
| 40  | Boolean expression expected (Нужно логическое выражение)                                             | Вместо условия в условном операторе или операторе цикла стоит числовая или символьная переменная                                                                                                  |
| 41  | Operand types do not match operator (Типы операндов не соответствуют оператору)                      | 1. Применение логических операций к числовым или символьным переменным, арифметических операций – к символьным и т.п.<br>2. Применение операций <b>div</b> , <b>mod</b> к вещественным переменным |
| 42  | Error in expression (Ошибка в выражении)                                                             | Забыли поставить символ арифметической операции (часто пользователи забывают ставить символ умножения)                                                                                            |
| 62  | Division by zero (Деление на ноль)                                                                   | Имеется знак деления, но не записан знаменатель выражения                                                                                                                                         |
| 64  | Cannot Read or Write variable s of this type (Невозможно ввести или вывести переменные данного типа) | 1. В списке ввода или вывода указано имя массива или записи без указания индекса элемента или имени поля<br>2. В списке ввода указано имя константы                                               |

| Код | Сообщение об ошибке                                                       | Типичные причины возникновения                                                                                                                                                               |
|-----|---------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 85  | ";" expected<br>(Отсутствует символ ";")                                  | Символ ";" забыли поставить в конце предыдущего оператора                                                                                                                                    |
| 88  | "(" expected<br>(Отсутствует символ "(" )                                 | Ошибка в расстановке скобок в арифметическом выражении                                                                                                                                       |
| 89  | ")" expected<br>(Отсутствует символ ")" )                                 | 1. Ошибка в расстановке скобок в арифметическом выражении<br>2. В процедуре вывода данных к целому числу применен формат вывода вещественного числа                                          |
| 91  | ":=" expected<br>(Отсутствует символ ":=" )                               | 1. В операторе присваивания вместо ":=" использован знак равенства<br>2. В имени переменной имеется пробел                                                                                   |
| 92  | "[" expected<br>(Отсутствует символ "[" )                                 | Индекс элемента массива записан в круглых скобках вместо квадратных                                                                                                                          |
| 93  | "]" expected<br>(Отсутствует символ "]" )                                 | 1. Индекс элемента массива записан в круглых скобках вместо квадратных<br>2. Для элемента одномерного массива записаны два индекса                                                           |
| 94  | ". " expected<br>(Отсутствует символ ". " )                               | Ошибка в организации блоков. Записанное слово <b>end</b> оказалось парным к слову <b>Begin</b> , открывающему раздел операторов (было пропущено слово <b>begin</b> в начале какого-то блока) |
| 97  | Invalid FOR control variable<br>(Неверная управляющая переменная для FOR) | В заголовке счетного цикла использованы величины вещественного типа (должны быть только целые)                                                                                               |

| Код                                                    | Сообщение об ошибке                               | Типичные причины возникновения                                                                                                                                                                                                                                            |
|--------------------------------------------------------|---------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 113                                                    | Error in statement<br>(Ошибка в предложении)      | 1. Опечатка при записи оператора (например, в служебном слове)<br>2. Поставлен символ ";" перед словом <b>else</b> в полном условном операторе                                                                                                                            |
| 121                                                    | Invalid qualifier<br>(Неверный определитель)      | 1. Имя массива записано без указания индекса элемента<br>2. Число индексов у элемента массива не соответствует объявленной размерности массива<br>3. К имени простой переменной приписан индекс                                                                           |
| <b>II. Ошибки процесса выполнения (Runtime errors)</b> |                                                   |                                                                                                                                                                                                                                                                           |
| 106                                                    | Invalid numeric format<br>(Неверный формат числа) | 1. При вводе числа с клавиатуры введен символ, не используемый при записи чисел (буква и т.п.)<br>2. Дробная часть числа при вводе отделена запятой, а не точкой<br>3. Вводимые значения разделены запятыми, а не пробелами                                               |
| 200                                                    | Division by zero<br>(Деление на ноль)             | Переменная, стоящая в знаменателе, не была инициализирована                                                                                                                                                                                                               |
| 201                                                    | Range check error<br>(Ошибка проверки границы)    | 1. Попытка присвоить величине значение, выходящее за пределы диапазона значений данного типа<br>2. Попытка использовать элемент массива с индексом, выходящим за пределы объявленного диапазона значений индекса (например, найти 11-й элемент в массиве из 10 элементов) |

| Код | Сообщение об ошибке                                                            | Типичные причины возникновения                                                                                                                                     |
|-----|--------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 205 | Floating point overflow<br>(Переполнение в операции с плавающей точкой)        | Получилось слишком большое вещественное число                                                                                                                      |
| 207 | Invalid floating point operation<br>(Недопустимая операция с плавающей точкой) | Вычисление выражения вне области его определения, в том числе:<br>1. Вычисление корня из отрицательного числа<br>2. Вычисление логарифма от неположительного числа |

После запуска программы на выполнение окно редактирования заменяется на окно просмотра. По умолчанию это окно имеет черный фон, на котором символами белого цвета отображается информация, возникающая по ходу выполнения программы – символы, набираемые на клавиатуре при выполнении ввода, данные, выводимые на экран при выполнении процедур вывода и служебная информация. После завершения выполнения программы окно просмотра закрывается и вновь появляется окно редактирования. Поскольку выполнение небольшой программы обычно занимает доли секунды, пользователь даже не успевает разглядеть выведенные результаты. Для их просмотра можно:

а) принудительно перейти из окна редактирования в окно просмотра, нажав "горячую" комбинацию клавиш <Alt> + <F5> (повторное нажатие этой комбинации вновь открывает окно просмотра);

б) приостановить завершение программы, записав в ней перед заключительным словом **End** обращение к "пустой" процедуре ввода: **readln;**. Для завершения программы надо, после просмотра и анализа результатов, нажать клавишу ввода <Enter>.

В системах BP 7.0 и Free Pascal имеется стандартный модуль CRT, позволяющий управлять выводом информации на экран (открывать в окне просмотра внутренние окна, раскраши-

вать их разными цветами, менять цвет выводимых символов, перемещать курсор в заданную точку и т.п.). Подробный анализ этого модуля выходит за рамки данного пособия, однако одна из его процедур весьма полезна.

Она называется **clrscr** (сокращение от английского "clear screen" – "очистить экран") и при ее выполнении из окна просмотра убирается вся информация, оставшаяся от выполнения предыдущих программ. Обращение к этой процедуре записывают сразу после слова **Begin**, открывающего раздел операторов (разумеется, промежуточную очистку экрана можно производить и по ходу процесса выполнения программы, если записать обращение к процедуре в соответствующем месте программы). Для того, чтобы можно было использовать данную процедуру, необходимо во второй строке программы (сразу после заголовка) записать указание о подключении модуля CRT: **uses CRT;**

#### 3.13. Набор и редактирование программы

Файл программы на Паскале представляет собой обычный текстовый файл. Этот файл можно открыть с помощью текстового редактора **Word** или **Блокнот**, скопировать текст в буфер обмена, вставить его в другой текстовый документ и т.п.

Набор программы ведется с помощью клавиатуры так же, как и набор любого текстового файла. Как сказано выше, в Паскале нет разницы между заглавными и строчными буквами, поэтому их можно использовать в любых сочетаниях, чтобы сделать текст более удобным для чтения. Также не имеет значения наличие лишних пробелов между отдельными элементами конструкций языка (словами, знаками препинания и т.п.) и наличие пустых строк, формируемых дополнительным нажатием на клавишу ввода.

Для улучшения читаемости программы и облегчения анализа ее структуры рекомендуется использовать ступенчатый набор текста программы. В соответствии с принципами ступенчатого набора все элементы программы – отдельные операторы, операторы и блоки, входящие в состав других операторов и бло-

ков (например, тела циклов и ветви условных операторов) – образуют иерархическую (многоуровневую) систему:

- оператор или блок, входящий в состав другого оператора, находятся на один уровень ниже этого оператора;
- операторы, входящие в состав блока, находятся на один уровень ниже операторных скобок **begin ... end** этого блока;
- часть оператора, записанная в отдельной строке, находится на один уровень ниже начальной строки этого оператора.

Начало строки, соответствующей более низкому уровню, сдвигается на 1-2 позиции вправо по отношению к началу строки предыдущего более высокого уровня. Для этого в начало строки вставляются 1-2 пробела.

Кроме того, каждый новый оператор принято начинать с новой строки. Операторные скобки **begin** и **end** набираются в отдельных строках.

Ступенчатый набор использован в примерах программ и фрагментов программ, приведенных в тексте данной главы.

Для облегчения ступенчатого набора в системах программирования на Паскале по умолчанию включен режим автоотступа. В соответствии с ним:

- каждый сдвиг начала строки, соответствующий переходу на более низкий уровень и сформированный вставкой 1-2 пробелов, запоминается системой, как единое целое.
- нажатие на клавишу <Delete> удаляет один символ, находящийся над курсором. Пробелы, стоящие в начале строки, также удаляются в этом случае по одному.
- нажатие на клавишу <Backspace> удаляет один символ внутри строки, стоящий слева от курсора, либо одну группу пробелов в начале строки, образующую очередной ступенчатый отступ (формирующую очередной уровень).
- при нажатии на клавишу ввода <Enter> курсор автоматически перемещается в следующую строку в позицию, соответствующую началу строки такого же уровня.

В ходе набора возможны два режима ввода символов – режим вставки (включен по умолчанию) и режим замены. Переключение режимов производится нажатием на клавишу <Insert>.

Если курсор находится в середине уже существующей строки, то в режиме вставки новый символ, вводимый с клавиатуры, вставляется в позицию курсора между существующими символами. При этом оставшаяся часть строки сдвигается вправо. В режиме замены новый символ вставляется вместо существующего символа.

### 3.14. Этапы разработки компьютерной программы

Процесс создания программы для решения какой-либо задачи состоит из следующих этапов.

#### I. Постановка задачи.

а) Формализация задачи. На этом этапе определяется, какими исходными данными следует располагать для решения, какую форму должен иметь результат (вычисленное значение, словесный ответ на поставленный вопрос и др.). Если задача требует проведения математических вычислений, проводится математическое описание решения.

б) Спецификация элементов программы. Спецификация – это подробное описание элементов (констант, переменных, меток, подпрограмм). Для констант и переменных указываются:

- Имя (идентификатор);
- Вид (константа, переменная);
- Тип величины;
- Смысл (что обозначает данная величина в задаче);
- Роль в задаче (исходные данные, результат, либо вспомогательная величина, используемая в ходе решения).

Для подпрограмм:

- записывается форма заголовка и приводится спецификация формальных параметров;
- дается описание результатов, которые должны быть достигнуты при выполнении подпрограммы.

#### II. Алгоритмизация.

Составляется алгоритм решения, который изображается блок-схемой, или записывается псевдокодом. В основе любого алгоритма лежит линейная структура, показанная на рисунке

3.14. Каждый из разделов программы может иметь сложную внутреннюю структуру с разветвлениями и повторениями.

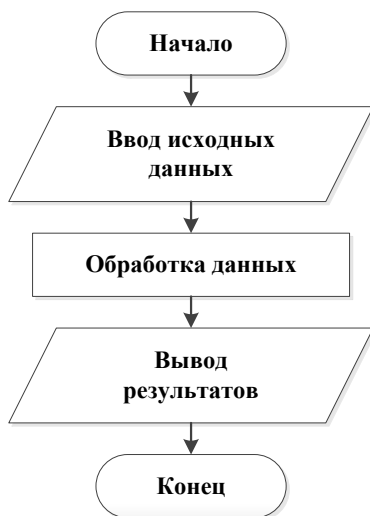


Рис. 3.14. Общая линейная структура программы

III. Составление и набор программы.

IV. Отладка программы.

Для отладки программы ее запускают на выполнение. Сначала система отыскивает в тексте синтаксические ошибки. Их необходимо проанализировать и исправить. После того, как программа запустится, ее тестируют, вводя такие наборы исходных данных, для которых заранее известен ответ. Если программа дает такой же ответ, можно предполагать, что она правильна

### 3.15. Общие требования к выполнению лабораторных работ

По каждой работе оформляется отчет в виде файла MS Word. Отчет должен содержать следующие пункты:

1. Титульный лист с указанием фамилии и инициалов студента, шифра учебной группы и номера варианта задания.
2. Запись условий решаемой задачи.



3. Постановку задачи
4. Блок-схему алгоритма, выполненную средствами графического редактора MS Word.
5. Текст программы
6. Скриншот экрана с изображением окна просмотра, в котором должны быть отражены результаты выполнения программы.

Текст отчета набирается шрифтом Times New Roman, кегль 12, текст программы – шрифтом Arial, кегль 14.

Примеры решения отдельных задач, приведенные далее в указаниях к отдельным лабораторным работам, оформлены в соответствии с требованиями к содержанию отчетов.

### 3.16. Лабораторная работа 1:

Запись сложных выражений.

Программирование линейного алгоритма

Цель работы: ознакомиться с основными символами и простейшими конструкциями языка Паскаль, освоить приемы программирования алгебраических выражений.

I. Рассмотрим несколько примеров записи алгебраических выражений:

- Выражение  $\operatorname{tg} x - \ln |a^3 - b^3|$ :

$$\sin(x)/\cos(x) - \ln(\operatorname{abs}(a*a*a - b*b*b))$$

Комментарий: так как функции тангенса (и котангенса) в языке Паскаль отсутствует, для вычисления используется отношение синуса и косинуса одного и того же аргумента. Вычисление небольшой целой степени (квадрата, куба) простого по записи аргумента можно заменить прямым умножением его на себя нужное число раз.

- Выражение  $e^{\frac{x}{2}} - \operatorname{tg} \frac{\pi}{x+4}$

$$\exp(x/2) - \sin(\operatorname{Pi}/(x+4))/\cos(\operatorname{Pi}/(x+4))$$

Комментарий: экспоненциальная функция  $e^x$  в языке Паскаль (и вообще в любых компьютерных программах) записывается в виде **exp(x)**. Для вычисления значения числа "Пи" в языке Паскаль используется встроенная функция **Pi** (эта функция не имеет аргумента и не требует записи скобок для его вставки).

- Выражение  $\sqrt[3]{\frac{x+y}{x}}$

$$\text{exp}(1/3*\ln((x+y)/x))$$

- Выражение  $\text{tg} \cos \sqrt[5]{\frac{x}{y}}$

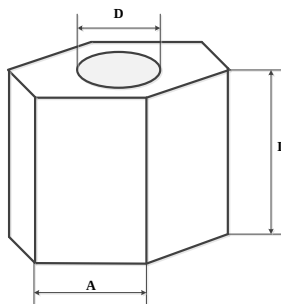
$$\sin(\cos(\exp(1/5*\ln(x/y)))) / \cos(\cos(\exp(1/5*\ln(x/y))))$$

II. Приведем примеры решения задач, связанных с программированием линейного вычислительного процесса.

### Пример 3.5.

#### I. Условия задачи:

Стальная деталь имеет форму правильной шестигранной призмы длиной L. Длина стороны основания призмы равна A. По оси детали просверлено сквозное отверстие диаметром D. Определить массу детали, если плотность стали равняется  $7900 \text{ кг/м}^3$ .



#### II. Постановка задачи.

##### а) Формализация.

Масса тела равняется произведению его объема на плотность материала. Для того чтобы вычислить объем детали, надо из объема шестигранной призмы вычесть объем, занимаемый

отверстием, т.е., объем цилиндра длиной **L** и диаметром **D**. Математическое описание задачи выглядит так:

$$\text{Объем призмы: } V_n = \frac{3A^2\sqrt{3}}{2}L$$

$$\text{Объем отверстия: } V_{\text{ц}} = \frac{\pi D^2}{4}L$$

$$\text{Масса детали: } m = \rho(V_n - V_{\text{ц}})$$

б) Спецификация данных.

Очевидно, что исходными данными для вычисления являются размеры детали **A**, **D**, **L** и плотность стали **ρ**. Т.к. размеры в задаче не указаны, программа должна предусматривать возможность вычисления массы деталей разных размеров, поэтому размеры следует объявить, как переменные, и вводить по ходу программы с клавиатуры. Плотность стали – это известная (справочная) величина. Ее можно использовать непосредственно при вычислении массы, как число, либо описать, как константу с заданным значением. Задание справочных величин, как заранее объявленных констант целесообразно в следующих случаях:

а) в сложных программах большого размера объявление констант с именами, имеющими наглядный смысл (например, соответствующими традиционным обозначениям физических величин), делает программу более ясной для понимания;

б) если запись числа является сравнительно громоздкой (например, содержащей много значащих цифр), а это число используется в программе неоднократно.

Результатом работы программы является число – значение массы **m**. Его можно вычислить с помощью одного выражения, подставив в формулу вычисления массы выражения для объемов. Однако при вычислении сложных выражений их лучше вычислять по частям, сохраняя промежуточные значения отдельных частей в виде дополнительных вспомогательных переменных. Введем такие переменные – **VP**, **VC** – для объемов призмы и отверстия (цилиндра).

### Глава 3. Основы программирования на языке ПАСКАЛЬ

Все переменные являются простыми и принадлежат к вещественному типу **real** (целочисленные типы выбираются только для переменных, используемых при подсчете или перечислении штучных объектов).

Тогда спецификация данных будет иметь вид:

| Имя       | Вид        | Тип         | Смысл                | Роль в программе |
|-----------|------------|-------------|----------------------|------------------|
| <b>A</b>  | Переменная | <b>real</b> | Длина стороны призмы | Исходная         |
| <b>L</b>  | Переменная | <b>real</b> | Длина детали         | Исходная         |
| <b>D</b>  | Переменная | <b>real</b> | Диаметр отверстия    | Исходная         |
| <b>RO</b> | Константа  | <b>real</b> | Плотность            | Исходная         |
| <b>VP</b> | Переменная | <b>real</b> | Объем призмы         | Вспомогательная  |
| <b>VC</b> | Переменная | <b>real</b> | Объем отверстия      | Вспомогательная  |
| <b>M</b>  | Переменная | <b>real</b> | Масса детали         | Результат        |

III. Блок-схема алгоритма приведена на рисунке 3.15.

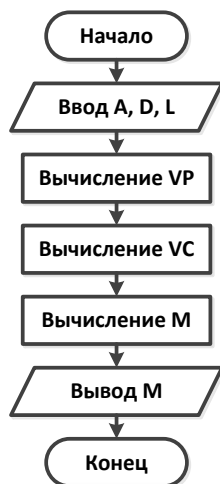


Рис. 3.15. Блок-схема алгоритма

IV. Текст программы:

```
program MASSA;
uses CRT;
const RO=7900;
var A, D, L, M, VP, VC : real;
Begin
  clrscr; {Очистка экрана}

  {Ввод исходных данных:}

  writeln ('Введите размер стороны призмы, диаметр
отверстия и длину детали в метрах: ');
  {Это была инструкция для пользователя}
  readln (A, D, L);

  {Вычисление:}

  VP := 3*R*R*sqrt(3)*L/2;
  VC := Pi*D*D*L/4;
  M :=RO*(VP – VC);

  {Вывод результата:}

  writeln ('Масса детали = ', M:8:2, 'кг');
  readln; {Задержка окончания программы}
End.
```

Замечание к тексту программы: при выполнении команды ввода **readln(...)** компьютер останавливается и ожидает ввода данных в порядке, соответствующем списку ввода. Чтобы пользователь не перепутал порядок, в котором надо вводить значения, в программу перед командой ввода вставляется команда вывода, по которой на экран выводится текстовое пояснение – что следует сделать. Такие пояснения называются *инструкциями для пользователя*.

Вид окна просмотра с результатами работы программы (в системе Free Pascal) показан на рисунке 3.16.

### Пример 3.6.

**I. Условия задачи:** Из трещины в трубопроводе происходит утечка ядовитого газа. Физические расчеты показывают, что смертельно опасная концентрация газа в помещении будет достигнута за **T** сек. Необходимо выразить это время в часах, минутах, секундах.

### **II. Постановка задачи.**

а) Исходными данными к задаче является значение временного интервала **T**. Результатами должны быть количество полных часов в этом интервале **H**, количество целых минут, уже не составляющих полного часа – **M** – и количество секунд, уже не составляющих целой минуты – **S**. При этом исходная величина временного интервала в общем случае должна выражаться вещественным числом, так как это физическая величина (которая может быть вычислена с точностью, например, до долей секунды). Значения **H** и **M** являются целыми числами, т.к. выражают количество полных часов и минут, а количество секунд также должно быть вещественным.

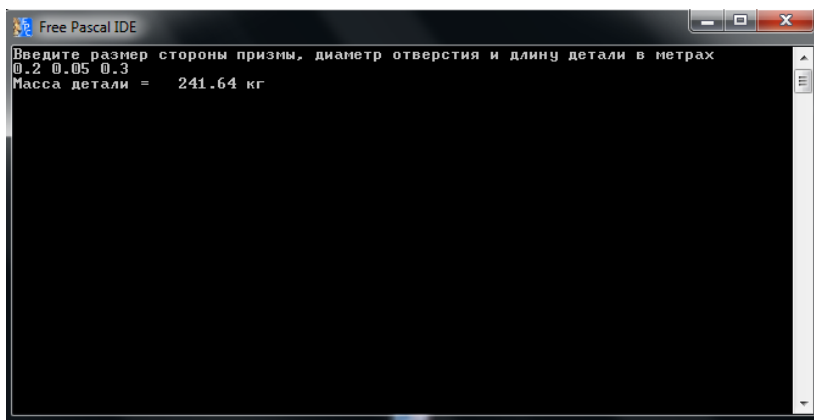


Рис. 3.16. Результаты работы программы (пример 3.5)

б) Формализация задачи.

Чтобы определить количество часов, входящих в заданный временной интервал, надо выделить в нем число порций, равных 1 часу (3600 сек). Это можно сделать с помощью операции целочисленного деления **div**. Однако эта операция применима только к целочисленным операндам, поэтому необходимо преобразовать вещественное число **T** в целое. Для этого следует использовать функцию **trunc(...)**. В итоге число часов выразится формулой

$$H = \text{trunc}(T) \text{ div } 3600$$

Если теперь от **T** отнять число (**3600·H**), выражающее число секунд, образующих **H** часов, то останется число секунд, уже не образующих целого часа. Для экономии памяти результат вычитания можно временно сохранить в ячейке памяти, соответствующей переменной **S**:

$$S = T - 3600 \cdot H$$

Из этого числа надо выделить число порций по 60, т.е., целых минут. Это делается так же, как и выделение числа часов:

$$M = \text{trunc}(S) \text{ div } 60$$

Число секунд **S**, не образующих целой минуты, получится, если от имеющегося промежуточного значения **S** отнять число секунд, образующих целые минуты – (**60·M**):

$$S := S - 60 \cdot M$$

в) Спецификация данных:

| Имя      | Вид        | Тип            | Смысл                                      | Роль в программе  |
|----------|------------|----------------|--------------------------------------------|-------------------|
| <b>T</b> | Переменная | <b>real</b>    | Время достижения опасной концентрации газа | Исходная величина |
| <b>H</b> | Переменная | <b>integer</b> | Число часов                                | Результат         |
| <b>M</b> | Переменная | <b>integer</b> | Число минут                                | Результат         |
| <b>S</b> | Переменная | <b>real</b>    | Число секунд                               | Результат         |

III. Блок-схема алгоритма приведена на рисунке 3.17.

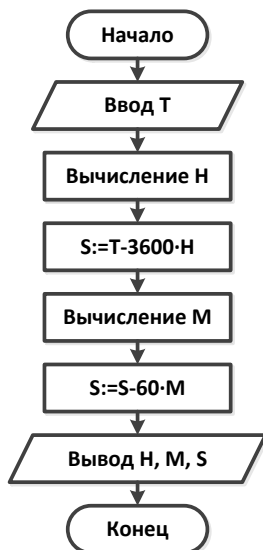


Рис. 3.17. Блок-схема алгоритма

IV. Текст программы:

```
program GAS;
uses CRT;
var T, S: real;
    H, M: integer;
Begin
  clrscr;
  write ('Введите величину интервала времени: ');
  readln (T);
  H:=trunc(T) div 3600;
  S:=T-3600*H;
  M:=trunc(S) div 60;
  S:=S-60*M;
  write ('Опасное время составляет ', H:3, ' ч', M:3,
    ' мин', S:5:1, ' сек');
  readln;
End.
```



### Глава 3. Основы программирования на языке ПАСКАЛЬ

Вид окна просмотра с результатами работы программы (в системе Free Pascal) показан на рисунке 3.18.

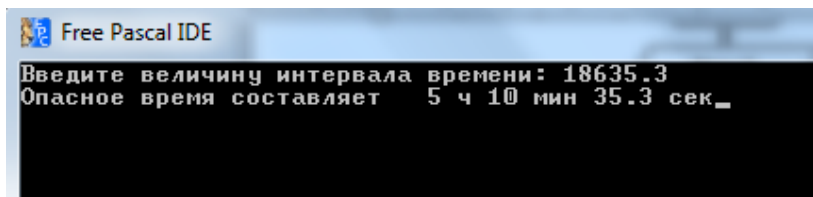


Рис. 3.18. Результаты работы программы

#### Задания для самостоятельного выполнения:

I. Записать на языке Паскаль 4 математических выражения (согласно индивидуальному варианту задания, табл. 3.8).

Таблица 3.8

Варианты заданий на запись математических выражений

| № | Выражение 1                           | Выражение 2                             | Выражение 3                                | Выражение 4                                 |
|---|---------------------------------------|-----------------------------------------|--------------------------------------------|---------------------------------------------|
| 1 | $\cos x \sin x^2$                     | $\ln \sin x^3 + \sqrt{a}$               | $a^3 \sqrt{b+c}$                           | $\frac{a+bc}{x+\frac{z}{y}}$                |
| 2 | $\cos^2 x + \operatorname{tg}(x+2)$   | $\ln \cos e^x$                          | $\sqrt[5]{a+bc} + \sin x^2$                | $\sin\left(\frac{x+a}{b} + \sqrt{c}\right)$ |
| 3 | $\sqrt{\sin^2 x}$                     | $\ln \sin \sqrt{(x+a)^3}$               | $\sqrt[5]{c+\sqrt{b}}$                     | $\frac{a+b}{c+\frac{y}{x-p}}$               |
| 4 | $\sin^2 x + \sqrt{\ln x^3}$           | $\sqrt{\operatorname{tg} \ln \cos x^3}$ | $x^p + \ln \sqrt[3]{x}$                    | $\frac{x+y}{a} + \sqrt{b+c^2}$              |
| 5 | $\sin^2 x + \operatorname{tg}(a+b)^2$ | $\ln \sin \sqrt{(a+b)^3}$               | $\sin e^{y+m}$                             | $\frac{x+y}{a} + \sqrt{b+c^2}$              |
| 6 | $\ln x^2 + \cos^2(x+y)$               | $\frac{a+b \sin x}{\sqrt{x} + \ln y}$   | $\operatorname{tg} \cos \ln \sqrt{x^3}$    | $ae^{\sin x^2}$                             |
| 7 | $\frac{ab}{\sin^2 x + \cos y^2}$      | $\frac{x+\sqrt{y}}{m-\frac{p}{\cos x}}$ | $\operatorname{tg} \ln \sqrt{\cos(x+y)^3}$ | $pe^{\cos^2 y^3}$                           |

### Глава 3. Основы программирования на языке ПАСКАЛЬ

| №  | Выражение 1                    | Выражение 2                              | Выражение 3                                  | Выражение 4                |
|----|--------------------------------|------------------------------------------|----------------------------------------------|----------------------------|
| 8  | $\sqrt{\cos^3(a+b)}$           | $\frac{x+b\sqrt{x}}{c-tgy}$              | $\cos \ln tg \sqrt{(x+y)^5}$                 | $e^{y+\ln x}$              |
| 9  | $\sqrt{ab} + \cos^2(x+y^3)$    | $\frac{a+x}{m+\frac{p}{\ln x + \cos y}}$ | $\ln \cos tg \sqrt{(a+b)^3}$                 | $ae^{\sin x + b^2}$        |
| 10 | $\cos(y+x)^2 + \sin^2 x$       | $tg \cos \ln \sqrt{2,6x^3}$              | $\frac{a+b^2(a-\ln t)}{\sqrt{c} + m \cos y}$ | $x^{-2b}$                  |
| 11 | $\sqrt{(a+b)^3} + \cos^2 y^3$  | $\frac{x+y(m-p)}{a+t}$                   | $tg \sin \ln \sqrt{a^3}$                     | $me^{(a+b)^2 \sin y}$      |
| 12 | $\ln y^2 + \sqrt{m+p^3}$       | $\frac{x-a}{c-\frac{b}{\cos t + m}}$     | $e^{m+tgx}$                                  | $\ln tg \cos e^{m+p}$      |
| 13 | $\sqrt{\sin x^2 + \cos^2 y}$   | $\frac{m+\cos y}{\sqrt{c} + \sqrt{b+p}}$ | $\cos e^{x+y}$                               | $tg \cos \ln \sqrt{x+a^3}$ |
| 14 | $tg t^2 + \cos \sqrt{a+b}$     | $\sin x + \frac{a+b}{\sqrt{c}}$          | $e^{\cos x + \sqrt{y}}$                      | $2,3e^{\ln \cos tg(x+y)}$  |
| 15 | $\ln x^2 + \sqrt{tg^3 t}$      | $\frac{x}{x^2+y} \sqrt{a+b}$             | $e^{m+x^2 \sin y}$                           | $\cos \ln tge^{m+y}$       |
| 16 | $\sqrt{\sin x^2 + \cos^2 y}$   | $\frac{a+b}{\sin y + \frac{x}{m^2}}$     | $\cos \ln \frac{x+b}{\sqrt{m+2,3p}}$         | $e^{2,3c+x \sin \sqrt{y}}$ |
| 17 | $\frac{a+b}{cb^2}$             | $\frac{\sin ax}{\ln a}$                  | $x^{0,3} + 2xy$                              | $\sqrt{a \ln 3}$           |
| 18 | $\arcsin x^2 + \cos^2 x$       | $\cos x + \frac{a}{b+\frac{c}{p}}$       | $ae^{t \cos y}$                              | $\ln tg \cos \sqrt{a+b}$   |
| 19 | $\cos y^3 + \sin^4(x+a)$       | $\frac{a+t}{m^2} + \frac{c}{y+b}$        | $\cos e^{t^2+mp}$                            | $\sqrt{tg \ln(x^3+c)}$     |
| 20 | $\sqrt{x^3} + \arccos y$       | $m + \frac{p \sin y}{\cos^2 x + a}$      | $\sqrt{m} + e^{t+\sin y^2}$                  | $\ln \cos y + x^2$         |
| 21 | $\sin y + \cos^2 \sqrt{x}$     | $e^{t+c \ln x}$                          | $a+b \cos x + \frac{m}{b+y}$                 | $tge^{\ln x+y}$            |
| 22 | $\ln x^4 + \cos^2 \frac{a}{b}$ | $m + \frac{p}{\cos(x+y^2)}$              | $e^{2,3x+m \cos^2 y}$                        | $e^{\ln \cos \sqrt{x^3}}$  |

| №  | Выражение 1                     | Выражение 2                             | Выражение 3               | Выражение 4                  |
|----|---------------------------------|-----------------------------------------|---------------------------|------------------------------|
| 23 | $\cos x^2 + \sqrt{\sin^3 x}$    | $\frac{p}{a+b} \sin \sqrt{\frac{x}{y}}$ | $e^{\ln x + \sqrt{m+p}}$  | $tg \ln \cos \sqrt{x} + y^3$ |
| 24 | $\sqrt{\sin^a} + tg^3 \sqrt{m}$ | $\frac{a}{b+c} \sqrt{\sin y}$           | $e^{\frac{x}{b+c}}$       | $tg \ln e^{m+y}$             |
| 25 | $2,6 \sin^2 x^3$                | $\frac{a+b}{c + \frac{y}{\ln m+t}}$     | $e^{\sqrt{x^2 + \cos y}}$ | $tg \ln \cos e^{m^3}$        |

II. Составить программу решения задачи согласно индивидуальному варианту задания:

1. Период колебаний **T** маятника длиной **L** определяется по формуле

$$T = 2\pi \sqrt{\frac{L}{g}}$$

Вычислить, сколько колебаний совершит маятник за время **T<sub>1</sub>**.

2. Вычислить силу притяжения **F** между телами массами **M<sub>1</sub>** и **M<sub>2</sub>**, находящимися на расстоянии **R** друг от друга:

$$F = G \frac{M_1 \cdot M_2}{R^2}$$

где  $G = 6.673 \cdot 10^{-11} \text{ м}^3/\text{кг}$ .

В качестве одной из масс при тестировании программы использовать массу Земли **M<sub>1</sub>**=5.98·10<sup>24</sup> кг, радиус Земли **R**=6371 км. При вычислении должна получиться сила тяжести в Ньютонах (Н).

3. Вычислить площадь треугольника, если задан его полупериметр **p** и углы (**α**, **β**, **γ**) в градусах. Площадь вычисляется по формуле

$$S = p^2 \cdot tg \frac{\alpha}{2} \cdot tg \frac{\beta}{2} \cdot tg \frac{\gamma}{2}.$$

Замечание: в программе необходимо предусмотреть перевод углов в радианную меру.

4. Маляру поручили покрасить крышу и стенки бака для бензина в форме цилиндра. Определить, сколько потребуется банок с краской, если известны размеры бака и площадь, которую можно покрасить из одной банки.

5. Втулка имеет форму полого цилиндра с размерами: **R** - внешний радиус, **T** - толщина стенки, **H** - высота. Плотность материала, из которого изготовлена втулка – **RO**. Определить массу втулки и площадь ее поверхности.

6. Из прямоугольного листа вырезают 2 круга, треугольник и прямоугольник заданных размеров. Определить процент используемого материала.

7. Дано натуральное число. Определить число сотен и тысяч в нем (т.е., третью и четвертую с конца цифры).

8. Вычислить объем детали и ее массу, если деталь имеет форму прямоугольного параллелепипеда с 4 сквозными отверстиями цилиндрической формы равных размеров. Размеры детали и плотность заданы.

9. Определить, какой процент составляет зеленая зона от площади жилого квартала, если квартал имеет форму прямоугольника, зеленая зона образует две пересекающихся под прямым углом полосы заданной ширины по осям этого прямоугольника. Известны размеры квартала и ширина зеленых полос.

10. Три электрических сопротивления соединены параллельно. Найти общее сопротивление такого соединения.

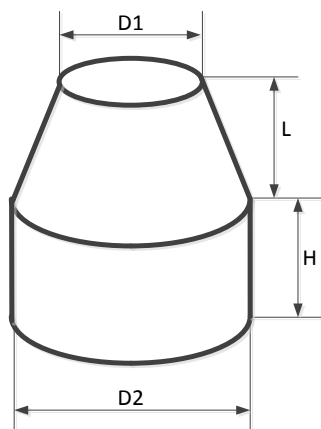
11. Найти периметр правильного n-угольника, описанного около окружности радиусом **r**. Периметр вычисляется по формуле

$$p = 2nr \cdot \operatorname{tg} \frac{\pi}{n}$$

12. Вычислить месячную выплату **m** возврата по кредиту в **S** рублей на **n** лет под процент **p** по формуле:

$$m = \frac{rS(1+r)^n}{12((1+r)^n - 1)}, \quad \text{где} \quad r = \frac{p}{100}$$

13. Сырье хранится в емкости, имеющей форму усеченного конуса, совмещенного с цилиндром. Определить объем емкости и массу сырья, если заданы размеры емкости и плотность сырья.



14. Определить массу газа, находящегося в баллоне объемом  $V$  (л) при температуре  $t$  ( $^{\circ}\text{C}$ ) и давлении  $p$  (Па). Использовать уравнение Менделеева – Клапейрона

$$pV = \frac{m}{\mu} RT$$

где  $\mu$  – молярная масса газа,  $R = 8314.41$  Дж/(кмоль·К) – универсальная газовая постоянная,  $T$  – абсолютная температура воздуха, К

Указание: обратить внимание на единицы измерения исходных величин. В программе предусмотреть перевод значений в международную систему единиц СИ.

15. В пункте с географическими координатами  $48^{\circ}$  северной широты и  $45^{\circ}$  восточной долготы высота Солнца 15 июня определяется формулой:

$$\sin(h) = 0,295 + 0,614 \cdot \cos(t) .$$

Здесь  $t = 15(T - 1)$ , где  $T$  – время в часах пополудни. Найти высоту Солнца в заданный момент времени  $T$ .

16. "Барометрическая" формула, определяющая связь среднего атмосферного давления и высоты местности над уровнем моря, имеет вид:

$$h = 44300 \left[ 1 - \left( \frac{p}{p_0} \right)^{0,19} \right],$$

где **h** – высота над уровнем моря в метрах; **p** – давление на данной высоте, мм ртутного столба, атмосферное давление **p<sub>0</sub>** = 760 мм рт. столба. Вычислить **h** при заданном давлении.

17. Даны два угла треугольника в градусах. Определить третий угол треугольника в радианах.

18. ("Задача жестианщика"). Из круга радиуса **R** вырезан прямоугольник, большая сторона которого равна **a**. Определить радиус круга, который можно вырезать из этого прямоугольника (экстремальные тесты: **a=2R**, **a=1,41421 R**).

19. Дана сторона треугольника и два угла, прилежащих к ней (в градусах, минутах, секундах). Определить остальные стороны треугольника. (Указание: воспользоваться теоремой синусов).

20. Длина отрезка задана в дюймах с точностью до сотых. Выразить ее в метрах, сантиметрах и миллиметрах, например: 49, 05 дюйма = 1 м 24 см 6 мм (1 дюйм = 2,54 см).

21. Селекционер вывел новый сорт пшеницы и снял с опытной делянки **k** кг семян. Посеяв 1 кг семян, можно за сезон собрать урожай **p** кг семян. Через сколько лет селекционер сможет засеять новой культурой поле площадью **S** га, если норма высева **n** кг/га.

22. Русские меры длины: 1 верста = 500 сажень, 1 сажень = 3 аршина, 1 аршин = 16 вершков, 1 вершок = 44,45 мм. Длина отрезка составляет **p** м. Выразить его в русских мерах.

23. Текущее время задано в часах, минутах и секундах. Округлить его до целых минут.

24. Заданы моменты начала и конца некоторого промежутка времени в часах, минутах и секундах (в пределах одних

суток). Найти продолжительность этого промежутка в тех же единицах.

25. Даны 4 целых числа. Первые три задают время старта ракеты в часах, минутах и секундах. Четвертое – продолжительность полета в секундах. Определить в часах, минутах и секундах время приземления ракеты.

### 3.17. Лабораторная работа 2: Программирование алгоритмов с разветвлением

Цель работы: знакомство с правилами записи логических выражений, операций сравнения, операторов IF, CASE; освоение приемов программирования алгоритмов с разветвлением.

Рассмотрим пример программирования разветвляющегося алгоритма.

#### Пример 3.7.

**I. Условия задачи:** Решить квадратное уравнение

$$ax^2 + bx + c = 0.$$

Если уравнение не имеет вещественных решений, вывести на экран сообщение об этом.

#### II. Постановка задачи.

Исходными данными в задаче являются коэффициенты уравнения **a, b, c**.

а) Формализация задачи.

Процесс решения начинается с вычисления дискриминанта

$$D = b^2 - 4ac.$$

Далее возможны три случая:

а) если дискриминант отрицателен, уравнение не имеет вещественных решений;

б) если дискриминант равен нулю, уравнение имеет один вещественный корень

$$x = -\frac{b}{2a}$$

в) если дискриминант положителен, уравнение имеет два вещественных корня

$$x_1 = \frac{-b - \sqrt{D}}{2a}, \quad x_2 = \frac{-b + \sqrt{D}}{2a}$$

#### б) Спецификация данных

Для сохранения результатов – значений корней – введем переменные **X1** и **X2**. Если корень один, его значение сохраним в переменной **X1**. Если решения нет, результатом будет вывод на экран текста "Нет корней".

Спецификация данных имеет вид:

| Имя       | Вид        | Тип         | Смысл                 | Роль в программе           |
|-----------|------------|-------------|-----------------------|----------------------------|
| <b>A</b>  | Переменная | <b>real</b> | Коэффициент уравнения | Исходные данные            |
| <b>B</b>  | Переменная | <b>real</b> | Коэффициент уравнения | Исходные данные            |
| <b>C</b>  | Переменная | <b>real</b> | Коэффициент уравнения | Исходные данные            |
| <b>D</b>  | Переменная | <b>real</b> | Дискриминант          | Вспомогательная переменная |
| <b>X1</b> | Переменная | <b>real</b> | Корень уравнения      | Результат                  |
| <b>X2</b> | Переменная | <b>real</b> | Корень уравнения      | Результат                  |

### III. Алгоритмизация задачи:

Алгоритм решения должен начинаться с задания исходных данных. Ими в данном случае являются коэффициенты уравнения – **a, b, c**. Затем вычисляется дискриминант.

Далее необходимо осуществить разветвление алгоритма, т.к. в зависимости от набора исходных данных может реализоваться один из трех вышеперечисленных случаев – "а", "б" или "в". Однако условный оператор предусматривают разветвление только на два возможных варианта. Поэтому сначала определим



– имеет ли вообще уравнение действительные корни. Условие отсутствия корней имеет вид:  $D < 0$ . При истинности этого неравенства действительных корней у уравнения не будет (реализуется вариант "а").

Ложное значение неравенства означает, что  $D \geq 0$  (здесь выполняется правило исключенного третьего).

Если  $D < 0$ , следует вывести сообщение, что действительных корней нет, и на этом закончить процесс решения. Однако в противоположном случае опять возникает две возможности – уравнение может иметь или один, или два корня (варианты "б" или "в"). Здесь снова надо организовать разветвление. Условием разветвления может служить как условие наличия одного корня –  $D = 0$ , так и условие наличия двух корней –  $D > 0$ . Важно, однако, понять, что, оказавшись на этой ветви первого разветвления, мы делаем выбор только из двух возможностей ( $D = 0$  или  $D > 0$ ), а значит, можем опять использовать правило исключенного третьего. Если поставить, например, условие  $D = 0$ , то его невыполнение автоматически будет означать, что оставшееся условие ( $D > 0$ ) выполнено, и специально проверять выполнение этого условия уже не нужно. На ветвях второго разветвления реализуются вычисления по вариантам "б" или "в".

Блок-схема программы показана на рисунке 3.19.

#### IV. Текст программы:

```
program SQ_EQ;  
uses CRT;  
var A,B,C,D,X1,X2:real;  
Begin  
  clrscr;  
  write ('Введите коэффициенты A, B, C: ');  
  readln (A,B,C);  
  D:=B*B-4*A*C;  
  if D<0 then  
    writeln ('Нет корней')  
  else  
    if D=0 then
```

**begin**

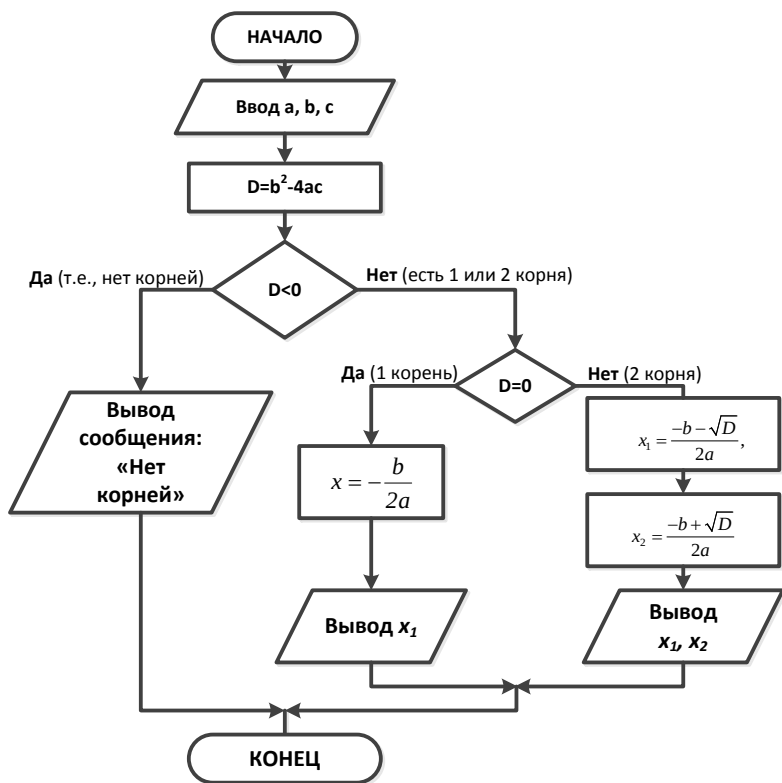


Рис. 3.19. Блок-схема программы решения квадратного уравнения

```

X1:=-B/2/A;
writeln ('Уравнение имеет один корень: ',
X1:8:3);
end
else
begin
X1:=(-B-sqrt(D))/2/A;
X2:=(-B+sqrt(D))/2/A;

```

```
writeln ('Корни уравнения: ',X1:8:3,' и  
        ',X2:8:3);  
end;  
readln;  
End.
```

Примеры работы программы (содержимое окна просмотра) показаны на рисунке 3.20.

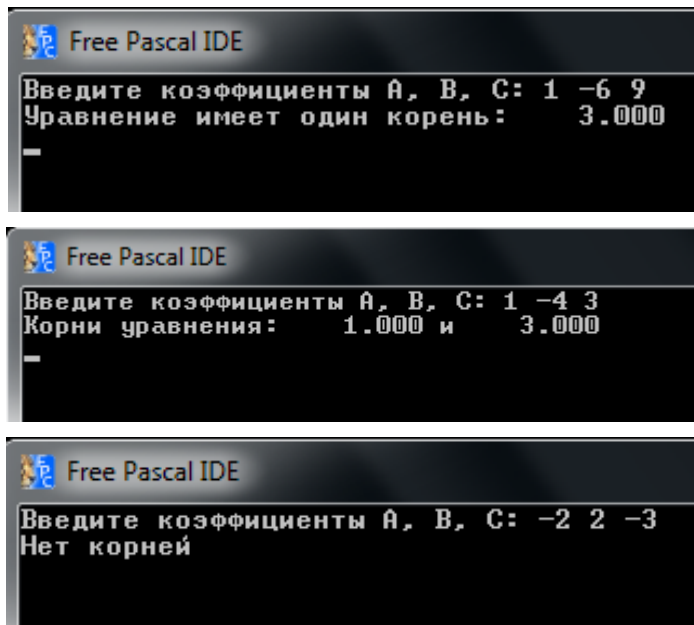


Рис. 3.20. Примеры работы программы решения квадратного уравнения

Задания для самостоятельного выполнения:

1. Даны 3 положительных числа. Составьте программу для определения возможности построения треугольника со сторонами, равными указанным числам.

2. Составьте программу для определения принадлежности точки **P** с координатами **(x, y)** области в виде прямоугольника с координатами главной диагонали: **(x<sub>1</sub>, y<sub>1</sub>)**, **(x<sub>2</sub>, y<sub>2</sub>)**.

3. Составьте программу для определения принадлежности точки **P** с координатами **(x, y)** области, которая задана кольцом с центром в начале координат и радиусами **R<sub>1</sub>** и **R<sub>2</sub>**.

4. Составьте программу для определения принадлежности точки **P** с координатами **(x, y)** области, которая образована первой дугой синусоиды и осью **Ox**.

5. Составьте программу для определения принадлежности точки **P** с координатами **(x, y)** области, которая задана квадратом со стороной **a**, повернутым на 45°, с центром в начале координат.

6. Составьте программу для определения принадлежности точки **P** с координатами **(x, y)** области в форме полукруга. Центр окружности лежит в точке **x<sub>0</sub>, y<sub>0</sub>**, ее радиус **-R**. Полукруг образуется отсечением от круга его нижней левой части, отсекающий диаметр проведен под углом 45° к осям координат.

7. Даны длины трех отрезков. Если эти отрезки могут быть сторонами треугольника, вычислить его площадь, используя формулу Герона:

$$S = \sqrt{p(p-a)(p-b)(p-c)},$$

где **p** – полупериметр треугольника.

8. Выяснить, сколько действительных корней имеет биквадратное уравнение:

$$ax^4 + bx^2 + c = 0$$

9. Определить, принадлежит ли треугольник, заданный координатами трех вершин, окружности с центром в начале координат и заданным радиусом.

10. Решить систему линейных уравнений:

$$\begin{cases} a_1x + b_1y = c_1 \\ a_2x + b_2y = c_2 \end{cases}$$

используя правило Крамера. Исследовать систему уравнений: сравнить с нулем главный определитель и побочные определители и сделать вывод о наличии решений системы. Если система имеет решение, то его надо найти, если нет – вывести на экран сообщение об этом.

11. Определить, удовлетворяет ли изделие стандарту, если размеры его **x**, **y**, **z**. Анализ по допускам выполняется следующими соотношениями: **c** <= **x** <= **D**; **s** <= **y** <= **T**; **z** >= **P**.

12. Подходный налог взимается в зависимости от годового дохода **S**:

$$p = \begin{cases} 0 & \text{при } S < 20\,000 \\ 0.13(S - 20\,000) & \text{при } S > 20\,000 \end{cases}$$

Определить сумму подходного налога для произвольного дохода **S**.

13. Можно ли из листа с размерами **a** и **b** вырезать круг радиусом **R**? Если можно, то подсчитать процент использованного материала.

14. По заданным координатам точки определить, какому квадранту координатной плоскости она принадлежит.

15. Треугольник задан длинами своих сторон **a**, **b**, **c**. Определить, является ли он остроугольным, прямоугольным или тупоугольным. Указание – используя теорему косинусов и тот факт, что наибольший угол треугольника противолежит его наибольшей стороне, определить знак косинуса наибольшего из углов.

16. Банк предлагает три вида срочных вкладов: под **p<sub>1</sub>**% на 3 месяца, под **p<sub>2</sub>**% на 6 месяцев и под **p<sub>3</sub>**% на год (все вклады – с капитализацией процентов). Какой вклад наиболее выгоден для вкладчика, если срок хранения составляет один год?

17. Написать программу, определяющую, можно ли прямоугольник со сторонами **a**, **b** поместить внутри прямоугольника со сторонами **c**, **d**.

18. Написать программу, определяющую, можно ли треугольник со сторонами **a**, **b**, **c** поместить внутри круга радиусом **R**.

19. Треугольник задан 3 сторонами. Определить вид этого треугольника (равнобедренный, равносторонний, разносторонний).

20. Даны координаты трех точек на плоскости. Определить, образуют ли эти точки прямоугольный треугольник. Указание: после определения длин сторон проверить выполнение теоремы Пифагора.

21. По заданным координатам двух точек определить какая из них расположена ближе к началу координат. Вывести на экран координаты этой точки.

22. Определить, какая из двух фигур (круг или квадрат) имеет большую площадь. Известно, что сторона квадрата равна **a**, радиус круга **r**. Вывести на экран название и значение площади большей фигуры.

23. Даны три числа, из которых одно из чисел кратно другому, а третье не кратно ни одному из остальных. Вывести на экран сообщение: "Число ... кратно числу ...".

24. Даны три числа. Составить программу, заменяющую каждое отрицательное число его модулем, а нулевое – значением числа  $\pi$ .

25. Определить, является ли данное число четным. Если оно четное, определить также, кратно ли оно еще и трем.

#### 3.18. Лабораторная работа 3:

##### Одномерные массивы и их обработка

Цель работы: знакомство с правилами описания и использования переменных типа "массив"; освоение приемов програм-

мирования алгоритмов с использованием циклических структур и массивов.

Массив (**array**) представляет собой набор однотипных простых значений, объединенных общим именем и различаемых внутри набора по номерам (индексам). В одномерном массиве у каждого элемента имеется один индекс, в многомерных массивах число индексов равно двум или более.

Математическим аналогом одномерного массива служит последовательность чисел, двумерного - матрица. В Паскале элементами массива могут быть не только числа, но и величины любого другого типа – символы, строки, записи и др.

Отдельный элемент массива записывается так: ставится общее имя массива и за ним в квадратных скобках – индекс данного элемента: **A[1]**, **POS1[N]** и т.п.

Массив должен быть описан (объявлен) в разделе описаний программы. Это можно сделать двумя способами

- Непосредственно описать массив, как переменную.
- Сначала описать тип данных, имеющих структуру массива, а потом описать сам массив, как переменную описанного типа.

Примеры описаний см. выше (§ 3.7).

При описании массива количество его элементов должно быть указано как конкретное число. Нельзя при описании массива указать число его элементов, как переменную, а потом в программе задать это число, например, вводом с клавиатуры (такие действия допустимы в ряде других языков программирования). Это создает затруднения, если впоследствии предполагается использовать программу для обработки массивов с разным числом элементов. Есть два способа обойти это затруднение:

1. Заранее описать массив с избыточным количеством элементов, а в программе с клавиатуры вводить значение конкретного числа элементов, которые будут практически использованы. Однако при этом приходится использовать лишние ячейки памяти, которые таким образом расходуются впустую;
2. Предварительно описать число элементов, как константу с конкретным значением, а при описании массива указать эту константу как количество элементов массива. Впоследствии при

необходимости обработать массив с другим числом элементов надо будет изменить только значение константы и не потребуются вносить изменения в остальной текст программы. Пример такого описания:

```
.....  
const N = 10;  
var A: array [1..N] of real;  
.....
```

В программировании числовые массивы используются при решении задач обработки экспериментальных и статистических данных, решении дифференциальных уравнений сеточными методами и др. Массивы записей являются основой баз данных.

Разберем поэтапно работу с одномерными массивами.

#### Ввод значений массива

Это действие должно быть первым в любой программе по обработке массива. Оно является типовым и выглядит так (для примера показан ввод с клавиатуры пяти элементов одномерного массива **A [ J ]**):

```
.....  
for J := 1 to 5 do  
  begin  
    write('Введите значение элемента A[ ', J : 3, ' ] : ');  
    {Это пояснительная инструкция для пользователя}  
    readln( A [ J ] );      {Ввод значения с клавиатуры}  
  end;  
.....
```

#### Обработка массива

Многообразные задачи обработки массивов можно в основном свести к комбинации сравнительно небольшого числа типовых алгоритмов. Некоторые из таких алгоритмов:



1. Преобразование элементов, в т.ч. выборочное, когда преобразуются только элементы, удовлетворяющие какому-либо условию (далее для краткости такие элементы именуются "подходящими");

2. Вычисление числовой характеристики массива (суммы элементов, их произведения, количества и др., в т.ч. выборочно – только для подходящих элементов);

3. Определение максимального (минимального) элемента и его номера;

4. Определение наличия в массиве подходящих элементов;

5. Определение номера первого или последнего из подходящих элементов;

6. Поиск максимального (минимального) среди подходящих элементов;

7. Фильтрация (создание нового массива, в который входят только подходящие элементы);

8. Сортировка массива – перестановка его элементов в порядке возрастания или убывания.

Далее разберем эти алгоритмы по порядку. В приводимых ниже примерах всюду будет рассматриваться массив **A[1..N]**, индекс элемента массива будет обозначаться **J**.

Основой большинства алгоритмов обработки массивов служит счетный цикл (рис. 3.21).

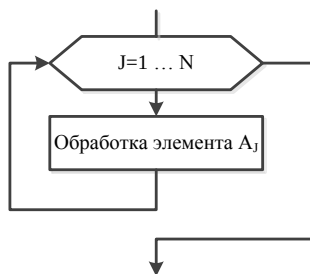


Рис. 3.21. Счетный цикл обработки всех элементов массива

При выполнении такого цикла по очереди обрабатываются все элементы массива от **A[1]** до **A[N]**. Примером реализации такой структуры является описанный выше ввод значений эле-

ментов с клавиатуры (обработкой элемента в этом случае является ввод с клавиатуры его значения).

Если обработке подлежат не все элементы, а только подходящие по условию, то телом цикла является неполный условный оператор (рис. 3.22).

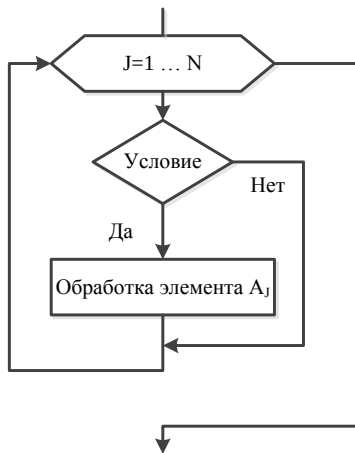


Рис. 3.22. Счетный цикл выборочной обработки отдельных элементов массива, подходящих по условию

Наконец, если в зависимости от выполнения условия подходящие элементы обрабатываются одним способом, а остальные – другим, то телом цикла будет полный условный оператор.

Примеры типичных условий, накладываемых на элементы массива:

а) Отбор элементов со значениями, кратными заданному числу **K** (только для элементов целого типа):

$$A[J] \bmod K = 0$$

б) Отбор элементов с индексами, кратными **K**:

$$J \bmod K = 0$$

в) Отбор элементов, значения которых попадают в интервал между числами **B** и **C**:

$$(A[J] > B) \text{ and } (A[J] < C)$$

г) То же для отрезка между числами **B** и **C**

**(A[J] >= B) and (A[J] <= C)**

Теперь перейдем к типовым алгоритмам.

1. Преобразование элементов массива.

Преобразование элементов полностью описывается схемами, представленными на рисунках 3.21, 3.22.

**Пример 3.8.** Увеличить все элементы массива в 1,5 раза (обработка всех элементов, схема – рис. 3.21):

```
.....  
for J := 1 to N do  
  A[J]:=1.5*A[J];  
.....
```

**Пример 3.9.** Увеличить в 1,5 раза те элементы массива, которые по абсолютной величине не превышают 5 (выборочная обработка, схема – рис. 3.22):

```
.....  
for J := 1 to N do  
  if abs(A[J])<=5 then  
    A[J]:=1.5*A[J];  
.....
```

**Пример 3.10.** Положительные элементы массива увеличить в 1,5 раза, а у отрицательных изменить знак (использование в теле цикла полного условного оператора)

```
.....  
for J := 1 to N do  
  if A[J]>0 then  
    A[J]:=1.5*A[J]  
  else  
    A[J]:= -A[J];  
.....
```

Замечание: если элемент равен нулю (т.е., не положительный и не отрицательный), то его преобразование любым из указанных способов не влияет на его значение – оно все равно останется равным нулю; поэтому нулевой элемент можно преобразовать любым способом. В данном примере если элемент равен нулю, то он не удовлетворяет указанному в операторе условию и преобразуется по второму способу, т.е.  $0 := -0$ . Отдельная обработка нулевого элемента в данном случае не нужна.

2. Вычисление какой-то характеристики массива (суммы элементов, их произведения, количества и др., в т.ч. выборочно – только для подходящих элементов);

В таких алгоритмах появляется необходимость еще в одном действии – инициализации (задании начального значения) вычисляемой характеристики. Блок-схема алгоритма (для случая обработки всех элементов массива) показана на рисунке 3.24.

Смысл обработки элемента зависит от решаемой проблемы. При вычислении суммы элементов значение очередного элемента прибавляется к общей сумме, при вычислении произведения соответствующая переменная умножается на очередной элемент, при подсчете количества вычисляемая переменная увеличивается на единицу и т.д. (см. примеры).

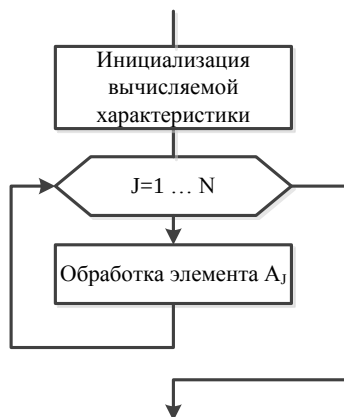


Рис. 3.23. Вычисление характеристики массива

**Пример 3.11.** Вычислить произведение всех элементов массива (организация цикла по схеме рис. 3.21):

```
.....  
pr:=1;  
for J := 1 to N do  
  pr:=pr*A[J];  
.....
```

**Пример 3.12.** Вычислить сумму положительных элементов массива (организация цикла по схеме рис. 3.22):

```
.....  
sum:=0;  
for J := 1 to N do  
  if A[J]>0 then  
    sum:=sum+A[J];  
.....
```

**Пример 3.13.** Определить количество четных элементов массива (организация цикла по схеме рис. 3.22):

```
.....  
kol:=0;  
for J := 1 to N do  
  if A[J] mod 2 = 0 then  
    kol:=kol+1;  
.....
```

3. Определение значения максимального (минимального) элемента в массиве и его номера.

Структурно этот алгоритм (рис. 3.24) похож на алгоритм № 2 с выборочной обработкой элементов (**MAX**– значение максимального элемента, **NOM** – его номер).

Отличие состоит в том, что определяются две характеристики массива – значение элемента и его номер. Поэтому при инициализации переменных и обработке очередного элемента всякий раз выполняется не одно, а два действия – одно для значения элемента **MAX**, другое – для его номера **NOM**.

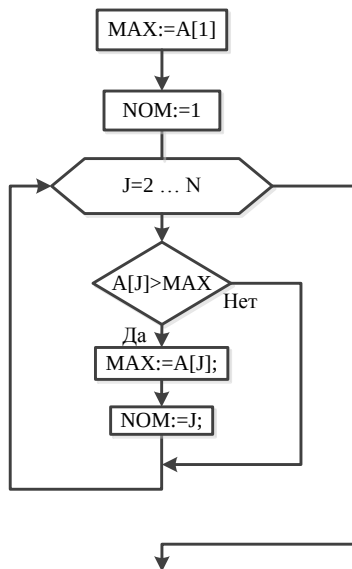


Рис. 3.24 Определение значения максимального элемента и его номера.

Смысл алгоритма в том, что сначала предполагается: максимальным может оказаться первый элемент массива. На основе этого предположения производится инициализация переменных **MAX** и **NOM**. Затем по очереди рассматриваются последующие элементы (начиная со второго) и их значения сравниваются с **MAX**. Если обнаружится элемент, имеющий большее значение, чем **MAX**, то это значение запоминается в переменной **MAX**, а номер элемента – в переменной **NOM**. Меньшие по значению элементы пропускаются. Таким образом, переменные **MAX** и **NOM** на каждом шаге хранят значение и номер наибольшего из уже просмотренных элементов. После завершения цикла они

сохранят соответствующие значения уже для всего просмотренного массива

Соответствующий фрагмент программы:

```
.....  
MAX := A[1];  
NOM := 1;  
for J := 2 to N do  
  if A[J] > MAX then  
    begin  
      MAX := A[J];  
      NOM := J;  
    end;  
.....
```

Поскольку в условном операторе на ветви "Да" (после слова **then**) необходимо выполнить два оператора присваивания, они объединяются в блок.

Если необходимости в определении номера максимального элемента нет, действия с переменной **NOM** не нужны. Тогда отпадает надобность и в создании блока.

В случае, если требуется определить значение минимального элемента, блок-схема и соответствующий ей фрагмент программы выглядят так же (разумеется, вместо имени переменной **MAX** естественно использовать имя **MIN**). Меняется только знак неравенства – вместо символа > ("больше") ставится символ < ("меньше").

#### 4. Определение наличия в массиве подходящих элементов.

Определить, имеются ли в массиве подходящие элементы, можно различными способами. Наиболее простой из них – подсчитать количество подходящих элементов в массиве (см. пример 3.13). Если это количество окажется равным нулю, значит, в массиве нет подходящих элементов.

Другой способ – использовать некую "сигнальную" переменную, которая может принимать всего два значения, напри-

мер, **0** или **1** для переменной целого типа, "yes" или "no" (тип **string** – строка), **TRUE** или **FALSE** (логический тип **Boolean**). Такие служебные переменные в программировании называют "флажками". Одно из значений флажка (например, **0** или **FALSE**) соответствует случаю, когда в массиве нет подходящих элементов. Другое значение соответствует случаю, когда в массиве есть хотя бы один подходящий элемент.

Сначала следует предположить, что в массиве нет подходящих элементов и присвоить флажку соответствующее начальное значение (например, **0**). Затем с помощью счетного цикла (рис. 3.22) просматриваются по очереди элементы массива. Если будет обнаружен подходящий элемент, флажок "переключается" – ему присваивается второе из возможных значений, т.е., **1**. Если при дальнейшем просмотре будут обнаружены другие подходящие элементы, флажку будет снова присваиваться все то же значение **1**. Таким образом, после окончания цикла флажок будет иметь значение **0**, если флажок ни разу не переключался (не были обнаружены подходящие элементы), или **1**, если флажок был переключен хотя бы один раз.

Структура алгоритма (рис. 3.25) опять соответствует схеме алгоритма № 2 с выборочной обработкой подходящих элементов (в качестве определяемой характеристики массива выступает флажок – переменная **FLAG**):

**Пример 3.14.** Определить наличие отрицательных элементов с использованием числового флажка целого типа:

```
.....  
var FLAG: integer; {объявление переменной-флажка в  
разделе описаний}  
.....  
FLAG := 0;  
for J := 1 to N do  
  if A[J]<0 then  
    FLAG := 1;  
if FLAG=1 then  
  writeln (' в массиве есть отрицательные элементы')
```



```
else  
  writeln ('отрицательных элементов нет');
```

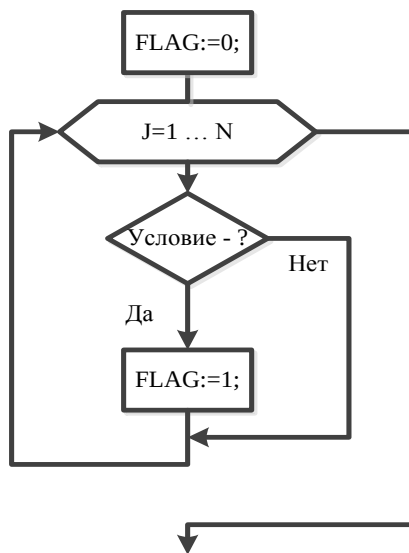


Рис. 3.25. Определение наличия в массиве подходящих элементов

**Пример 3.15.** Определить наличие четных элементов с использованием флажка логического типа:

```
.....  
var FLAG :boolean;  
  {объявление логической переменной-флажка в  
  разделе описаний}  
.....  
FLAG :=false;  
for J := 1 to N-1 do  
  if A[J] mod 2 = 0 then  
    FLAG :=true;  
if FLAG then  
  writeln (' в массиве есть четные элементы')  
else
```

**writeln ('четных элементов в массиве нет');**

.....

Обратите внимание, как используется переменная логического типа в условном операторе: **if FLAG then....** Здесь не требуется записывать равенство или неравенство, т.к. логическая переменная уже хранит ответ на воображаемый вопрос (условие), который можно было бы задать, например, так: "Верно ли, что в массиве есть четные элементы?" При использовании числового флажка условие выбора создается с помощью проверки равенства **FLAG=1**.

Надо отметить, что описанные выше способы определения наличия подходящих элементов неэкономичны, т.к. даже после обнаружения первого подходящего элемента программа все равно продолжает просмотр массива до конца. Если массив маленький, это не имеет особого значения. Однако на практике при решении задач моделирования в физике и технике приходится иметь дело с массивами огромных размеров, содержащими десятки и даже сотни тысяч элементов. В этом случае желательно прекращать просмотр массива сразу после обнаружения первого подходящего элемента. Эту задачу решает следующий алгоритм.

#### 5. Определение номеров первого (последнего) из подходящих элементов.

Поскольку заранее неизвестно, каким по счету окажется первый подходящий элемент, неизвестно и количество повторений цикла обработки (проверки) очередного элемента. Поэтому рассматриваемый алгоритм неудобно реализовать с помощью счетного цикла, у которого число повторений известно заранее. Логичнее будет применить один из циклов, выход из которых совершается не после определенного числа повторений, а при выполнении определенного условия, т.е., цикл с предусловием или с постусловием.

Алгоритмически схема поиска первого подходящего элемента выглядит так, как показано на рисунке 3.26.

Сначала программа берет первый элемент (**J:=1**) и проверяет для него заданное условие. Если оно выполнено, значит, искомый подходящий элемент найден и поиск можно прекра-

щать. Если условие не выполнено, программа переходит к следующему элементу ( **$J:=J+1$** ), и вновь возвращается к проверке условия. Проверка прекращается, когда для очередного элемента условие окажется выполненным, поэтому после ее окончания значение номера искомого элемента будет сохранено в переменной **J**.

Такой алгоритм очень похож на организацию цикла с предусловием. Однако в цикле с предусловием повторение цикла совершается при выполнении условия, а в алгоритме, показанном на рисунке 3.26 – при его нарушении. Чтобы использовать цикл с предусловием, надо изменить логическое значение, соответствующее поставленному для элемента цикла условию, на противоположное. Для этого можно:

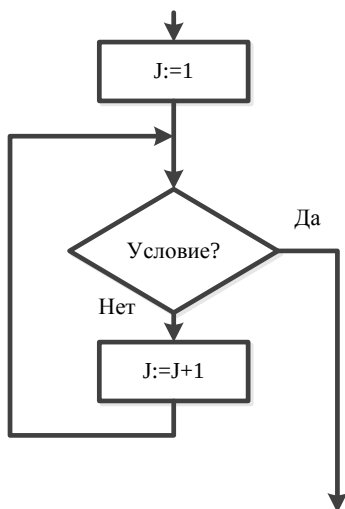


Рис. 3.26. Поиск первого подходящего элемента

а) применить логическое отрицание, т.е., записать в заголовке цикла: **not (условие)**;

б) просто заменить условие на противоположное.

**Пример 3.16.** Поиск первого отрицательного элемента можно провести следующим образом:

```
.....  
J:=1;  
while not (A[J]<0) do  
  J:=J+1;  
.....
```

или так:

```
.....  
J:=1;  
while (A[J]>=0) do  
  J:=J+1;  
.....
```

Описанная схема имеет один недостаток – она работает только в том случае, когда в массиве обязательно есть хотя бы один подходящий элемент. Если же нужных элементов нет, программа последовательно переберет все элементы массива вплоть до **A[N]**, а затем выполнит очередной переход **J:=J+1** и попытается найти следующий элемент (**A[N+1]**), которого не существует. Такое действие приводит к ошибке процесса выполнения № 201 (**Range check error**) и выполнение программы прекращается.

Необходимо обеспечить выход из цикла не только при обнаружении подходящего элемента, но и в случае, когда проверен последний элемент, а подходящий так и не был найден. Т.е., выполнение цикла необходимо прекратить в одном из двух случаев:

- а) найден подходящий элемент;
- б) значение индекса **J** вышло за границы диапазона **[1 .. N]**.

Блок-схема алгоритма показана на рисунке 3.27.

В заголовке цикла создано сложное условие, с использованием операции дизъюнкции **or**. При этом, как и на схеме рисунка 3.25, показанный цикл не соответствует схеме цикла с предусловием, т.к. повторение происходит при невыполнении поставленного условия.

Чтобы использовать здесь цикл с предусловием, надо заменить имеющееся условие на его отрицание. Это можно сделать, используя логическое отрицание **not**, или же применить логическое правило де Моргана (см. § 1.4).

Согласно этому правилу, логическое выражение (для примера рассмотрен поиск первого отрицательного элемента)

**not( (A[J]<0) or (J>N) )**

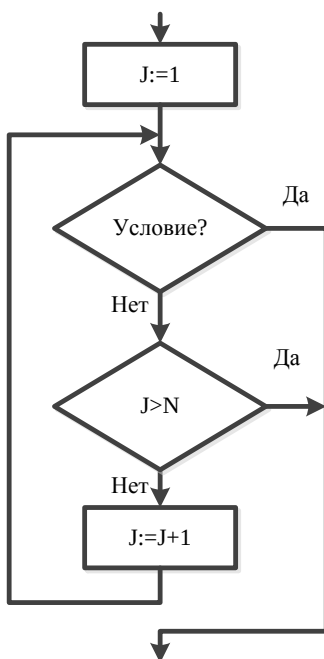


Рис. 3.27. Поиск первого подходящего элемента с проверкой их наличия в массиве

эквивалентно высказыванию

**(A[J]>=0) and (J<=N)**

Если после завершения цикла переменная **J** окажется больше, чем число элементов массива **N**, это будет означать, что в массиве не было найдено подходящих элементов.

В рассмотренном выше примере поиск будет теперь выглядеть так (после завершения цикла выводится на экран либо номер найденного элемента, либо сообщение о том, что подходящих элементов в массиве не обнаружено):

**Пример 3.17:**

```
.....  
J:=1;  
while (A[J]>=0) and (J<=N) do  
  J:=J+1;  
if J<=N then  
  writeln('номер первого подходящего элемента: ',  
          J:3)  
else  
  writeln('подходящих элементов в массиве нет');  
.....
```

В таком виде алгоритм можно использовать и для проверки наличия в массиве подходящих элементов, причем для больших массивов он экономичнее алгоритма № 4.

Если надо найти в массиве последний подходящий элемент, инициализация переменной **J** имеет вид **J:=N**, т.е., поиск начинается с последнего элемента массива. Далее при поиске совершаются переходы к предшествующим элементам: **J:=J-1**; в заголовке цикла должно стоять не условие **J<=N**, а условие **J>=1**.

**6. Поиск максимального (минимального) среди подходящих элементов.**

Эта задача отличается от алгоритма № 3 двумя обстоятельствами:

а) при поиске максимального элемента в массиве к условию поиска **A[J] > MAX** (см. алгоритм № 3) следует добавить условие отбора подходящих элементов;

б) поиск следует начинать не с первого элемента массива, а с первого подходящего элемента. Поэтому перед началом поиска следует найти в массиве первый подходящий элемент с помощью алгоритма № 5.

Если первый подходящий элемент массива имеет номер **NOM**, то алгоритм поиска максимального подходящего элемента имеет вид, показанный на рис. 3.28 (в блок-схеме **NOM** – номер первого подходящего элемента, **NOM1** – номер максимального среди подходящих элементов):

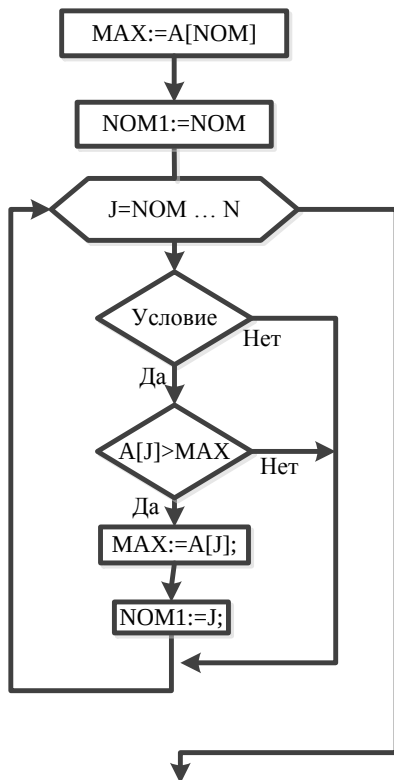


Рис. 3.28. Поиск максимального подходящего элемента

**Пример 3.18.** Поиск максимального четного элемента и определение его номера (в данном примере не используется переменная **NOM1**; найденный номер первого подходящего элемента **J** сразу запоминается в переменной **NOM**, которая в данном случае должна хранить искомый номер максимального подходящего элемента).

Текст фрагмента программы:

```
.....  
J:=1;  
while (A[J] mod 2 = 1) and (J<=N) do  
  J:=J+1;  
if J<=N then  
  begin  
    NOM=J;  
    MAX := A[J];  
    for J := NOM+1 to N do  
      if (A[J] mod 2 = 0) and (A[J]>MAX) then  
        begin  
          MAX := A[J];  
          NOM := J;  
        end;  
    end  
  else  
    writeln('подходящих элементов в массиве нет');  
  .....  
end
```

7. Фильтрация (создание нового массива, в который входят только подходящие элементы).

В этом случае результатом решения является новый массив (дадим ему имя **В**). Число его элементов равно числу подходящих элементов исходного массива **А**, и его можно, разумеется, подсчитать (см. алгоритм № 2), но во время объявления массива **В** оно еще неизвестно. Известно только, что в массиве **В** может оказаться не меньше элементов, чем в массиве **А** (если все элементы массива **А** окажутся подходящими). Поэтому при объявлении массивов **А**, **В** для них указывается одинаковое число элементов – **N**.

Далее, необходимо ввести новую целочисленную переменную (обозначим ее **К**), которая будет указывать, сколько элементов из массива **А** занесено в массив **В**.

Идея алгоритма основана на схеме выборочной обработки подходящих элементов (см. рис. 3.22). Сначала предполагается,



что в исходном массиве нет подходящих элементов ( $K:=0$ ). Затем программа по очереди просматривает элементы исходного массива (это делается с помощью обычного счетного цикла). Когда обнаруживается подходящий элемент, выполняются следующие действия по его обработке:

- а) переменная **K** увеличивается на 1;
- б) элементу **V[K]** присваивается значение обнаруженного подходящего элемента **A[J]**;

При этом элемент **V[K]** всякий раз оказывается последним среди заполненных реальными значениями элементов массива **V**. Остальные элементы массива **V** остаются неопределенными и какое-то использование их в программе не предусматривается.

Блок-схема алгоритма показана на рисунке 3.29.

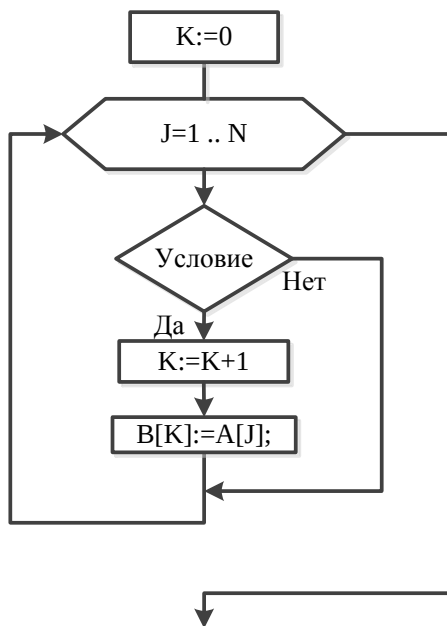


Рис. 3.29. Фильтрация одномерного массива

**Пример 3.19.** Выбор в новый массив отрицательных элементов массива **A**.

```
.....  
K := 0;  
for J := 1 to N do  
  if A[J]<0 then  
    begin  
      K := K+1;  
      B[K]:=A[J];  
    end;  
  .....  
.....
```

8. Сортировка массива – перестановка его элементов в порядке возрастания или убывания.

Задача сортировки чрезвычайно важна и часто встречается на практике, поэтому существует большое количество алгоритмов сортировки массивов. Отдельные алгоритмы оказываются эффективными в разных ситуациях – для больших массивов, для массивов, имеющих большое количество одинаковых элементов, и т.п.

Метод "пузырька" – эффективный и простой метод сортировки небольших массивов. Принцип его заключается в следующем (для примера рассмотрим сортировку по возрастанию). Программа просматривает попарно элементы массива: первый и второй, второй и третий и т.д. Каждый раз, когда последующий элемент оказывается больше предыдущего, программа меняет их местами. Нетрудно понять, что при этом большой по значению элемент будет постепенно перемещаться в направлении конца массива, пока не встретит на пути элемент, еще больший, затем передвигаться будет этот больший элемент и т.д. После проверки всех пар, вплоть до пары элементов с номерами (**N-1**) и **N** (легко подсчитать, что таких пар всегда на 1 меньше, чем самих элементов, так что на первом этапе проверяется **N-1** пар), самый большой элемент массива встанет на положенное для него – последнее – место. Затем подобным образом проверяют оставшиеся (**N-1**) элементов, затем (**N-2**) и так далее, пока не останется только одна пара элементов: первый и второй. Переста-

вив их в нужном порядке, получают массив, в котором все элементы выстроены в порядке возрастания.

Алгоритм осуществляется с помощью вложенных циклов (рис. 3.30).

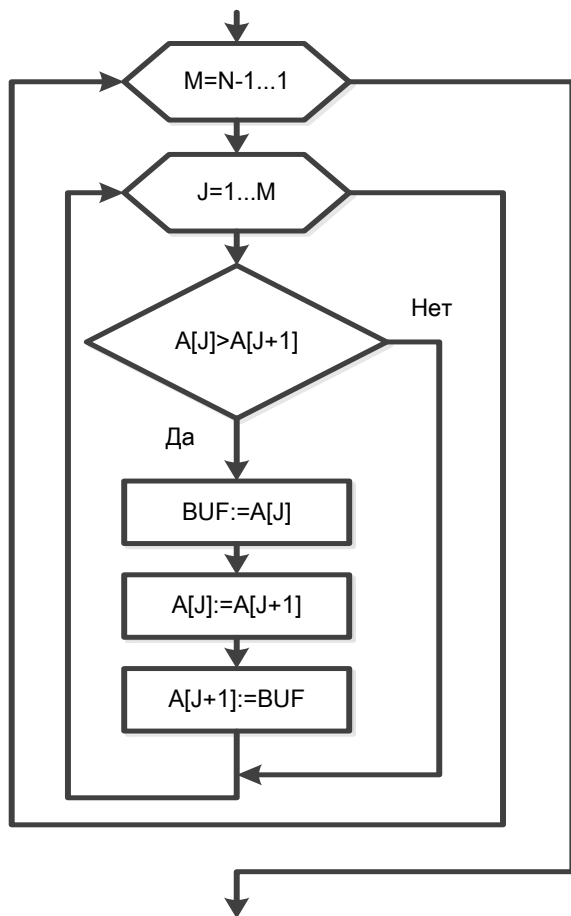


Рис. 3.30. Сортировка одномерного массива методом пузырька

Внутренний цикл – это цикл проверки некоторого количества (**M**) пар элементов с перестановкой их при необходимости. Внешний цикл позволяет постепенно уменьшать значение **M** от

(**N-1**) до **1**. В качестве внешнего цикла использован счетный цикл с уменьшением значения счетчика:

**for M:=N-1 downto 1 do... .**

Для того, чтобы поменять местами значения элементов в двух ячейках памяти, в данном алгоритме используется буферная переменная (ячейка памяти), как описано выше при рассмотрении примера оформления процедуры **OVMEN(AF,BF)** (см. § 3.11).

**Пример 3.20.** Сортировка массива по возрастанию:

```
.....  
for M := N-1 downto 1 do  
  for J := 1 to M do  
    if A[J] > A[J+1] then  
      begin  
        BUF := A[J] ;  
        A[J] := A[J+1] ;  
        A[J+1] := BUF ;  
      end;
```

.....

Если описать в программе процедуру **OVMEN(AF, BF)**, можно упростить запись алгоритма сортировки:

```
.....  
for M := N-1 downto 1 do  
  for J := 1 to M do  
    if A[J] > A[J+1] then  
      OVMEN(A[J], A[J+1]);  
.....
```

В заключение рассмотрим пример программы, решающей более сложную задачу.

**Пример 3.21.** Даны: натуральное число, действительные числа  $X_1, X_2, \dots, X_N$ . Вычислить значение  $Q=(1+R)/(1+S)$ , где  $R$  – среднее арифметическое всех тех членов последовательности

$X_1, X_2, \dots, X_N$ , которые меньше 10, а  $S$  – среднее арифметическое ее членов, превосходящих 10.

#### I. Постановка задачи.

а) Формализация.

Среднее арифметическое нескольких чисел равно их сумме, поделенной на их количество. Следовательно, для решения задачи необходимо последовательно найти:

а) количество и сумму тех членов последовательности, которые меньше 10 (обозначим эти величины **SUMR** и **KOLR**);

б) среднее арифметическое таких членов – **R**;

в) количество и сумму тех членов последовательности, которые больше 10 (обозначим эти величины **SUMS** и **KOLS**);

г) среднее арифметическое таких членов – **S**;

д) искомую величину **Q**.

Члены последовательности, равные 10, по условию задачи следует пропускать и не обрабатывать.

В общем случае (раз в задаче не идет речь о свойстве кратности элементов массива целому числу), элементы массива рассматриваются, как вещественные числа. К этому же типу, очевидно, должны принадлежать суммы каких-либо членов, т.е., **SUMR** и **SUMS**, а также величины **R**, **S**, **Q**. Количества элементов **KOLR**, **KOLS** должны быть целыми числами.

Переменные **KOLR**, **KOLS**, **SUMR** и **SUMS** вычисляются с помощью типового алгоритма № 2. Формулы для вычисления остальных переменных:

$$R = \frac{SUMR}{KOLR}, \quad S = \frac{SUMS}{KOLS}, \quad Q = \frac{1+R}{1+S}$$

б) Спецификация данных для нашей задачи выглядит так:

| Имя         | Вид        | Тип            | Смысл                                  | Роль в программе |
|-------------|------------|----------------|----------------------------------------|------------------|
| <b>N</b>    | Константа  | <b>integer</b> | Число элементов массива                | Исходная         |
| <b>X</b>    | Переменная | <b>real</b>    | Исходный массив                        | Исходная         |
| <b>Q</b>    | Переменная | <b>real</b>    | Искомая величина                       | Результат        |
| <b>KOLR</b> | Переменная | <b>integer</b> | Количество элементов, меньших 10       | Вспомогательная  |
| <b>KOLS</b> | Переменная | <b>integer</b> | Количество элементов, больших 10       | Вспомогательная  |
| <b>SUMR</b> | Переменная | <b>real</b>    | Сумма элементов, меньших 10            | Вспомогательная  |
| <b>SUMS</b> | Переменная | <b>real</b>    | Сумма элементов, больших 10            | Вспомогательная  |
| <b>R</b>    | Переменная | <b>real</b>    | Среднее значение элементов, меньших 10 | Вспомогательная  |
| <b>S</b>    | Переменная | <b>real</b>    | Среднее значение элементов, больших 10 | Вспомогательная  |

## II. Алгоритмизация задачи.

Решение сводится к последовательному применению алгоритмов поиска суммы и количества элементов, удовлетворяющих условиям  $X[J] < 10$  и  $X[J] > 10$  (варианты алгоритма 2, см. примеры 3.5, 3.6). Затем вычисляются средние арифметические **R**, **S** и результат **Q**.

Блок-схема программы представлена на рисунках 3.31, 3.32.

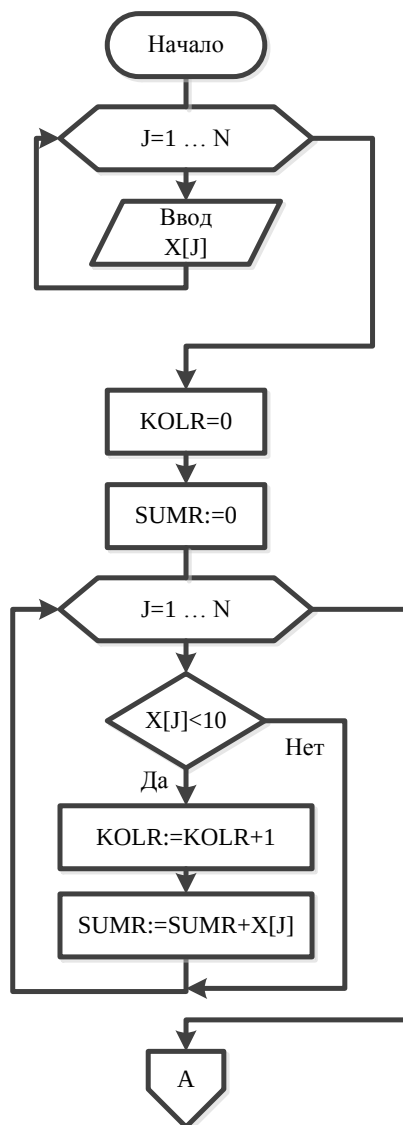


Рис. 3.31. Блок-схема программы (начало)

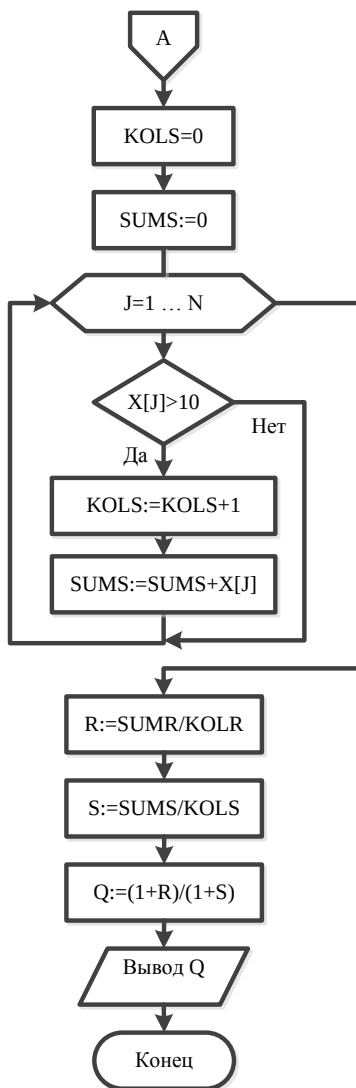


Рис. 3.32. Блок-схема программы (окончание)



**III.      Текст программы:**

```
program MASSSIV;  
uses CRT;  
const N=10;  
var  
  J, KOLR, KOLS: integer;  
  SUMR, SUMS, R, S, Q: real;  
  X: array [1..N] of real;  
Begin  
  clrscr; {Очистка экрана}  
  
  {Ввод исходных данных:}  
  
  for J := 1 to 5 do  
    begin  
      write ('Введите значение элемента X [ ', J:3, ' ]: ');  
      readln( X [ J ] );  
    end;  
  
  {Окончание ввода исходных данных}  
  
  {Расчет суммы и количества элементов, меньших 10:}  
  
  KOLR:=0;  
  SUMR:=0;  
  for J:=1 to N do  
    if X[J]<10 then  
      begin  
        KOLR:=KOLR+1;  
        SUMR:=SUMR+X[J];  
      end;  
  
  {Расчет суммы и количества элементов, больших 10:}  
  
  KOLS:=0;  
  SUMS:=0;  
  for J:=1 to N do
```

```
if X[J]>10 then
begin
  KOLS:=KOLS+1;
  SUMS:=SUMS+X[J];
end;

{Расчет средних арифметических:}

R:=SUMR/KOLR;
S:=SUMS/KOLS;

{Вычисление результата:}

Q:=(1+R)/(1+S);

{Вывод результата:}

writeln('Результат вычислений Q= ', Q:8:3);
readln;
End.
```

Задания для самостоятельного выполнения:

Замечание: в каждой программе должен быть предусмотрен контрольный вывод на экран исходного массива.

**I. Задачи на применение типового алгоритма с простым условием отбора подходящих элементов.**

- 1) В массиве чисел **A[1 .. N]** найти минимальное по абсолютной величине число и его номер.
- 2) Для массива чисел **A[1 .. N]** найти произведение элементов, попадающих в заданный отрезок **[C, D]**.
- 3) Для массива чисел **A[1 .. N]** найти сумму тех элементов, которые больше первого числа **A1**.
- 4) Для массива чисел **A[1 .. N]** найти количество отрицательных чисел.
- 5) В массиве **A[1 .. N]** отрицательные числа заменить нулями.

### *Глава 3. Основы программирования на языке ПАСКАЛЬ*

6) В массиве чисел  $A[1 \dots N]$  изменить знак отрицательных чисел на противоположный.

7) В массиве чисел  $A[1 \dots N]$  найти первое число, которое больше заданного числа  $P$ .

8) В массиве чисел  $A[1 \dots N]$  найти последнее отрицательное число.

9) Дан массив  $A[1 \dots N]$ . Элементы, стоящие на чётных местах, разделить на 3, а стоящие на нечётных местах – умножить на 2.

10) В массиве чисел  $A[1 \dots N]$  найти последнее число, меньшее по абсолютной величине заданного числа  $L$ .

11) Из массива  $A[1 \dots N]$  получить массив  $B$ , в который войдут элементы, попадающие в интервал  $(C; D)$ .

12) Из массива  $A[1 \dots N]$  получить массив  $B$ , в который войдут только отрицательные элементы массива  $A$ .

13) Преобразовать массив  $A[1 \dots N]$  по правилу: если  $A_i > 0$ , то увеличить его в 10 раз, иначе - заменить нулем.

14) Найти номер первого четного элемента массива  $A[1 \dots N]$ .

15) Найти сумму элементов массива  $A[1 \dots N]$ , не принадлежащих отрезку  $[2; 7]$ .

16) Заменить на единицу все элементы массива  $A[1 \dots N]$ , не входящие в отрезок  $[-2; 2]$ .

17) Заменить нулями элементы массива  $A[1 \dots N]$ , которые не являются нечетными числами.

18) Найти сумму тех элементов массива  $A[1 \dots N]$ , которые удовлетворяют условию  $|A_i| < i^2$ .

19) Найти и вывести на экран те элементы массива  $A[1 \dots N]$ , которые являются удвоенными четными числами.

20) Получить удвоенную сумму всех положительных элементов массива  $A[1 \dots N]$ .

21) Для массива чисел  $A[1 .. N]$  найти сумму отрицательных чисел.

22) Для массива чисел  $A[1 .. N]$  найти количество чисел, равных заданному числу  $D$ .

23) Все отрицательные элементы массива  $A[1 .. N]$  увеличить на 5.

24) Произвести сортировку массива  $A[1 .. N]$  в порядке убывания его элементов.

25) Программа должна ответить на вопрос: верно ли, что наибольший элемент массива  $A[1 .. N]$  по модулю больше 10?

#### **II. Задачи на применение типового алгоритма со сложным условием отбора подходящих элементов.**

26) Для массива чисел  $A[1 .. N]$  найти сумму положительных элементов, имеющих чётные номера.

27) Для массива чисел  $A[1 .. N]$  найти сумму квадратов чисел, стоящих на нечётных местах

28) Определить количество элементов массива  $A [1 .. N]$ , имеющих четные номера и являющихся четными числами.

29) Дан массив натуральных чисел  $A[1 .. N]$ . Определить количество элементов, имеющих четные порядковые номера и являющихся нечетными числами.

30) Найти сумму тех нечетных элементов массива  $A[1 .. N]$ , которые меньше 10.

31) Определить количество четных элементов массива  $A [1 .. N]$ , попадающих в отрезок  $[10; 20]$ .

32) Найти максимальный четный элемент массива  $A [1 .. N]$ , не превосходящий 50.

33) Найти максимальный из элементов массива  $A[1 .. N]$ , кратных пяти.

34) Найти наименьшее из четных чисел в массиве  $A[1 .. N]$ .

35) Найти максимальный из элементов массива  $A[1 \dots N]$ , стоящих на нечетных местах.

36) Для массива  $A[1 \dots N]$  найти сумму чисел до первого, имеющего ненулевую дробную часть.

37) В массиве чисел  $A[1 \dots N]$  найти минимальный элемент и его номер среди элементов, кратных 3.

38) Для массива чисел  $A[1 \dots N]$  найти произведение положительных элементов, имеющих нечётные номера.

39) Для массива чисел  $A[1 \dots N]$  найти количество положительных элементов, имеющих чётные номера.

40) Определить количество нечетных элементов массива  $A[1 \dots N]$ , попадающих в отрезок  $[5; 10]$ .

41) Для массива чисел  $A[1 \dots N]$  найти количество четных чисел, превышающих заданное число  $D$ .

42) Все положительные четные элементы массива  $A[1 \dots N]$  увеличить на 10.

43) В массиве чисел  $A[1 \dots N]$  найти первое нечетное число, которое больше заданного числа  $P$ .

44) Выяснить, верно ли, что в массиве  $A[1 \dots N]$  имеются три идущих подряд нулевых элемента.

45) Найти сумму положительных элементов массива  $A[1 \dots N]$ , не принадлежащих отрезку  $[5; 15]$ .

46) Найти произведение нечетных элементов массива  $A[1 \dots N]$  с четными номерами.

47) В массиве  $A[1 \dots N]$  содержится четное число элементов. Определить среднее арифметическое для элементов, стоящих на четных местах.

48) Определить сумму положительных элементов массива  $A[1 \dots N]$ , не превышающих 5.

49) Определить количество четных элементов массива  $A[1 \dots N]$ , не превышающих 10 по абсолютной величине.

50) Программа должна вывести на экран ответ на вопрос: "Верно ли, что первый отрицательный элемент массива по абсолютной величине превышает 5?"

#### **III. Задачи, требующие видоизменения или комбинирования типовых алгоритмов**

51) Для массива **A [1 .. N]** получить массив **B [1 .. N]** по правилу:

$$B_i = \begin{cases} 1 & \text{если } |A_i| > 3; \\ 0 & \text{в противном случае.} \end{cases}$$

52) Для массива **A [1 .. N]** получить массив **B [1 .. N]** по правилу:

$$B_i = \begin{cases} A_i^2 & \text{если } A_i \text{ кратно } 3; \\ A_i / 3 & \text{в противном случае.} \end{cases}$$

53) В массиве **A[1 .. N]** уменьшить все элементы на их среднее арифметическое.

54) Для массива чисел **A[1 .. N]** найти среднее гармоническое его элементов по формуле

$$\frac{1}{A_{\text{ср геом}}} = \frac{1}{N} \left( \frac{1}{A_1} + \frac{1}{A_2} + \dots + \frac{1}{A_N} \right)$$

55) Для массива чисел **A[1 .. N]** найти среднее квадратическое его элементов по формуле

$$A_{\text{ср кв}} = \sqrt{\frac{1}{N} (A_1^2 + A_2^2 + \dots + A_N^2)}$$

56) Получить новый массив **B[1 .. ]** из исходного массива **A[1 .. N]** удвоением всех четных элементов. Нечетные элементы в массив не включать.

57) Для массива чисел **A[1 .. N]** найти среднее геометрическое его элементов по формуле

$$A_{\text{ср геом}} = \sqrt[N]{A_1 \cdot A_2 \cdot \dots \cdot A_N}$$

58) Дан массив  $A[1 \dots N]$ . Получить  $R = \min(A_2, A_4, \dots) + \max(A_1, A_3, \dots)$ .

59) Дан массив  $A[1 \dots N]$ . У всех отрицательных элементов массива, стоящих после первого положительного, изменить знак на противоположный.

60) Выяснить, какое число встречается в массиве  $A[1 \dots N]$  раньше – положительное или отрицательное (замечание: несколько первых элементов массива могут оказаться равными нулю).

61) Дан массив  $A[1 \dots N]$ . Проверить, какое из двух чисел больше по абсолютной величине:

- а) наибольший из нечетных элементов;
- б) наименьший из четных элементов.

62) Заменить все элементы массива чисел  $A[1 \dots N]$ , большие 7, количеством таких элементов.

63) Заменить наименьший элемент массива  $A[1 \dots N]$  средним арифметическим всех его членов (если таких элементов несколько, то заменить все).

64) Дан массив  $A[1 \dots N]$ . Получить новый массив по правилу

$$B_i = \begin{cases} 1, & \text{если } A_i > \max(A_1, A_2, A_3, A_4), \\ 0 & \text{в противном случае.} \end{cases}$$

65) Для массива  $A[1 \dots N]$  получить массив  $B[1 \dots N]$  по правилу

$$B_i = \begin{cases} A_i & \text{при } 0 < A_i < 10, \\ 0 & \text{в противном случае.} \end{cases}$$

66) Дан массив  $A[1 \dots N]$ . Получить новый массив  $B_i = i - A_i$  и подсчитать количество нулевых элементов нового массива.

67) Найти наибольшее из последовательности чисел

$$A_k = k \cdot e^{\sin(k+1)} \quad (k = 1, 2, \dots, N)$$

68) В массиве **A[1 .. N]** определить число соседств двух чисел одного знака, причем модуль первого числа должен быть больше модуля второго числа.

69) Найти номер последнего нечетного члена массива **A[1 .. N]**; если нечетных членов нет, то ответом должно быть число **N+1**.

70) Требуется умножить все элементы массива **A[1 .. N]** на квадрат его наименьшего члена, если **A1 ≥ 0**, и на квадрат его наибольшего члена, если **A1 < 0**.

71) В массиве **A[1 .. N]** определить число соседств двух положительных чисел и двух чисел разного знака. Какое число больше?

72) Из массива **A[1 .. N]** получить новый массив **B[1 .. ]**, в котором будут выстроены в порядке возрастания все отрицательные элементы исходного массива.

73) Составить массив из положительных значений выражения

$$A_k = \sin(k+1) \cdot e^{\sin(k+1)} \quad (k = 1, 2, \dots, N)$$

74) В массиве **A[1 .. N]** определить значение второго по величине элемента.

**Замечание:** Для решения этой задачи рекомендуется произвести сортировку исходного массива.

75) Массив **A[1 .. N]** состоит из нечетного количества несовпадающих элементов. Определить значение "среднего" элемента, т.е., такого, что число элементов, превосходящих его по величине, равно числу элементов, не превосходящих его.

**Замечание:** Для решения этой задачи рекомендуется произвести сортировку исходного массива.



## ЛИТЕРАТУРА

1. Информатика. Базовый курс / Под ред. С.В.Симоновича. – СПб.: Питер, 2005. – 640 с.
2. Брукшир, Дж. Г. Введение в компьютерные науки. – М.,СПб.: Вильямс, 2001. – 688 с.
3. Сырецкий, Г.А. Информатика. Фундаментальный курс. – Т. 1. – Основы информатики и вычислительной техники. – СПб.: БХВ-Санкт-Петербург, 2005. – 832 с.
4. Гурский, Д.А. Вычисления в MathCAD / Д.А.Гурский. – Минск: Новое знание, 2003. – 814 с.
5. Плис А.И., Сливина Н.А. MathCAD 2000: Математический практикум для экономистов и инженеров. – М.: Финансы и статистика, 2000. – 256 с.
6. Культин, Н.Б. TurboPascal в задачах и примерах / Н.Б.Культин.- СПб.: БХВ-Петербург, 2006.-256с.
7. Фаронов, В.В. Турбо Паскаль 7.0. Начальный курс. Учебное пособие / В.В.Фаронов. – М.: Нолидж, 2001. – 576 с.
8. Немнюгин С.А. Turbo Pascal. – СПб.: Питер, 2001. – 496 с.

УЧЕБНОЕ ИЗДАНИЕ

**Крюкова** Татьяна Петровна  
**Печерских** Ирина Александровна  
**Романова** Вера Васильевна,  
**Семенов** Андрей Германович  
**Столетова** Елена Александровна  
**Яковлева** Любовь Анатольевна

**ИНФОРМАТИКА.  
ТЕОРИЯ, ВЫЧИСЛЕНИЯ, ПРОГРАММИРОВАНИЕ**

Учебное пособие для практических и лабораторных работ

Для студентов вузов

Редактор ...  
Технический редактор ...  
Художественный редактор ...

ЛР № 020524 от 02.06.97  
Подписано в печать ... Формат 60×84<sup>1/16</sup>  
Бумага офсетная. Гарнитура Times.  
Уч.-изд. л. 14,2. Тираж 400 экз.  
Заказ № ...

Оригинал-макет изготовлен в редакционно-издательском отделе  
Кемеровского технологического института пищевой промышленности  
650056, г. Кемерово, 6-р Строителей, 47

ПЛД №44-09 от 10.10.99  
Отпечатано в лаборатории множительной техники  
Кемеровского технологического института пищевой промышленности  
650002, г. Кемерово, ул. Институтская, 7