

ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ОБРАЗОВАНИЮ
Государственное образовательное учреждение
высшего профессионального образования
«Пензенский государственный университет»

Д. В. Такташкин, А. Н. Ракова, Л.В. Гурьянов

Информатика. Язык программирования Turbo Pascal.

Методические указания к лабораторным работам
по программированию на языке Turbo Pascal



Пенза
Издательство
Пензенского государственного
университета
2009



Рецензенты:

Такташкин, Д. В.

Ш55 Информатика. Язык программирования
Pascal : Методические указания к лабораторным работам
по программированию на языке Turbo Pascal /
Д. В. Такташкин, А. Н. Ракова, Л. В. Гурьянов. – Пенза :
Изд-во Пенз. гос. ун-та, 2009. – 85 с.

Рассматриваются элементы диалоговой среды Turbo Pascal, структура программ на языке Turbo Pascal, основные типы данных, преобразования типов данных и действия над ними, операторы языка, работа с массивами данных, работа с процедурами и функциями. Приведены примеры реализации программ. Предложены задания для выполнения лабораторных работ.

Учебное пособие подготовлено на кафедре «Математическое обеспечение и применение ЭВМ» и предназначено для изучения курса информатики.

УДК 681.3

© Такташкин Д. В., Ракова А. Н., Гурьянов Л. В.,
2009

© Издательство Пензенского государственного
университета, 2009

ОГЛАВЛЕНИЕ

Введение.....	4
Общие сведения. Пакет программ Turbo Pascal.....	5
Лабораторная работа № 1. Программирование алгоритмов линейной структуры.....	8
Лабораторная работа № 2. Программирование алгоритмов разветвленной структуры.....	16
Лабораторная работа № 3. Организация циклов.....	24
Лабораторная работа № 4. Работа с одномерными массивами.....	39
Лабораторная работа № 5. Работа с двумерными массивами.....	50
Лабораторная работа № 6. Процедуры и функции	61
Лабораторная работа № 7. Файлы и их применение в Turbo Pascal	75
Рекомендованная литература	86

Введение

В настоящее время средства вычислительной техники используются практически во всех сферах деятельности человека, и овладение навыками программирования этих вычислительных средств на одном из языков высокого уровня является обязательным элементом образования и культуры каждого инженера.

Наибольшее распространение находят языки программирования, имеющие высокий уровень формализации и позволяющие создавать хорошо структурированные программы, предназначенные для решения обширного класса задач из самых разнообразных сфер инженерной деятельности. К таким языкам относятся Turbo Pascal, C/C++ , Modula, Ada и т. д.

Одним из простых для первоначального изучения, но в тоже время достаточно универсальным языком программирования является Turbo Pascal. Данный язык, названный в честь французского математика и философа Блеза Паскаля (1623-1662), был создан как учебный язык программирования в 1968-1971 годах швейцарским ученым Никласом Виртом на кафедре информатики Стэнфордского университета (Цюрих). В настоящее время Turbo Pascal имеет более широкую сферу применения, чем предусматривалось при его создании. Этот язык отличается простотой понимания, стройностью и структурностью алгоритмов, быстротой компилятора и удобными средствами создания и отладки программ.

В настоящий момент пользуются популярностью такие версии языка как TMT Pascal, Free Pascal и GNU Pascal. Продолжает использоваться и Turbo Pascal. Развитием языка Turbo Pascal является Object Pascal – версия языка Паскаль расширенная средствами объектно-ориентированного программирования. Последние версии Turbo Pascal лежат в основе среды программирования Delphi.

Общие сведения.

Пакет программ Turbo Pascal

Пакет программ Turbo Pascal содержит среды для разработки программ версий под DOS и под Windows. В пакет программ входят программы компиляции, отладки, а также дополнительные утилиты.

Программная среда Turbo Pascal

Для запуска среды Turbo Pascal необходимо запустить **bp.exe** из папки **BIN**.

Верхняя строка представляет собой меню системы, доступ к которому осуществляется мышью или нажатием клавиши **F10**. Используя мышь или клавиши курсора пользователь может выбрать нужный пункт меню. Некоторые команды меню доступны через «горячие клавиши». Например, для выбора пункта меню **File**→**Open** достаточно нажать клавишу **F3**.

Текст программы набирается в окне редактора (см. рисунок 1), на краях которого представлены следующие элементы:

- имя файла программы на диске;
- элементы управления окном мышью;
- полосы прокрутки;
- координаты положения курсора в тексте программы.

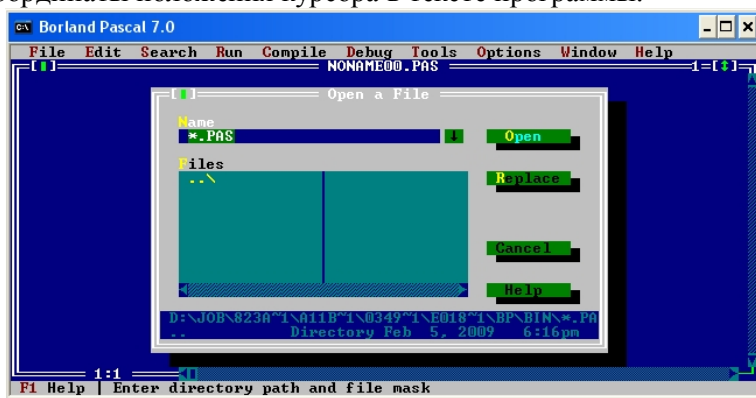


Рис. 1 – Интерфейс среды Turbo Pascal

Меню среды Turbo Pascal

Основное меню среды содержит следующие команды:

File – набор команд для работы с файлами (открытие, сохранение программы);

Edit – набор команд для работы с текстом;
Search – команды поиска по тексту программы;
Run – команды, управляющие выполнением программы;
Compile – команды компиляции программы, т.е. получение исполняемого файла;
Debug – команды отладки;
Tools – дополнительные утилиты;
Options – настройки среды;
Window – команды навигации между открытыми программами;
Help – помощь.

Для использования помощи можно воспользоваться сочетанием клавиш **Ctrl+F1** – помощь по ключевому слову, **Alt+F1** – предыдущая справка.

Сохранение программы на диске

Набранный текст программы можно сохранить на диске, нажав клавишу **F2** или выбрав пункт меню **File**→**Save**. Появится окно сохранения программы (см. рисунок 2).

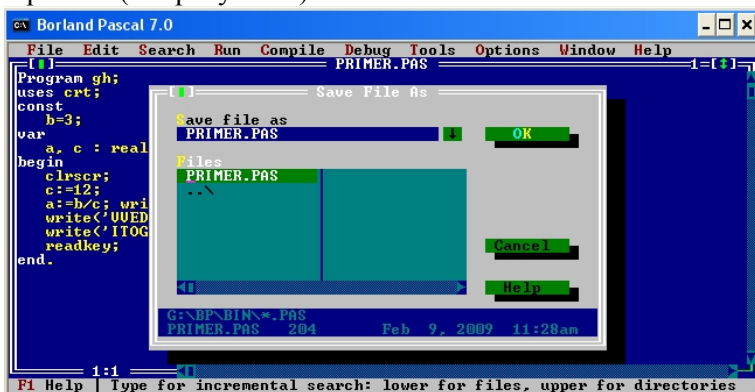


Рис. 2 – Сохранение программы в среде Turbo Pascal

В появившемся окне следует выбрать каталог и ввести имя файла программы, под которым она будет сохранена на диске и нажать **OK**.

Запуск и отладка программы

Для запуска программы необходимо ее скомпилировать, т.е. получить исполняемый (**exe**) файл. Это делается командой меню **Compile**→**Make** или клавишей **F9**. Запуск программы – командой меню **Run**→**Run** (сочетание клавиш **Ctrl+F9**).

В процессе компиляции программы выявляются синтаксические ошибки. При наличии таких ошибок среда сообщает тип ошибки и указывает место в тексте программы, где она присутствует. В случае затруднения пользователь может посмотреть помощь через пункт меню **Help**→**Topic Search** или используя сочетание клавиш **Ctrl+F1**.

В программах сложной структуры для выявления ошибки используется отладка. Используя сочетание клавиш **Ctrl+F8**, на строке с курсором устанавливается контрольная точка (**Breakpoint**). В процессе выполнения программы на контрольной точке будет произведена остановка выполнения программы и переход в режим ее отладки. Возможно два вида выполнения команд строки: **F7** – выполнение команд подпрограмм пошагово и **F8** – выполнение команд подпрограмм за один шаг.

На любом шаге возможен просмотр значений переменных: при нажатии **Ctrl+F7** появляется окно, в котором следует ввести имя интересующей переменной. При нажатии **OK** в отдельном окне появится имя и значение этой переменной. При выполнении программы изменения значений выбранных переменных автоматически обновляются (см. рисунок 3).

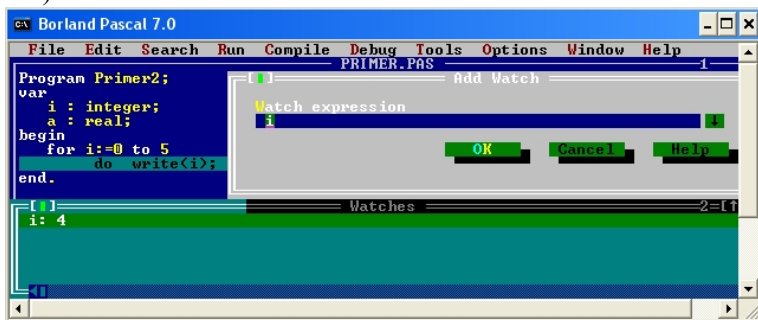


Рис. 3 – Отладка программы в среде Turbo Pascal

При нажатии клавиши **F9** выполнение программы будет продолжено. Для просмотра результата работы программы следует использовать сочетание клавиш **Alt+F5**. Для прекращения работы программы следует использовать сочетание клавиш **Ctrl+F2**.

Для завершения работы со средой Borland Pascal 7.0 следует воспользоваться пунктом меню **File**→**Exit** или сочетанием клавиш **Alt+X**.

Лабораторная работа № 1.

Программирование алгоритмов линейной структуры

Цель работы: изучение структуры программы на языке Turbo Pascal; типов данных; составление простых неразветвленных программ.

Теоретические сведения

Программа на языке Turbo Pascal состоит из нескольких блоков. Порядок следования блоков определяется использованием их друг другом.

Блок объявления констант начинается служебным словом **const**. После чего указывается имя константы, знак равенства и ее значение. В Turbo Pascal допускается использование констант в виде выражений. Пример объявления констант на языке Turbo Pascal:

```
const
  a=5;
  b=7/(4+2.6);
  c='this is constant';
```

Блок объявления переменных начинается служебным словом **var**. Переменные перечисляются через запятую, в конце списка ставится двоеточие, после которого указывается их тип. Пример объявления переменных на языке Turbo Pascal:

```
var
  a, b : integer;
  x   : real;
  c   : char;
```

Основные стандартные типы данных языка Turbo Pascal:

boolean – булевская переменная, может принимать значение true (истина) или false (ложь);

integer – целочисленная переменная (значение от –32768 до 32767);

real – переменная с плавающей точкой (значение по модулю от 2.9×10^{-39} до 1.7×10^{38});

double – переменная с плавающей точкой (значение по модулю от 5.0×10^{-324} до 1.7×10^{308});

char – переменная одного символа;

string – строковая переменная.

Блок программы начинается ключевым словом **begin** и заканчивается ключевым словом **end**. Эти же ключевые слова используются для группировки команд. В программе последовательно перечисляются команды языка. Текст программы заканчивается символом «точка».

Основные операторы языка Turbo Pascal

Все команды языка Turbo Pascal разделяются символом «точка с запятой» («;»).

read, readln – ввод данных с клавиатуры (**readln** требует ввода данных через Enter, **read** – через пробел), в скобках перечисляются переменные, в которые будет осуществляться ввод данных. **readln(x, y)**; осуществляет ввод значения переменной x, затем y;

write, writeln – вывод текста и значений переменных на экран (**writeln**, в отличие от **write**, осуществляет переход на следующую строчку по завершении выполнения команды), **write(x, 'Абвгд', y, '_', 2)**; при значениях **x = 4, y = 7** выводит на экран «4Абвгд7_2»;

:= – оператор присваивания. **x := 5 + 7**; устанавливает значение переменной **x = 12**;

Комментарии в программе записываются внутри фигурных скобок { комментарий } или внутри символов (* комментарий *).

Математические операции языка Turbo Pascal

*	– умножение	(2*3 результат: 6);
/	– деление	(30/2 результат: 1.5E+01);
+	– сложение	(2+3 результат: 5);
–	– вычитание	(5–3 результат:2);
div	– целочисленное деление	(5 div 2 результат: 2);
mod	– остаток от деления	(5 mod 2 результат: 1).

Операции отношения

>	– больше;	<>	– не равно;
<	– меньше;	>=	– больше или равно;
=	– равно;	<=	– меньше или равно.

Основные математические функции

Abs(X) – возвращает абсолютное значение числа X;

Sin(X), Cos(X) – возвращает синус и косинус числа X, где X – угол в радианах;

Exp(X) – возвращает число, равное e^x ;

Ln(X) – возвращает число, равное натуральному логарифму от числа X;

Sqr(X) – возвращает число, равное квадрату числа X;

Sqrt(X) – возвращает число равное квадратному корню из числа X;

Frac(X) – возвращает число равное дробной части числа X;

Int(X) – возвращает число, равное целой части числа X;

Random(X) – возвращает случайное число в диапазон [0,X-1].

Содержание лабораторной работы

В лабораторной работе требуется разработать программу на языке программирования Turbo Pascal, которая вычисляет математическое выражение в соответствии с Вашим вариантом. В заданном выражении переменная x задается пользователем. Отчет о проделанной работе должен содержать: название и цель работы; номер варианта для выполнения задания и условие своего варианта; блок-схему решения задачи; тексты программ; полученные при расчетах численные результаты и выводы по проделанной лабораторной работе.

Варианты заданий

$$1. \quad y = \frac{x \cos^2 \frac{x}{2} + \sqrt[3]{|x|} \operatorname{tg}^3 x}{1 + e^{\frac{-x^2}{2}}}.$$

$$2. \quad y = \frac{18,3e^{\frac{-x^2}{2}} - 6,3e^{\frac{-x}{3}}}{7,4 \operatorname{tg} \frac{x}{2} - \sqrt[3]{2x}}, \text{ где } x \neq 0.$$

$$3. \quad y = e^{\frac{x^2}{2}} + \sqrt[3]{\frac{\operatorname{tg}^3 x^3}{1 + \sqrt{x}}}.$$

$$4. \quad y = \ln \frac{e^{-x^2} - \sqrt[4]{x} + \cos \frac{x}{2}}{\sqrt{|1 + \operatorname{tg}^2 x^2|}}.$$

$$5. \quad y = \frac{x^2 \sqrt{x} - x \cos^2 \frac{x}{2}}{\sqrt[4]{2\sqrt{1+e^{\ln x}}}}.$$

$$6. \quad y = \frac{x \ln^3 x^3 - \left| \sin^2 \frac{x}{2} - 1 \right|}{\sqrt{e^{\sin x - x^2} + 1}}, \text{ где } x > 0.$$

$$7. \quad y = \ln \left| \frac{e^{(-a^2+b)} + 2\sqrt{ab}}{2 \sin^2 \frac{a}{b} + b} \right|, \text{ где } a, b > 0.$$

$$8. \quad y = \left| \frac{\sqrt{1 - \ln|x^2 - 1|}}{tg^3 x^3 + 2 \ln|x+1|} \right|, \text{ где } x \neq \pm 1.$$

$$9. \quad y = \sqrt[4]{\frac{1 - tg^3 x^3}{1 + e^{\sin^2 x}}}.$$

$$10. \quad y = \ln \left| \frac{x^2}{2x} \right| + \sqrt[3]{\left| \frac{e^{(-1+x)}}{tg^2 x^2 + 1} \right|}, \text{ где } x \neq 0.$$

$$11. \quad y = \frac{5e^{\frac{1}{\cos^2 x + tg^2 \frac{x}{2}}}}{\sqrt[3]{tg^2 x^2 + 1}} + 1.$$

$$12. \quad y = \frac{3\sqrt[3]{6 - \sin \frac{x}{2}} + 3e^{-tg^2 x}}{2 \ln^2(x^2 + 1)}.$$

$$13. \quad y = \ln \sqrt[5]{\frac{tg^3 x^3 - 1}{2e^{-x^2} + 2 \cos \sqrt{x}}}.$$

$$14. \quad y = 2 \ln \left| \frac{4 - \sqrt[3]{\cos^3 \frac{x}{3} - 30}}{e^{\frac{1}{\lg x} + x^2}} \right|, \text{ где } x \neq k\pi, k \in \mathbb{Z}.$$

$$15. \quad y = \sqrt[3]{\cos \frac{\ln x^2 - 1}{1 + e^{x^2} - \operatorname{tg}(\sqrt[3]{x})}}, \text{ где } x \neq 0.$$

$$16. \quad y = 2e^{\frac{1}{\operatorname{tg} x} + x^2} + \ln \frac{1 - \ln^3 \frac{x}{2}}{\sqrt[4]{\operatorname{tg}^2 \frac{x}{2} + 1}}, \text{ где } x \neq k\pi, k \in \mathbb{Z}.$$

$$17. \quad y = \frac{3 \sqrt[4]{2 \sin \frac{x^2}{2} + \operatorname{tg}^2 \frac{x}{2}}}{2x + 1}, \text{ где } x \neq -\frac{1}{2}.$$

$$18. \quad y = \sqrt[5]{\left(\frac{\cos^3(x^3) - \sqrt[3]{\operatorname{tg}^2 x}}{e^{-x^2} + 2 \ln(x + 2)} \right)}, \text{ где } x > -2.$$

$$19. \quad y = \frac{3,5e^{\frac{x^2}{3}} - 8,3e^{\frac{x^2}{4}}}{3,3 \operatorname{tg} \frac{x}{6} - \sqrt[3]{8,8x^2}}, \text{ где } x \neq 0.$$

$$20. \quad y = \sqrt[3]{\operatorname{tg}^2 \frac{x}{2} + 3 \sqrt[5]{\frac{\ln^2 \frac{x}{2}}{2 \operatorname{tg}\left(\frac{x}{2}\right)}}}, \text{ где } x \neq 2k\pi, k \in \mathbb{Z}.$$

$$21. \quad y = \sqrt[5]{\ln \frac{x^2 + 5}{\sin^2 \frac{x}{2} + \cos^2 x}}.$$

$$22. \quad y = \sqrt[3]{\sqrt[5]{\left|1 - 3\sin^2 \frac{x}{2}\right|} + 2e^{-tg^3 x}}.$$

$$23. \quad y = \ln \frac{\left| \frac{x}{3} - 2\cos^3 \sqrt{\left|1 - \frac{x^2}{2}\right|} \right|}{\sqrt[4]{\left|5 - e^{-x^2 \cos x}\right|}}.$$

$$24. \quad y = \frac{e^{-tg \frac{x}{2}}}{\sin \frac{\left|1 - x^2\right|}{\sqrt{\left|\cos^3 x^3\right| + 2}}}, \text{ где } x \neq \pm 1, x \neq p + 2pk, k \in \mathbb{Z}.$$

$$25. \quad y = \frac{\ln|x| - e^{-x^2} + \sqrt[3]{|x|}}{tg^2 x^3 - x^2 + 3}, \text{ где } x \neq 0.$$

$$26. \quad y = \frac{2\left(\ln|x| - \sqrt[4]{|x+1|}\right)}{e^{\frac{-3}{\sqrt[3]{|ax+x^2|}}} + 2tgx^4}, \text{ где } x \neq 0.$$

$$27. \quad y = \sqrt[5]{1 + \sqrt[4]{\frac{tg|x^2 - 2|}{e^{\sqrt{|x|}} + \ln(x^2)}}}, \text{ где } x \neq 0.$$

$$28. \quad y = \frac{e^{-ax^2}}{\ln^3 \left| \frac{\sin^2 x^2 - \cos^2 x}{e^{-x^2}} \right|}.$$

$$29. \quad y = \ln^4 \left| \frac{x^2 - x^2}{e^{-x^2}} \right| + 5e^{-tg^3 x^3}.$$

$$30. \quad y = \frac{\sin(x^3 + 1)}{tg^4(\sqrt{|x-1|}) + e^{-x^2}}.$$

Пример программы на языке Turbo Pascal

Разработать алгоритм и составить программу вычисления значения выражения $y = \frac{e^{2x}}{\sin\left(\sqrt{|x^2 - 1|}\right)}$. Вывести исходные данные и результат

с пояснительным текстом.

Схема алгоритма работы программы представлена на рисунке 4.

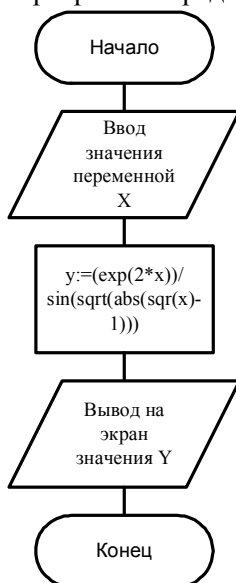


Рис. 4 – Схема алгоритма работы программы

```
Program Primer1;  
var  
  x, y : real;  
begin  
  write('Введите значение переменной X ');  
  readln(x);  
  y:=exp(2*x))/sin(sqrt(abs(sqr(x)-1)));  
  writeln('Значение выражения');  
  writeln('y=exp(2*x))/sin(sqrt(abs(sqr(x)-1)))');  
  writeln('равно ', y:8:4);  
end.
```

Контрольные вопросы

1. Как пользоваться меню среды Turbo Pascal?
2. Что отображается в окне редактора среды Turbo Pascal?
3. Какие основные функции меню среды Turbo Pascal?
4. Как сохранить программу на диске?
5. Что такое компиляция программы?
6. Как запустить программу на выполнение?
7. Что такое синтаксическая ошибка программы?
8. Как находить синтаксические ошибки программы?
9. Как объявляются переменные в языке Turbo Pascal?
10. Какие основные типы данных в языке Turbo Pascal?
11. Как группируются команды?
12. Как пользоваться комментариями?
13. Как ввести данные с клавиатуры?
14. Как вывести данные на экран?
15. Что такое присваивание?

Лабораторная работа № 2.

Программирование алгоритмов разветвленной структуры

Цель работы: изучение составного оператора, оператора условия, оператора выбора, оператора безусловного перехода и составление программ с разветвленной структурой на языке Turbo Pascal.

Теоретические сведения

Программы с разветвленной структурой используются, когда в алгоритме решения задачи предусмотрены альтернативные пути ее решения, т.е. из нескольких альтернатив выбирается одна, в зависимости от условия (условием является логическое выражение).

Составной оператор

Составной оператор – это последовательность произвольных операторов программы, заключенная в операторные скобки – зарезервированные слова `begin ... end`. При этом после `end` ставится точка с запятой (;). Язык Turbo Pascal не накладывает никаких ограничений на характер операторов входящих в составной оператор. Среди них могут быть и другие составные операторы.

`begin`

.....

`begin`

.....

`end;`

.....

`end;`

Pascal допускает произвольную глубину их вложения.

Условный оператор

Условный оператор позволяет проверить некоторое условие и в зависимости от результатов проверки выполнить то или иное действие. Таким образом, условный оператор – это средство ветвления вычислительного процесса. Структура условного оператора имеет следующий вид:

`if <условие> then <оператор_1> else <оператор_2>;`

где `if, then, else` – зарезервированные слова (если, то, иначе);

`<условие>` – произвольное выражение логического типа;

<оператор_1>, <оператор_2>– любые операторы языка Turbo Pascal.

Условный оператор работает по следующему алгоритму. Вначале вычисляется условное выражение <условие>. Если результат есть TRUE (истина), то выполняется <оператор_1>, а <оператор_2> пропускается; если результат есть FALSE (ложь), наоборот, <оператор_1> пропускается, а выполняется <оператор_2>. Часть else <оператор_2> условного оператора может быть опущена. Тогда при значении TRUE условного выражения выполняется <оператор_1>, в противном случае этот оператор пропускается.

Оператор выбора

Оператор выбора позволяет выбрать одно из нескольких возможных продолжений программы. Параметром, по которому осуществляется выбор, служит *ключ выбора* – выражение любого порядкового типа (кроме типов REAL и STRING). Структура оператора выбора имеет вид:

```
case <ключ_выбора> of <список_выбора> else <операторы>
end;
```

Здесь case, of, else, end – зарезервированные слова (случай, из, иначе, конец);

<ключ_выбора> – ключ выбора;

<список_выбора> – одна или более конструкций вида:

<константа_выбора> : <операторы>;

<константа_выбора> – константа того же типа,
что и выражение <ключ_выбора>;

<операторы> – произвольные операторы Pascal.

Оператор выбора работает следующим образом. Вначале вычисляется значение выражения <ключ_выбора>, а затем в последовательности операторов <список_выбора> отыскивается такой, которому соответствует константа, равная вычисленному значению. Если в списке выбора не будет найдена константа, соответствующая вычисленному значению ключа, управление передается операторам, стоящим за словом else. Часть else <операторы> можно опустить. Тогда при отсутствии в списке выбора нужной константы оператор выбора просто завершит свою работу.

Метка и оператор безусловного перехода

Метка в языке Pascal – это произвольный идентификатор, позволяющий именовать некоторый оператор программы и таким образом ссылаться на него. Метка располагается непосредственно перед помеченным оператором и отделяется от него двоеточием (:). Оператор можно помечать несколькими метками, которые в этом случае отделяются друг от друга двоеточием (:). Перед тем как появиться в программе, метка должна быть описана. Описание меток состоит из зарезервированного слова **label** (метка), за которой следует список меток:

```
var  
label lb1;  
begin  
  goto lb1;  
  .....  
  lb1 : .....  
  .....
```

Действие оператора **goto** состоит в передаче управления соответствующему помеченному оператору.

При программировании следует избегать злоупотреблением операторами безусловного перехода, так как это затрудняет понимание программы, делает ее запутанной и сложной в отладке.

Содержание лабораторной работы

В лабораторной работе требуется разработать программу на языке программирования Pascal 7.0 с разветвленной структурой в соответствии с Вашим вариантом. Отчет о проделанной работе должен содержать: название и цель работы; номер варианта для выполнения задания и условие своего варианта; блок–схему решения задачи; тексты программ; полученные при расчетах численные результаты и выводы по проделанной лабораторной работе.

Варианты заданий

1. С клавиатуры введены два числа. Если первое число больше второго на 2, то увеличить оба числа на 3, в противном случае добавить к введенным числам –1.
2. С клавиатуры введены два числа. Если первое число меньше второго на 5, то к меньшему числу прибавить 3, а от большего отнять 1, в противном случае оба числа возвести в квадрат.

3. С клавиатуры введены два числа. Если числа равны, то найти их сумму. В противном случае вычислить модуль разности.
4. Если во введенных с клавиатуры трёх числах есть отрицательное число, получить произведение этих чисел, в противном случае – их сумму.
5. Если второе из двух введенных чисел А и В равно 8, то вычислить $A+B$, а если нет, то получить $A-B$.
6. Если введенное с клавиатуры натуральное число кратно 3, вычислить его квадрат, а если нет, то – корень квадратный из числа.
7. Найти минимальное из трёх введенных с клавиатуры чисел А, В, С.
8. Если первое из введенных с клавиатуры чисел А и В равно 5, то вычислить $A*B$, а если нет, то получить A/B .
9. Если введенное с клавиатуры натуральное число n – чётное, то вычислить квадрат, куб и четвертую степень и отпечатать эти значения. В противном случае выдать информацию «вычислять не будем».
10. Если в последовательности из трёх чисел, введенных с клавиатуры, есть число 7, то вычислить произведение этих чисел, в противном случае получить их сумму.
11. Найти максимальное число в последовательности трёх чисел А, В, С.
12. Даны четыре числа А, В, С, D. Найти минимальное. Результат хранить в ячейке `minimum`.
13. В последовательности четырёх чисел, заданных с клавиатуры, найти количество положительных чисел.
14. Составить программу, упорядочивающую три числа А, В, С так, что в ячейке А находится минимальное число, в ячейке В – среднее, а в ячейке С – максимальное.
15. С клавиатуры введены три числа. Найти минимальное из них. Если оно больше нуля, вычислить произведение всех трёх чисел, в противном случае вычислить сумму всех трех чисел. При выводе организовать соответствующий комментарий.
16. Даны четыре числа А, В, С, D. Найти максимальное и результат записать в ячейку с именем `maximum`.
17. Составить программу, которая бы находила максимальное из трех чисел А, В, С и, если оно больше 5, увеличивала бы все числа в два раза.

18. С клавиатуры заданы два числа. Если их среднее арифметическое превышает их среднее геометрическое на 3, уменьшить каждое число на 1, в противном случае увеличить оба числа на 1.
19. Даны три числа. Найти произведение двух минимальных из них.
20. С клавиатуры заданы три числа. Найти среднее арифметическое положительных из них.
21. В последовательности четырёх целых чисел, заданных с клавиатуры, найти количество нулей.
22. Даны два прямоугольных треугольника, катеты которых равны a_1 , a_2 , b_1 , b_2 , соответственно. Определить номер треугольника с большей гипотенузой.
23. С клавиатуры введены три числа. Расположить их в порядке убывания в тех же ячейках.
24. С клавиатуры введены три числа. Если их среднее арифметическое в 3 раза больше их среднего геометрического, уменьшить значение каждого из них в 2 раза. В противном случае к каждому числу добавить 1.
25. С клавиатуры заданы два числа. Если первое из них больше или равно второму, удвоить их значения, в противном случае уменьшить каждое из них на единицу.
26. Найти минимальное число в последовательности их трёх чисел A, B, C.
27. Если сумма двух чисел, введенных с клавиатуры, превышает 30, увеличить первое число в 5 раз, второе – в три раза, в противном случае определить модуль их разности.
28. Если модуль разности двух введенных с клавиатуры чисел превышает 0,1, определить корень квадратный из модуля произведения этих чисел, в противном случае увеличить значение каждого числа в 2 раза.
29. Если три введенных с клавиатуры числа – отрицательные, определить их произведение, в противном случае определить их среднее арифметическое.
30. Если два первых среди введенных с клавиатуры трёх чисел положительны, увеличить все числа в 3 раза. В противном случае определить произведение всех чисел.

Пример программы на языке Pascal

Разработать алгоритм и составить программу, в которой пользователь вводит с клавиатуры два числа, после чего ему предоставляется

следующий выбор: если первое число меньше второго, то увеличить наименьшее из введенных чисел в 5 раз и вычислить их сумму, в противном случае, уменьшить наибольшее из них в 3 раза и вычислить их произведение.

Схема алгоритма работы программы представлена на рисунке 5.

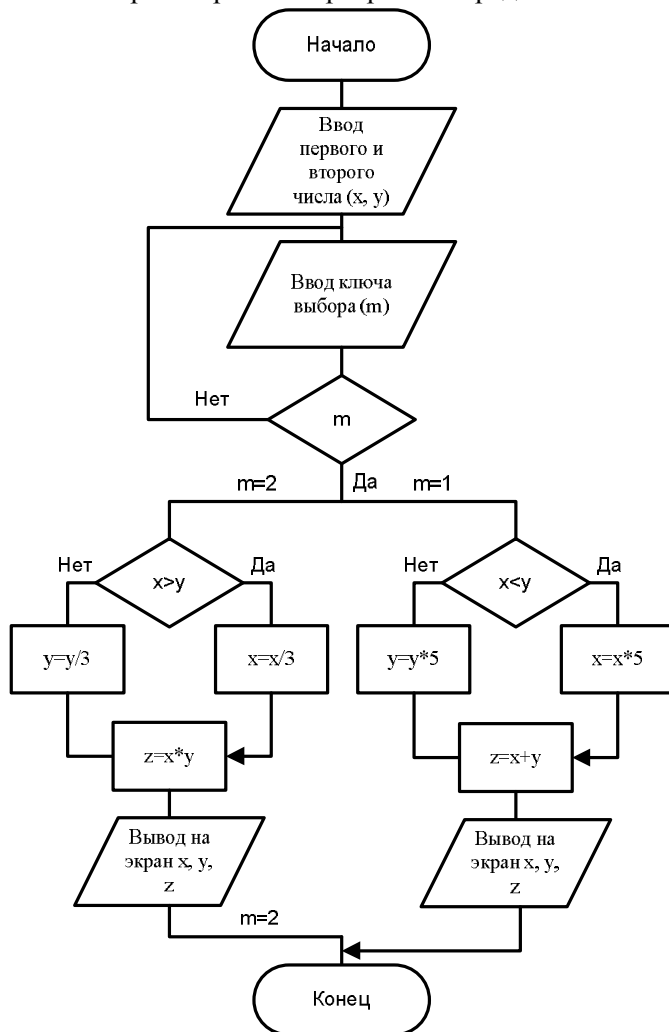


Рис. 5 – Схема алгоритма работы программы

```

Program Primer2;
label A;
var x, y, z : real;
    m      : integer;
begin
    write('Введите первое число '); readln(x);
    write('Введите второе число '); readln(y);
A : writeln('Нажмите 1 чтобы увеличить наименьшее из них
    в 5 раз и сложить два числа');
    writeln('Нажмите 2 чтобы уменьшить наибольшее из них
    в 3 раза и умножить два числа');
    readln(m);
    case m of
        1 : begin
            if x<y then x:=x*5 else y:=y*5;
            z:=x+y;
            writeln('Первое число равно ', x:8:4, ' Второе число
            равно ', y:8:4, ' Их сумма равна ', z:8:4);
            end;
        2 : begin
            if x>y then x:=x/3 else y:=y/3;
            z:=x*y;
            writeln('Первое число равно ', x:8:4, ' Второе число
            равно ', y:8:4, ' Их произведение равна ', z:8:4);
            end;
        else goto A;
    end;
end.

```

Контрольные вопросы

1. Что понимают под составным оператором?
2. Особенности организации составных операторов?
3. Что понимают под алгоритмом ветвления?
4. Привести примеры случаев ветвления.
5. Как обозначается ветвление в блок–схемах?
6. Какие операторы ветвления существуют в языке Turbo Pascal?
7. Структура условного оператора?
8. Какой формат имеет оператор IF?

9. Какие различия между полной и сокращенной формой условного оператора?
10. Структура оператора выбора?
11. Что такое ключ выбора, список выбора и константа выбора?
12. Какие различия между полной и сокращенной формой записи оператора выбора?
13. Что такое метка в языке Turbo Pascal?
14. Как описываются в программе метки?
15. Что такое оператор безусловного перехода?

Лабораторная работа № 3.

Организация циклов

Цель работы: изучение оператора «цикла с параметром», «цикла с предусловием», «цикла с постусловием» и составление программ с циклической структурой на языке Turbo Pascal.

Теоретические сведения

Циклы предназначены для многократного выполнения некоторой последовательности операторов, что значительно сокращает объем программы и повышает ее читабельность. В языке Turbo Pascal реализованы три оператора цикла. Формально эти операторы называют «цикл с параметром», «цикл с предусловием» и «цикл с постусловием». Для каждого из этих циклов существуют свои области применения. Эти области рассмотрим при подробном рассмотрении каждого из операторов цикла.

Цикл с параметром

Обобщенный синтаксис этого оператора следующий:

```
for <счетчик> := <нач_знач> to <кон_знач> do <тело_цикла>;
```

Здесь:

<счетчик> – параметр цикла – переменная типа Integer. Эта переменная при выполнении тела цикла при каждом выполнении последовательно принимает значения от **<нач_знач>** до **<кон_знач>** ;

<нач_знач>, **<кон_знач>** – начальное и конечное значение счетчика;

<тело_цикла> – последовательность операторов, которая выполняется в цикле. Тело цикла может содержать один оператор или несколько. В случае нескольких операторов их заключают в операторные скобки (составной оператор).

При выполнении программы процесс выполнения цикла выглядит следующим образом:

- первоначально (до цикла) значение переменной–счетчика в общем случае не определено;
- в момент первого выполнения оператора **for** счетчик принимает значение **<нач_знач>**;
- далее выполняются операторы **<тело_цикла>**. При выполнении этих операторов значение счетчика не изменяется;

- при достижении последнего оператора в теле цикла управление возвращается на оператор `for`, где значение счетчика увеличивается на 1 и сравнивается с `<кон_знач>`. Если счетчик стал больше, чем `<кон_знач>`, управление передается на следующий после цикла оператор.

Таким образом, можно сделать следующий вывод: цикл `for` выполняется столько раз, сколько различных значений есть между `<нач_знач>` и `<кон_знач>` включительно. Схема алгоритма цикла с параметром приведена на рисунке 6.

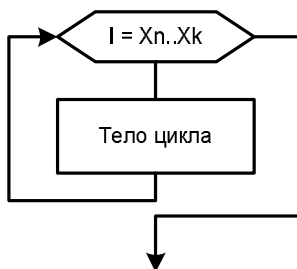


Рис. 6 – Схема алгоритма цикла с параметром

Если начальное значение равно конечному, цикл выполнится один раз. Если начальное значение больше конечного, цикл не выполнится ни разу.

В том случае, если требуется менять значения счетчика в сторону уменьшения – от большего к меньшему – то применяется несколько иная форма цикла:

`for <счетчик> := <нач_знач> downto <кон_знач> do <тело_цикла>;`

В этом случае при каждом выполнении оператора `for` значение счетчика уменьшается на 1.

Данный вид цикла применяется чаще всего для перебора элементов массива, и в тех случаях, когда количество проходов заранее известно.

Цикл с предусловием

Обобщенный синтаксис этого оператора следующий:

`while <условие> do <тело_цикла>;`

<Условие> является логическим выражением. Это означает, что результатом вычисления этого выражения является одно из значений TRUE или FALSE.

Тело цикла выполняется до тех пор, пока значением условия является TRUE. Обратите внимание, что возможна ситуация, когда условие всегда будет истинным. В этом случае цикл будет выполняться бесконечно. Если условие изначально ложно, тело цикла не выполнится ни разу, и управление сразу будет передано на оператор, следующий за телом цикла. Схема алгоритма цикла с предусловием приведена на рисунке 7.



Рис. 7 – Схема алгоритма цикла с предусловием

Этот цикл применяется в тех случаях, когда число итераций заранее неизвестно. Типичный пример – различные задачи поиска, или расчеты методом последовательных приближений.

Цикл с постусловием

Обобщенный синтаксис этого оператора следующий:

repeat <тело_цикла> **until** <условие>;

Как и в случае цикла с предусловием, условие является логическим выражением. В отличие от цикла **while**, этот цикл выполняется, пока условие ложно. Как только условие принимает истинное значение, выполнение цикла прекращается.

Схема алгоритма цикла с постусловием приведена на рисунке 8.



Рис. 8 – Схема алгоритма цикла с постусловием

Наиболее часто данный цикл применяется при вводе исходных данных и контроле корректности введенных данных.

Содержание лабораторной работы

В лабораторной работе требуется разработать программу на языке программирования Turbo Pascal с циклической структурой в соответствии с Вашим вариантом. Отчет о проделанной работе должен содержать: название и цель работы; номер варианта для выполнения задания и условие своего варианта; блок-схему решения задачи; тексты программ; полученные при расчетах численные результаты и выводы по проделанной лабораторной работе.

Варианты заданий

1. Разработать программу на языке программирования Turbo Pascal:
 - а) Вычислить функцию $y = 0,5x + \sin^2 x$ на отрезке $[0, 3p]$ с шагом $h = p/2$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.
 - б) Определить знак произведения функции $y = (3 - \sin 2x) \cos \frac{x}{3}$ на интервале изменения аргумента от x_n до x_k с шагом h .
2. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = |x^2 + 5x - 6|$ на отрезке $[-7, 7]$ с шагом $h = 0,5$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Найти, во сколько раз среднее арифметическое положительных значений отличается от среднего геометрического этих же значений функций $y = (3x - \cos x) \sin 2x$ на интервале изменения аргумента от x_n до x_k с шагом h .

3. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \sqrt{x(x-3)^4}$ на отрезке $[1, 6]$ с шагом $h = 0,25$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Найти среднее арифметическое значений функции $y = (1 - e^{-x^2}) \sin 2x$ на интервале изменения аргумента от x_n до x_k с шагом h .

4. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \sqrt{6x^2 - x}$ на отрезке $[-6, 6]$ с шагом $h = 0,5$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Найти среднее арифметическое положительных значений функции $y = (1 - e^{-x^2}) \cos 2x$ на интервале изменения аргумента от x_n до x_k с шагом h .

5. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = x + 1 + \sin(x - 1)$ на отрезке $[0, 4p]$ с шагом $h = p/6$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Определить знак максимального значения функции $y = (1 - \operatorname{ctg} 3x)e^{-x^2}$ на интервале изменения аргумента от x_n до x_k с шагом h .

6. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = x \sin x + \cos^2 x$ на отрезке $[-4p, 4p]$ с шагом $h = p/3$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Найти максимальное значение функции $y = (1 - e^{-x^2}) \sin 3x$ на интервале изменения аргумента от x_n до x_k с шагом h .

7. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \sqrt{1 + x^3}$ на отрезке $[0, 5]$ с шагом $h = 0,2$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Определить знак минимального значения функции $y = (1 - \operatorname{tg} 2x)e^{-x}$ на интервале изменения аргумента от x_n до x_k с шагом h .

8. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \ln x + \frac{1}{x}$ на отрезке $[0,2; 2]$ с шагом $h = 0,1$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Найти минимальное значение функции $y = (1 - e^{-x}) \cos 2x$ на интервале изменения аргумента от x_n до x_k с шагом h .

9. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = |\cos x|$ на отрезке $[-3p, 3p]$ с шагом $h = p/4$. На каждом шаге выводить на печать значения переменных

x и y . Для данной задачи написать программы, используя операторы цикла *for, while, repeat*.

б) Найти среднее геометрическое положительных значений функции $y = (e^{-x} - 2x) \cos x$ на интервале изменения аргумента от x_n до x_k с шагом h .

10. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \sqrt{\cos^2 x + 1}$ на отрезке $[0, 4p]$ с шагом $h = p/4$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for, while, repeat*.

б) Найти количество положительных и количество отрицательных значений функции $y = (1 - \operatorname{tg}^2 x) \sin x$ на интервале изменения аргумента от x_n до x_k с шагом h .

11. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \sqrt{1 + x^2}$ на отрезке $[-1, 1]$ с шагом $h = 0,1$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for, while, repeat*.

б) Определить сумму положительных значений функции $y = \left(\frac{1}{2} - e^{-x^2} \right) \sin 3x$ на интервале изменения аргумента от x_n до x_k с шагом h .

12. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = x^3 + \sin x$ на отрезке $[-2p, 2p]$ с шагом $h = p/6$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for, while, repeat*.

б) Определить произведение положительных значений функции $y = (1 - e^{-x^2}) \sin \frac{x}{2}$ на интервале изменения аргумента от x_n до x_k с шагом h .

13. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = x - \operatorname{tg} x$ на отрезке $[0, p/2]$ с шагом $h = p/10$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Найти среднее арифметическое положительных значений функции $y = \left(\cos \frac{x}{3} - \operatorname{tg} x \right) e^{-x}$ на интервале изменения аргумента от x_n до x_k с шагом h .

14. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \ln x^2 + 3x$ на отрезке $[1, 5]$ с шагом $h = 0,4$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Определить сумму отрицательных значений функции $y = (0,2 - e^{-x}) \cos 3x$ на интервале изменения аргумента от x_n до x_k с шагом h .

15. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \sqrt{\operatorname{ctg} x + 0,5}$ на отрезке $[-p/2, p/2]$ с шагом $h = p/10$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Определить, во сколько раз модуль суммы отрицательных значений функции отличается от суммы положительных значений функции $y = (x^2 + 1) \cos 3x$ на интервале изменения аргумента от x_n до x_k с шагом h .

16. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = x \sin x + e^x$ на отрезке $[-4p, 4p]$ с шагом $h = p/3$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Определить, на сколько модуль суммы отрицательных значений отличается от суммы положительных значений функции

$y = (\sin 3x + 2) \sin \frac{x}{2}$ на интервале изменения аргумента от x_n до x_k

с шагом h .

17. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \sqrt[3]{1-x^3}$ на отрезке $[-5,5]$ с шагом $h = 0,2$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Определить произведение отрицательных значений функции $y = (\sin 2x + 2x) \cos x$ на интервале изменения аргумента от x_n до x_k с шагом h .

18. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \sqrt{|\cos x|}$ на отрезке $[0,4p]$ с шагом $h = p/4$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Определить, во сколько раз отличается количество положительных от количества отрицательных значений функции $y = (\cos 3x - 1)e^{-x}$ на интервале изменения аргумента от x_n до x_k с шагом h .

19. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = x\sqrt{1-x^2}$ на отрезке $[-1,1]$ с шагом $h = 0,1$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Найти количество отрицательных значений функции $y = (2xe^{-x} - x) \sin x$ на интервале изменения аргумента от x_n до x_k с шагом h .

20. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \ln(x + \sqrt{x})$ на отрезке $[0,8; 2p]$ с шагом $h = 0,1$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Найти среднее арифметическое максимального и минимального значений функции $y = (2x - e^{-x^2}) \cos \frac{x}{2}$ на интервале изменения аргумента от x_n до x_k с шагом h .

21. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \frac{a}{x} + \sqrt{x^2 + 1}$ на отрезке $[1,2]$ с шагом $h = 0,5$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Определить, на сколько отличается количество положительных от количества отрицательных значений функции $y = (e^{-3x} - x) \cos \frac{x}{3}$ на интервале изменения аргумента от x_n до x_k с шагом h .

22. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \ln(x + 7\sqrt{x + 1,65})$ на отрезке $[0,7]$ с шагом $h = 0,1$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Определить, во сколько раз произведение положительных значений функции отличается от модуля произведения отрицательных значений функции $y = (x^3 - 1) \sin 2x$ на интервале изменения аргумента от x_n до x_k с шагом h .

23. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \sin^2 \sqrt{ax}$ при $a = 20,3$ на отрезке $[0,5; 2]$ с шагом $h = 0,1$. На каждом шаге выводить на печать значе-

ния переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Определить знак суммы значений функции $y = (x^2 + 1)\cos 2x$ на интервале изменения аргумента от x_n до x_k с шагом h .

24. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \sqrt{2,5x^2 + 0,4\sin x + 1}$ на отрезке $[-1,1]$ с шагом $h = 0,2$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Найти разность максимального и минимального значений функции $y = (1 - e^{-x})\sin \frac{x}{2}$ на интервале изменения аргумента от x_n до x_k с шагом h .

25. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \frac{\ln^3 x + x}{\sqrt{x + 2,2}}$ на отрезке $[0,2;2]$ с шагом $h = 0,1$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Найти произведение максимального и минимального значений функции $y = \left(\frac{1}{2} - e^{-x^2}\right)\cos 3x$ на интервале изменения аргумента от x_n до x_k с шагом h .

26. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \sqrt{|x^3 - 0,3x^2 + 0,7x|}$ на отрезке $[-3,2]$ с шагом $h = 0,1$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Найти корень квадратный из модуля произведения максимального и минимального значений функции $y = (0,3e^{-x} - 2)\sin x$ на интервале изменения аргумента от x_n до x_k с шагом h .

27. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \sqrt{|\sin x^3 - \cos x^2 + x|}$ на отрезке $[-2p, 2p]$ с шагом $h = p/2$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Определить знак суммы максимального и минимального значений функции $y = (3 - \cos 2x)\sin \frac{x}{3}$ на интервале изменения аргумента от x_n до x_k с шагом h .

28. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \frac{\sqrt{|x^2 - 1|}}{2x}$ на отрезке $[-10, 10]$ с шагом $h = 1,5$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Определить, является ли функции положительным числом. Функция имеет вид $y = (1 - e^{-x^3})\sin x$ на интервале изменения аргумента от x_n до x_k с шагом h .

29. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \sqrt{\frac{2\cos^2 x}{\sin x + \cos x}}$ на отрезке $[0, 3p]$ с шагом $h = p/2$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Определить знак максимального значения функции $y = (3 - \sin x e^{-x}) \cos x$ на интервале изменения аргумента от x_n до x_k с шагом h .

30. Разработать программу на языке программирования Turbo Pascal:

а) Вычислить функцию $y = \sqrt[3]{x(3x^3 + 7)}$ на отрезке $[-2, 0]$ с шагом $h = 0,2$. На каждом шаге выводить на печать значения переменных x и y . Для данной задачи написать программы, используя операторы цикла *for*, *while*, *repeat*.

б) Определить, превышает ли количество положительных значений функции $y = (1 - e^{-2x}) \cos 2x$ на интервале изменения аргумента от x_n до x_k с шагом h .

Пример программы на языке Turbo Pascal

Определить превышает ли количество положительных значений функции $y = \sin(3x) + \cos(x^2)$ количество ее отрицательных значений на интервале изменения аргумента от x_n до x_k с шагом h .

Схема алгоритма работы программы представлена на рисунке 9.

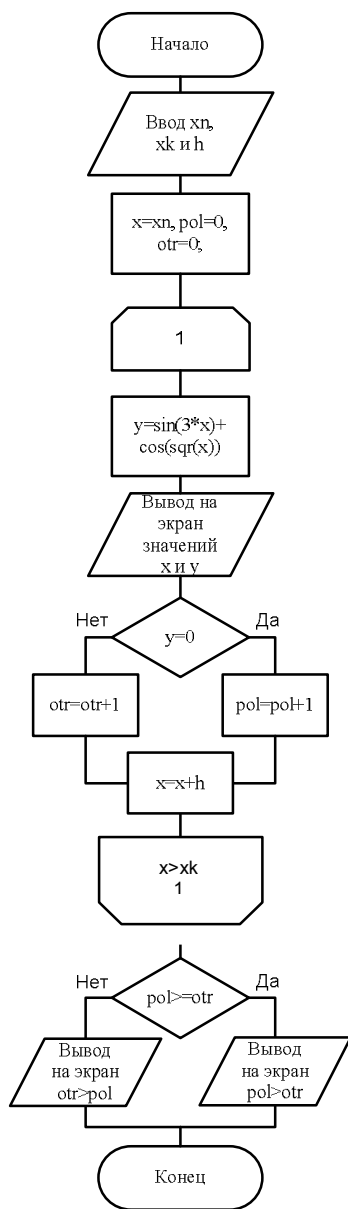


Рис. 9 – Схема алгоритма работы программы

Program Primer3;

var

y, x, xn, xk, h : real;

i, pol, otr : integer;

begin

write('Начальное значение переменной x='); readln(xn);

write('Конечное значение переменной x='); readln(xk);

write('Расчетный шаг h='); readln(h);

x:=xn; pol:=0; otr:=0;

repeat

y:=sin(3*x)+cos(sqr(x));

writeln('Текущие значения: x=', x:8:4, ' y=', y:8:4);

if y>=0 then pol:=pol+1 else otr:=otr+1;

x:=x+h;

until x >= xk;

if pol>otr then writeln('Количество Y_пол=', pol, ' Y_отт=',
otr, ' Y_пол>Y_отт') else writeln('Количество Y_пол=', pol, '
Y_отт=', otr, ' Y_пол<Y_отт');

end.

Контрольные вопросы

1. Что такое «цикл»?
2. Какие виды циклов есть в Паскале?
3. Обобщенный синтаксис оператора цикла с параметром?
4. Обобщенный синтаксис оператора цикла с предусловием?
5. Обобщенный синтаксис оператора цикла с постусловием?
6. Чем отличается цикл с предусловием от цикла с постусловием?
7. Сколько раз минимум может выполняться цикл с предусловием?
8. Сколько раз минимум может выполняться цикл с постусловием?
9. Какой тип переменной используется для цикла с параметром?
10. Какая форма цикла с параметром применяется для цикла «от большего к меньшему значению»?
11. Какой тип должно иметь условие для цикла с постусловием?
12. Каков шаг изменения переменной–счетчика в цикле с параметром?
13. Для решения каких задач применяется цикл с параметром?
14. Для решения каких задач применяется цикл с предусловием?
15. Для решения каких задач применяется цикл с постусловием?

Лабораторная работа № 4.

Работа с одномерными массивами

Цель работы: изучение принципов работы с одномерными массивами на языке программирования Pascal. Получение навыков применения основных алгоритмов для решения задач с использованием одномерных массивов.

Теоретические сведения

Массив – это конечная поименованная совокупность элементов одинакового типа. Число элементов в массиве называется размерностью массива. Каждый элемент массива задается своим порядковым номером в массиве – так называемым *индексом*. Примером может служить список фамилий студентов одной группы, где каждый студент однозначно определяется своим порядковым номером в списке (индексом в массиве).

Перед использованием массив, как и любая переменная в Pascal-программах, должен быть объявлен в разделе объявления переменных. В общем виде объявление массива выглядит так:

```
<имя> : array [<нижний_индекс> .. <верхний_индекс>] of  
<тип>;
```

Здесь:

<имя> – имя переменной массива;

array – ключевое слово, обозначающее, что переменная является массивом;

<нижний_индекс> и **<верхний_индекс>** – целые числа, определяющие диапазон изменения индексов (номеров) элементов массива и, неявно, количество элементов (размер) массива;

<тип> – тип элементов массива.

Примеры объявления массивов:

```
temper: array[1..31] of real;
```

```
koef: array[0..2] of integer;
```

```
name: array[1..30] of string[25];
```

Размещение массива в памяти происходит до выполнения программы, поэтому при описании индекса можно применять только константы или константные выражения. Использовать для этого переменные нельзя!

Обычно при описании массива верхняя граница его индекса задается в виде именованной константы. Именованная константа объявля-

ется в разделе описания констант, который располагается перед разделом объявления переменных. Например, массив названий команд – участниц чемпионата по футболу может быть объявлен следующим образом:

```
const
NT=18; {число команд}
SN=25; {предельная длина названия команды}
var
team: array[1..NT] of string[SN]
```

Массив может быть также описан следующим образом:

```
type <имя_типа> = array[<нижний_индекс> .. <верх-
ний_индекс>] of <тип>;
```

где

type – зарезервированное слово, используемое для создания пользовательского типа данных.

Примеры описания типа массивов:

Type

M=array[1..10] of integer;

AR=array[0..15] of real;

VT=array[-5..5] of string[10]

В первом операторе описан тип массива из целых чисел, которые нумеруются от 1 до 10. Во втором операторе элементами массива являются вещественные числа пронумерованные от 0 до 15. Соответственно в третьем примере описан тип массива строковых переменных с изменением индексов от –5 до 5.

После задания типа массива переменные этого типа описываются обычным образом:

```
var
a:m;
b:ar;
c:vt;
```

Тип элементов массива может быть любым, кроме файлового, тип индексов – интервальным, перечисляемым или byte.

С массивами в целом можно выполнять только одну операцию: присваивание. При этом массивы должны быть одного типа, например:

```
d:=a;
```


С отдельными элементами массива выполняются все остальные действия. Для обращения к элементу массива после имени массива указывается номер элемента массива в квадратных скобках:

a[3]

С элементом массива можно делать все, что допустимо для переменных этого же типа. Присвоение значения элементам массива можно делать до начала выполнения программы присвоением, можно вводить значения как исходные данные, а можно получать в результате расчетов.

К типичным операциям с массивами можно отнести: вывод массива; ввод массива; сортировка массива; поиск в массиве заданного элемента; поиск в массиве максимального или минимального элемента.

Содержание лабораторной работы

В лабораторной работе требуется разработать программу на языке программирования Turbo Pascal 7.0 реализующую алгоритмы обработки одномерных массивов в соответствии с Вашим вариантом. Отчет о проделанной работе должен содержать: название и цель работы; номер варианта для выполнения задания и условие своего варианта; блок-схему решения задачи; тексты программ; полученные при расчетах численные результаты и выводы по проделанной лабораторной работе.

Варианты заданий

1. Разработать программу на языке программирования Turbo Pascal:
 - а) В массиве $t(n)$, где n – заданное число, сформированном случайным образом, найти количество элементов меньших 0,3.
 - б) В массив произвольного размера внесены суммы месячного заработка в порядке возрастания табельного номера. Найти количество работающих, чья зарплата ниже средней и количество работающих, чья зарплата выше средней.
2. Разработать программу на языке программирования Turbo Pascal:
 - а) Массив $a(30)$ сформирован случайным образом. Найти в нём количество элементов меньших среднего арифметического положительных элементов этого массива.
 - б) В массив внесена стоимость книг в порядке возрастания их номеров в каталоге. Найти самую дорогую и самую дешевую книги (их номера), если всего книг 30.
3. Разработать программу на языке программирования Turbo Pascal:

- а) В массиве $s(20)$ найти нулевые элементы и заменить их средним арифметическим положительных элементов массива $t(15)$.
- б) В массив занесено население 12-ти городов в соответствии с их кодировочными номерами от 1 до 12. Найти номер города, население которого минимально. Вывести числовое значение населения этого города.
4. Разработать программу на языке программирования Turbo Pascal:
- а) Массив $b(25)$ сформирован случайным образом. Поменять местами в этом массиве максимальный и минимальный элементы.
- б) В массив внесены суммы вкладов вкладчиков, которые зашифрованы номерами. Найти среднюю сумму вкладов. Определить сколько вкладчиков имеют вклады выше средней суммы.
5. Разработать программу на языке программирования Turbo Pascal:
- а) В массиве $y(60)$, сформированном случайным образом, произвести сортировку, расположив элементы в порядке возрастания.
- б) В массив внесен месячный баланс 12 подразделений предприятия. Найти количество подразделений, имеющих отрицательный баланс. Найти номер подразделения с наибольшей суммой баланса.
6. Разработать программу на языке программирования Turbo Pascal:
- а) В массиве $y(25)$, сформированном случайным образом, найти среднее геометрическое модулей всех ненулевых элементов.
- б) В массив внесен годовой баланс предприятия. Найти среднее значение суммы баланса за год. Определить, в какие месяцы (по номерам) баланс был меньше средней величины.
7. Разработать программу на языке программирования Turbo Pascal:
- а) В массиве $b(20)$, сформированном случайным образом, произвести сортировку, расположив элементы в порядке убывания.
- б) В массив внесены оценки ученика по определенному предмету (их 10). Найти среднюю оценку ученика по этому предмету. Определить сколько пятерок получил ученик в четверти.
8. Разработать программу на языке программирования Turbo Pascal:
- а) В массиве $d(12)$ найти среднее геометрическое положительных элементов массива.
- б) В массив внесены результаты контрольной работы в классе по математике. Отсутствующие ученики кодируются оценкой 0. Определить количество отсутствующих и получивших 4 и 5 (вместе).
9. Разработать программу на языке программирования Turbo Pascal:

а) Элементы массива y сформированы по следующему закону:

$$y_i = \begin{cases} \cos^2 i^2, & \text{если } i - \text{четное}, \\ \sin i/2, & \text{если } i - \text{нечетное}, \end{cases} \quad \text{найти среднее арифметическое положи-}$$

тельных элементов массива, где $i \in [1..n]$. Определить в полученном массиве y количество положительных и отрицательных элементов.

б) В массив внесены результаты контрольной работы в классе по физике. Найти средний балл учащихся, полученный по этой контрольной. Подсчитать количество пятерок, четверок, троек и двоек (в классе 25 учащихся).

10. Разработать программу на языке программирования Turbo Pascal:

а) В массиве $t(30)$, сформированном случайным образом, найти количество элементов, стоящих на четных местах, удовлетворяющих условию $a < t_i < b$, где a и b – заданные числа.

б) В массиве содержатся результаты соревнований по плаванию, введенные по возрастанию номеров участников. Определить номер участника, показавшего лучший результат, если их было 25.

11. Разработать программу на языке программирования Turbo Pascal:

а) В массиве $y(30)$, сформированном случайным образом, найти максимальный элемент и его номер.

б) В массиве хранится 10 случайных чисел в интервале $[-1,1]$. Найти количество положительных и отрицательных чисел. Определить, во сколько суммарное количество положительных чисел превышает отрицательных.

12. Разработать программу на языке программирования Turbo Pascal:

а) В массиве $b(40)$, сформированном случайным образом, найти количество элементов, стоящих на нечетных местах, удовлетворяющих условию $d < b_i < t$, где d и t – заданные числа.

б) В массиве содержится сумма заработной платы работающих в цехе по порядку их номеров в ведомости (их всего 20). Найти среднюю заработную плату. Определить сколько работающих получают больше средней заработной платы, а сколько меньше.

13. Разработать программу на языке программирования Turbo Pascal:

а) Элементы массива a сформированы следующим образом:

$$a_i = \begin{cases} \lg^2 i, & \text{если } i - \text{нечетное}, \\ \sin i/2, & \text{если } i - \text{четное}. \end{cases} \quad \text{Найти среднее арифметическое положи-}$$

тельных элементов этого массива, где $i \in [1..n]$.

б) В массив внесен рост учеников класса из 20 человек. Найти самый максимальный и самый минимальный рост.

14. Разработать программу на языке программирования Turbo Pascal:

а) Сформировать массив с по следующему принципу:

$$c_i = \begin{cases} \operatorname{tg} i^2, & \text{если } i - \text{нечетное}, \\ i^2, & \text{если } i - \text{четное}. \end{cases} \quad i \in [1..n].$$

Перенести положительные элементы массива в массив у (поряд) а отрицательные элементы в массив х (поряд).

б) В массиве содержится сумма заработной платы работающих в отделе по порядку их номеров в ведомости (всего работающих 15). Найти номера в ведомости с максимальной и минимальной заработной платой.

15. Разработать программу на языке программирования Turbo Pascal:

а) В массиве b(25), сформированном случайным образом, найти количество элементов, удовлетворяющих условию $l < b_i < m$, где l, m – заданные числа.

б) В массив внесены результаты соревнований по бегу в порядке возрастания номеров участников (их всего 20). Найти значение самого лучшего результата и самого худшего.

16. Разработать программу на языке программирования Turbo Pascal:

а) Даны два массива. Массив t(20) задан произвольно. Элементы массива d(20) сформированы следующим образом:

$$d_i = \begin{cases} \frac{i+1}{2} \cos t_i, & \text{если } i - \text{четное}, \\ 1 - 2 \sin \frac{t_i}{2}, & \text{если } i - \text{нечетное}. \end{cases}$$

Заменить все отрицательные элементы массива d(20) средним арифметическим положительных элементов массива t. Вывести новый массив d.

б) В массиве хранятся оценки студента в сессию (результаты пяти экзаменов). Определить к какой категории (отличник, занимающийся на 4 и 5, троечник) относится данный студент.

17. Разработать программу на языке программирования Turbo Pascal:

а) В массиве t(10) найти среднее геометрическое положительных элементов и среднее арифметическое отрицательных элементов.

б) Экзаменационная ведомость может содержать n фамилий с оценками по физике. Определить количество абитуриентов, получивших двойки и пятерки.

18. Разработать программу на языке программирования Turbo Pascal:

а) Даны два массива. Массив $d(16)$ задается произвольно. Элементы массива $c(16)$ формируются следующим образом:

$$c_i = \begin{cases} \frac{d_i}{2} \cos d_i, & \text{если } i - \text{кратно } 3 \\ \operatorname{tg} d^2, & \text{во всех других случаях.} \end{cases} \quad \text{Расположить элементы массива } c \text{ в}$$

порядке убывания.

б) Товарная ведомость содержит n позиций со стоимостью товаров. Найти количество товаров, цена которых выше некоторой величины.

19. Разработать программу на языке программирования Turbo Pascal:

а) Даны два массива $t(20)$ и $s(20)$. Найти среднее геометрическое положительных элементов массива t и заменить им все отрицательные элементы массива s .

б) Товарная ведомость содержит n позиций со стоимостью товаров. Расположить суммы стоимости в порядке убывания.

20. Разработать программу на языке программирования Turbo Pascal:

а) Даны два массива. Элементы массива $d(15)$ заданы произвольно, а элементы массива $a(15)$ сформированы случайным образом. Сложить массивы d и a . Найти в новом массиве второй отрицательный элемент и вывести его на печать. Если количество отрицательных элементов меньше двух, то дать об этом сообщение.

б) Товарная ведомость содержит n позиций со стоимостью товаров. Расположить суммы стоимости в порядке их возрастания.

21. Разработать программу на языке программирования Turbo Pascal:

а) Даны два массива $x(12)$ и $y(10)$. В массиве y все нулевые элементы заменить средним арифметическим элементов массива x .

б) Балансовый отчет содержит n позиций с указанием величины и знака баланса. Распечатать вначале значения всех положительных балансов, а затем значения всех отрицательных балансов.

22. Разработать программу на языке программирования Turbo Pascal:

а) Произвести сортировку массива $t(30)$, записав положительные элементы в массив y подряд, а отрицательные – в массив x подряд.

б) Ведомость заработной платы содержит n позиций. Определить сколько человек получают зарплату на 60 процентов больше средней, сколько человек получают зарплату на 50 процентов ниже средней.

23. Разработать программу на языке программирования Turbo Pascal:

а) Элементы массива $y(20)$ сформированы следующим образом:

$$y_i = \begin{cases} 2 \operatorname{tg} i^2, & \text{если } i - \text{четное} \\ \cos^2 \frac{i}{2} - 1, & \text{если } i - \text{кратно } 5 \\ i^2 - 10, & \text{во всех других случаях.} \end{cases} \quad \text{Найти максимальный и минимальный}$$

элементы массива y и поменять их местами.

б) В списке указаны суммы вкладов n вкладчиков. Расположить эти суммы в порядке убывания сумм вкладов.

24. Разработать программу на языке программирования Turbo Pascal:

а) Найти разность элементов двух массивов $t(12)$ и $n(12)$ и определить номер строки, для которой эта разность максимальна.

б) Ведомость заработной платы сотрудников учреждения содержит n позиций. Найти минимальную заработную плату и определить насколько она ниже средней.

25. Разработать программу на языке программирования Turbo Pascal:

а) Даны два массива $k(15)$ и $d(5)$. Из отрицательных элементов массивов k и d сформировать массив $z(20)$, содержащий только отрицательные элементы. Если отрицательных элементов в массивах k и d окажется меньше 20, дополнить оставшиеся места -1 .

б) Результаты соревнований по плаванию оформлены в виде протокола, содержащего информацию о времени прохождения дистанции. Определить сколько из n участников заплыва заслуживают присвоения I разряда (результат $\leq t$) и сколько – II разряда ($t < \text{результат} \leq l$).

26. Разработать программу на языке программирования Turbo Pascal:

а) Массив $b(28)$ сформирован случайным образом. Найти максимальный элемент в нем. Рассортировать элементы после этого элемента в порядке убывания.

б) Балансовый отчет содержит n позиций с указанием величины и знака баланса. Распечатать вначале значения всех положительных балансов, затем значения всех отрицательных балансов.

27. Разработать программу на языке программирования Turbo Pascal:

а) Найти сумму элементов двух массивов $a(15)$ и $b(15)$ и определить номер строки, для которой эта сумма максимальна.

б) Результаты месячной работы n участников цеха сведены в таблицу, где указан баланс доходов–расходов. Определить количество участников, которые имели отрицательный баланс.

28. Разработать программу на языке программирования Turbo Pascal:

а) Известны данные по продаже компьютеров в течении недели. Найти общее количество проданных компьютеров.

б) Подсчитать среднемесячную зарплату сотрудника предприятия и найти зарплату, которая наиболее близка к средней. В качестве результата вывести среднюю зарплату, наиболее близкую и ее номер в массиве.

29. Разработать программу на языке программирования Turbo Pascal:

а) Курс доллара в течение года менялся в диапазоне от 28 руб. до 30 руб. Найти наибольшее значение курса доллара. В качестве результата вывести номер месяца и значение курса доллара.

б) Известен месячный план выпуска некоторой продукции и объемы выпущенной продукции заводом за год (помесячно). Определить месяц, в котором было максимальное отклонение от плана. В качестве результата вывести номер месяца и отклонение.

30. Разработать программу на языке программирования Turbo Pascal:

а) Даны результаты сдачи экзамена по информатике группы из 15 студентов. Подсчитать количество студентов, не сдавших экзамен, в численном и в процентном соотношении.

б) Известны данные по продаже компьютеров в течение недели. Расположить эти данные в порядке возрастания.

Пример программы на языке Turbo Pascal

Предположим, известны результаты соревнований по стрельбе, в которых принимали участие 9 человек. Расположить данные результаты в порядке возрастания набранных при стрельбе очков.

Схема алгоритма работы программы представлена на рисунке 10.

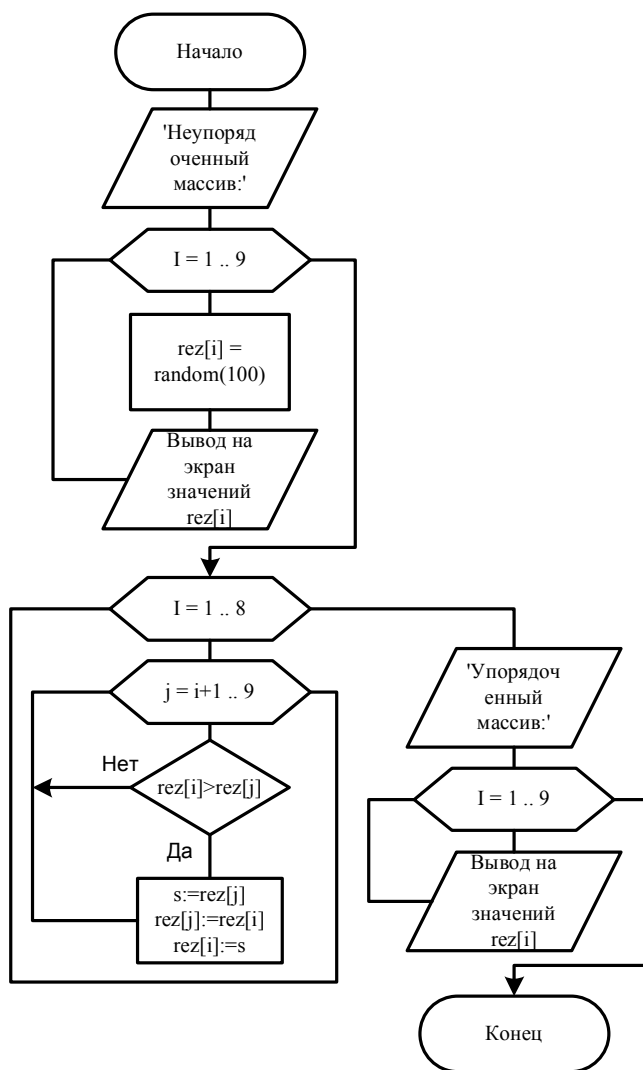


Рис. 10 – Схема алгоритма работы программы

```

Program Primer4;
var
  rez : array[1..9] of integer;
  i, j, s : integer;
begin

```



```

randomize;
writeln;
writeln('Неупорядоченный массив:');
for i:=1 to 9 do begin rez[i]:=random(101); write(rez[i]:5); end;
writeln;
for i:=1 to 8 do
  for j:=i+1 to 9 do
    if rez[i]>rez[j] then
      begin
        s:=rez[j]; rez[j]:=rez[i]; rez[i]:=s;
      end;
writeln('Упорядоченный массив:');
for i:=1 to 9 do write(rez[i]:5);
end.

```

Контрольные вопросы

1. Что понимают под массивом данных?
2. Что называют размерностью массива?
3. Что понимают под индексом элемента массива?
4. Какой массив называется одномерным?
5. Приведите примеры одномерных массивов.
6. Как описываются одномерные массивы на языке Turbo Pascal?
7. Как задается диапазон изменения индексов массива?
8. Как обозначаются индексы массивов на языке Turbo Pascal?
9. Какого типа могут быть элементы массива?
10. Какого типа могут быть индексы элементов массива?
11. Какие стандартные алгоритмы по работе с одномерными массивами Вы знаете?
12. Какими способами может быть заполнен массив? Приведите примеры.
13. Как будет выглядеть блок схема для решения задачи сортировки одномерного массива?
14. Как будет выглядеть блок схема для решения задачи поиска в одномерном массиве заданного элемента?
15. Как будет выглядеть блок схема для решения задачи поиска в одномерном массиве максимального или минимального элемента?

Лабораторная работа № 5.

Работа с двумерными массивами

Цель работы: изучение принципов работы с двумерными массивами на языке программирования Turbo Pascal. Получение навыков применения основных алгоритмов для решения задач с использованием двумерных массивов.

Теоретические сведения

Исходные данные для решения многих задач удобно представить в виде таблицы. Колонки и строки таблицы, как правило, содержат однородную информацию, если использовать терминологию Turbo Pascal – данные одинакового типа. Поэтому в программе для хранения и обработки табличных данных можно использовать совокупность одномерных массивов или двумерный массив.

В общем виде описание двумерного массива выглядит следующим образом:

`<Имя> : array [<нижняя_граница_индекса1> .. <верхняя_граница_индекса1>, <нижняя_граница_индекса2> .. <верхняя_граница_индекса2>] of <тип>;`

Здесь:

`<Имя>` – имя массива;

`array` – ключевое слово, показывающее, что объявляемый элемент данных является массивом;

`<нижняя_граница_индекса1>`, `<верхняя_граница_индекса1>`, `<нижняя_граница_индекса2>`, `<верхняя_граница_индекса2>` – целые константы, определяющие диапазоны изменения индексов и, следовательно, число элементов массива;

`<тип>` – тип элементов массива.

Пример объявления двумерного массива:

`Product: array[1..3,1..4] of integer;`

Количество элементов массива может быть вычислено по формуле:

$$(\text{верхняя_граница_индекса1} - \text{нижняя_граница_индекса1} + 1) * (\text{верхняя_граница_индекса2} - \text{нижняя_граница_индекса2} + 1),$$

Таким образом, массив `Product` состоит из 12 элементов типа `integer` $((3-1+1)*(4-1+1)=3*4=12)$.

Приведем еще примеры описания двумерных массивов в Pascal-программах:

Type MATR=array[1..4,1..5] of integer;

Type B= array[2..9,0..6] of real;

Type C= array[-1..4,-1..4] of char.

Также допускается указание имени другого типа массива в качестве типа элементов массива, например:

Type VEC= array[1..4] of real;

MAS= array[1..5] of vec.

В результате приведенного выше описания тип массива MAS будет объявлен как тип двумерного массива, первый индекс которого будет меняться от 1 до 5, а второй индекс – от 1 до 4, т.е. размерность массива составит 5*4 элементов.

Чтобы использовать элемент массива, нужно указать имя массива и индексы элемента. Первый индекс обычно соответствует номеру строки таблицы, второй – номеру колонки. Двумерные массивы, в которых диапазоны индексов начинаются с 1, также называются иногда матрицами. Если число строк матрицы равняется числу столбцов, то матрицы такого вида называются квадратными. Элементы квадратной матрицы вида $B[1,1]$, $B[2,2]$, ..., $B[N,N]$ составляют главную диагональ матрицы. Иногда вводят понятие побочной диагонали квадратной матрицы, которую составляют элементы $B[1,N]$, $B[2,N-1]$, $B[3,N-2]$, ..., $B[N,1]$, где N – число строк (столбцов) матрицы.

При вводе и выводе значений элементов двумерных массивов используются вложенные циклы, в которых внешний оператор цикла, как правило, задает изменение строк массива, внутренний оператор цикла – изменение столбцов.

Пример использования вложенных циклов:

```
begin
  for i:=1 to 3 do
    for j:=1 to 4 do
      begin
        writeln('Введите элемент a[' , i , ' , ' , j , ' ]');
        readln(a[i,j]);
      end;
    end;
  end;
```

К типичным операциям с матрицами и двумерными массивами можно отнести: вывод массива; ввод массива; сортировка массива; поиск в массиве элементов удовлетворяющих определенному усло-

вию; поиск в массиве максимального или минимального элемента; нахождение суммы элементов строк и столбцов матрицы.

Содержание лабораторной работы

В лабораторной работе требуется разработать программу на языке программирования Turbo Pascal реализующую алгоритмы обработки двумерных массивов в соответствии с Вашим вариантом. Отчет о проделанной работе должен содержать: название и цель работы; номер варианта для выполнения задания и условие своего варианта; блок–схему решения задачи; тексты программ; полученные при расчетах численные результаты и выводы по проделанной лабораторной работе.

Варианты заданий

1. Разработать программу на языке программирования Turbo Pascal:
 - а) В матрице $c(8,5)$ поменять местами третий и пятый столбцы.
 - б) В матрице $d(8,8)$ заменить отрицательные элементы, расположенные выше главной диагонали, средним геометрическим положительных чисел.
2. Разработать программу на языке программирования Turbo Pascal:
 - а) В матрице $t(7,8)$ найти среднее арифметическое элементов, расположенных в четных столбцах.
 - б) Из элементов матрицы $a(5,5)$, удовлетворяющих условию $-5 \leq a(i, j) \leq 5$, построить вектор $b(25)$, заменив недостающие элементы нулями.
3. Разработать программу на языке программирования Turbo Pascal:
 - а) В матрице $a(8,7)$ поменять местами третий и шестой столбцы. Найти среднее арифметическое всех элементов матрицы.
 - б) Найти сумму положительных элементов строк матрицы $d(8,7)$. Результат поместить в вектор $c(8)$.
4. Разработать программу на языке программирования Turbo Pascal:
 - а) В матрице $a(6,8)$ найти количество положительных элементов.
 - б) Сложить две матрицы $a(7,3)$ и $b(7,3)$ и найти наибольший элемент в полученной сумме.
5. Разработать программу на языке программирования Turbo Pascal:

- а) В матрице $a(6,8)$ найти количество положительных и отрицательных элементов. Определить, каких элементов больше и на сколько.
- б) Сложить две матрицы $a(7,3)$ и $b(7,3)$, найти наибольший и наименьший элементы в полученной сумме. Найденные элементы поменять местами.
6. Разработать программу на языке программирования Turbo Pascal:
- а) В матрице $d(5,7)$ найти среднее арифметическое отрицательных элементов и количество положительных элементов.
- б) Из матрицы $t(7,8)$ выбрать положительные элементы и разместить их подряд в вектор $z(56)$. Если положительных элементов в z будет меньше 56, дополнить их +1.
7. Разработать программу на языке программирования Turbo Pascal:
- а) В матрице $c(7,7)$ найти количество положительных элементов, расположенных на главной диагонали.
- б) В матрице $d(8,6)$ найти максимальный и минимальный элементы и их координаты.
8. Разработать программу на языке программирования Turbo Pascal:
- а) В матрице $d(7,8)$ найти среднее арифметическое отрицательных элементов.
- б) Произвести сортировку матрицы $b(7,8)$, записав ее положительные элементы в вектор x (подряд), а отрицательные – в вектор y (подряд).
9. Разработать программу на языке программирования Turbo Pascal:
- а) В матрице $d(7,6)$ поменять местами третью и шестую строки.
- б) В матрице $a(6,6)$ выбрать все отрицательные элементы, расположенные выше главной диагонали, и поместить их в вектор z подряд.
10. Разработать программу на языке программирования Turbo Pascal:
- а) В матрице $d(7,7)$ найти количество положительных элементов, расположенных на главной диагонали.
- б) Найти суммы отрицательных элементов столбцов матрицы $t(6,8)$ и поместить результат в вектор m .
11. Разработать программу на языке программирования Turbo Pascal:

а) Найти среднее арифметическое отрицательных элементов матрицы $t(9,6)$.

б) В матрице $d(8,8)$ найти минимальный элемент среди элементов, расположенных ниже главной диагонали.

12. Разработать программу на языке программирования Turbo Pascal:

а) В матрице $t(7,5)$ поменять местами первый и последний столбцы, а затем вторую и первую строки.

б) В матрице $d(8,8)$ заменить отрицательные элементы, расположенные ниже главной диагонали, нулями, а положительные – единицами.

13. Разработать программу на языке программирования Turbo Pascal:

а) Найти среднее геометрическое положительных элементов матрицы $t(10,10)$.

б) В матрице $c(12,12)$ найти среднее арифметическое отрицательных элементов, расположенных выше побочной диагонали.

14. Разработать программу на языке программирования Turbo Pascal:

а) В матрице $t(6,7)$ найти сумму положительных элементов и сумму элементов третьей строки. Результат вывести с пояснительным текстом.

б) Произвести вычитание из матрицы $a(7,3)$ матрицы $c(7,3)$ и в полученной разности найти минимальный элемент и его номер.

15. Разработать программу на языке программирования Turbo Pascal:

а) В матрице $s(12,14)$ найти среднее арифметическое отрицательных элементов, а также сумму элементов шестой строки.

б) В матрице $d(9,9)$ найти количество положительных элементов, расположенных ниже побочной диагонали.

16. Разработать программу на языке программирования Turbo Pascal:

а) В матрице $k(3,8)$ найти среднее геометрическое положительных элементов и сумму элементов седьмого столбца.

б) Из матрицы $m(7,8)$ выбрать элементы, удовлетворяющие условию $k < m_{i,j} < l$, и поместить их в вектор $z(56)$. Если таких элементов меньше 56, то на оставшиеся места поместить 0.

17. Разработать программу на языке программирования Turbo Pascal:

а) В матрице $b(7,7)$ поменять местами первую и седьмую строки и найти среднее арифметическое положительных элементов.

б) Из отрицательных элементов вектора $d(9)$ сформировать матрицу $a(3,3)$. Если отрицательных элементов в d меньше 9, дополнить оставшиеся места – 1. Транспонировать полученную матрицу.

18. Разработать программу на языке программирования Turbo Pascal:

а) В матрице $d(5,7)$ найти среднее геометрическое положительных элементов, расположенных в четных строках.

б) Из элементов вектора $b(49)$, сформированных случайным образом, построить матрицу $d(7,7)$.

19. Разработать программу на языке программирования Turbo Pascal:

а) В матрице $d(8,3)$ найти среднее арифметическое положительных элементов каждого столбца и сформировать из них вектор.

б) Умножить матрицу $a(n,m)$ на вектор $b(m)$, где $n = 5$, $m = 6$. В полученном произведении произвести сортировку элементов по убыванию $(n,m) < 11$.

20. Разработать программу на языке программирования Turbo Pascal:

а) В матрице $t(7,8)$ поменять местами третью и пятую строки и найти среднее арифметическое элементов, расположенных в четных столбцах.

б) Записать на место отрицательных элементов матрицы $a(10,10)$ нули, а на место положительных – единицы. Вывести на печать нижнюю треугольную матрицу в общепринятом виде.

21. Разработать программу на языке программирования Turbo Pascal:

а) В матрице $d(6,8)$ поменять местами третий и пятый столбцы и найти среднее геометрическое элементов, расположенных под главной диагональю.

б) Рассортировать элементы матрицы $d(n,n)$, где $n=5$, так, чтобы в массиве x были расположены подряд только положительные и нулевые элементы, а в массиве y (поряд) – только отрицательные. Расположить элементы массивов x и y в порядке возрастания их величины.

22. Разработать программу на языке программирования Turbo Pascal:

а) Транспонировать матрицу $f(10,8)$ и вывести на печать элементы главной диагонали и диагонали, расположенной под главной. Результаты разместить в двух строках.

б) Найти среднее геометрическое положительных элементов каждого столбца матрицы $d(n,m)$. Поместить их в массив $x(n, m < 12)$.

23. Разработать программу на языке программирования Turbo Pascal:

а) Найти среднее геометрическое положительных элементов матрицы $d(n,n)$, где $n=6$, расположенных на побочной диагонали.

б) Из положительных элементов массивов $a(15)$ и $b(5)$ сформировать матрицу $c(4,5)$. Если положительных элементов в массиве будет меньше 20, заполнить свободные места числом +1.

24. Разработать программу на языке программирования Turbo Pascal:

а) В матрице $c(7,8)$ найти среднее арифметическое элементов каждой строки и поместить эти значения в массив $m(7)$.

б) Вычислить сумму и число положительных элементов матрицы $b(n,n)$, находящихся над главной диагональю ($n < 11$).

25. Разработать программу на языке программирования Turbo Pascal:

а) Произвести транспонирование матрицы $t(8,6)$ и поменять местами третью и шестую строки.

б) Для целочисленной матрицы найти для каждой строки число элементов, кратных 5, и наибольший из полученных результатов.

26. Разработать программу на языке программирования Turbo Pascal:

а) В матрице $a(9,9)$ поменять местами третий и пятый столбцы и найти среднее геометрическое положительных элементов, расположенных под главной диагональю.

б) В матрице $s(n,m)$ в каждой строке расположить элементы в порядке их возрастания $n, m < 10$.

27. Разработать программу на языке программирования Turbo Pascal:

а) Найти наибольший элемент матрицы $a(13,13)$, расположенный выше главной диагонали, и номера столбца и строки, где он находится.

б) Из положительных элементов вектора $z(60)$ сформировать матрицу $b(10,6)$. Если положительных элементов в векторе окажется меньше 60, дополнить оставшиеся места числом +1.

28. Разработать программу на языке программирования Turbo Pascal:
- а) Найти минимальный элемент матрицы $c(5, 7)$ и обнулить строку, где он располагается.
 - б) В матрице $k(8, 12)$ найти сумму отрицательных элементов каждой строки и разместить значения этих сумм в вектор d .
29. Разработать программу на языке программирования Turbo Pascal:
- а) Найти максимальный элемент матрицы $d(7, 7)$ и поместить в строку и столбец, где он находится, число +1.
 - б) Из положительных элементов матрицы $a(6, 6)$, расположенных выше главной диагонали, сформировать вектор m .
30. Разработать программу на языке программирования Turbo Pascal:
- а) Найти среднее арифметическое отрицательных элементов матрицы $c(7, 6)$ и заменить им все нули матрицы.
 - б) Из отрицательных элементов вектора $z(70)$ сформировать матрицу $k(7, 10)$. Если отрицательных элементов в векторе окажется меньше 70, дополнить оставшиеся места нулями.

Пример программы на языке Turbo Pascal

Известны результаты 5 студентов по итогам экзаменов по химии и информатике. Найти фамилии студентов, сдавших оба экзамена на отлично.

Схема алгоритма работы программы представлена на рисунке 11.

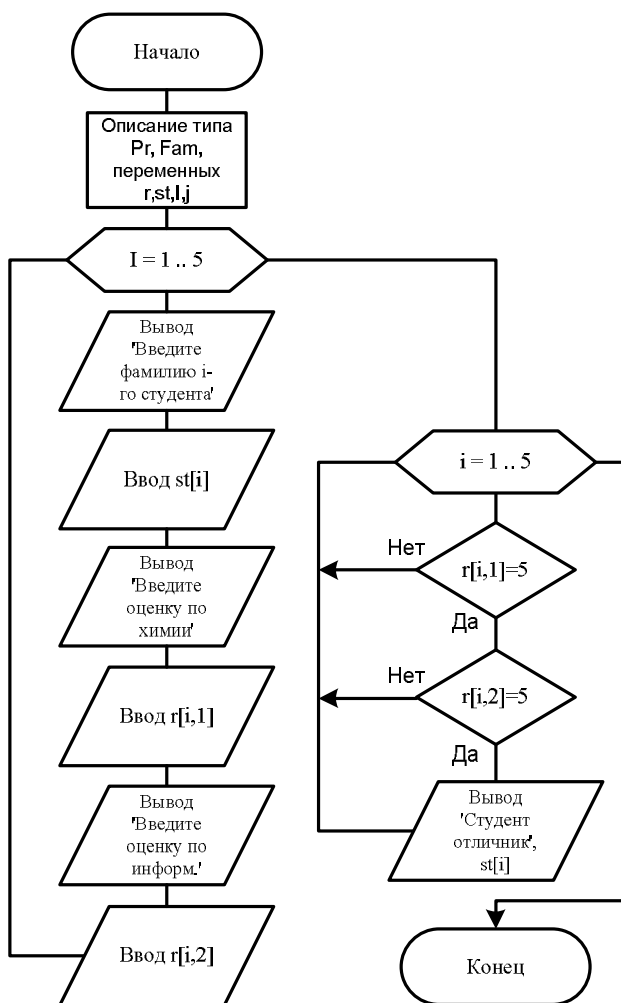


Рис. 11 – Схема алгоритма работы программы

```

Program Primer5;
  type pr=array [1..5,1..2] of integer;
  fam=array[1..5] of string[10];
var r:pr;
  st:fam;
  i,j:integer;
begin

```

```

for i:=1 to 5 do
  begin
    writeln('Введите фамилию ',i,'-го студента');
    readln(st[i]);
    writeln('Введите оценку данного студента по химии
    (от 2 до 5)');
    readln(r[i,1]);
    writeln('Введите оценку данного студента по информатике
    (от 2 до 5)');
    readln(r[i,2]);
  end;
for i:=1 to 5 do
  if (r[i,1]=5) and (r[i,2]=5) then
    writeln('Студент – отличник – ',st[i]);
end.

```

Контрольные вопросы

1. Какой массив называется двухмерным?
2. Приведите примеры двухмерных массивов.
3. Как может быть вычислено количество элементов двухмерного массива?
4. Дайте понятие квадратной матрицы, диагоналей квадратной матрицы.
5. Какие способы объявления многомерных массивов вы знаете?
6. Как обратиться к элементу многомерного массива?
7. Приведите пример описания двумерных массивов на языке Turbo Pascal.
8. Поясните порядок использования вложенных циклов при вводе элементов двумерного массива.
9. Какие стандартные алгоритмы по работе с двухмерными массивами Вы знаете?
10. Какими способами может быть заполнен двухмерный массив? Приведите примеры.
11. Как будет выглядеть блок схема для решения задачи поиска элемента двухмерного массива удовлетворяющего определенному условию?
12. Как будет выглядеть блок схема для решения задачи поиска в двухмерном массиве максимального или минимального элемента?

13. Как будет выглядеть блок схема для решения задачи нахождения суммы элементов строк матрицы?
14. В каких случаях допускается обращение к многомерному массиву целиком?
15. Какая конструкция применяется для обработки n -мерного массива.

Лабораторная работа № 6.

Процедуры и функции

Цель работы: изучение принципов работы с процедурами и функциями на языке программирования Turbo Pascal. Получение навыков по подготовке, редактированию, компиляции и выполнению программ, использующих процедуры Procedure и функции Function.

Теоретические сведения

Как известно, синтаксически Turbo Pascal программа состоит из необязательного заголовка и блока. Блоки могут быть: глобальные и локальные.

Глобальный блок – это основная программа, он должен быть в любом случае. *Локальные блоки* – это процедуры и функции, их присутствии необязательно. Соответственно объекты программы (типы, константы, переменные и т.д.) глобального блока называются глобальными, а локального блока локальными.

Часто при разработке программ возникает необходимость в реализации одного и того же алгоритма в разных точках программы и с разными исходными данными. Чтобы избежать повторения, алгоритм составляется пользователем один раз и оформляется в виде процедур или функций. Затем этот алгоритм вызывается в соответствующих (нужных) местах программы. Использование процедур и функций позволяет:

- сократить объем и улучшить структуру программы;
- уменьшить вероятность ошибок и облегчить процесс отладки программы;
- дает возможность выполнить разработку программы разным программистам.

На языке Turbo Pascal процедуры и функции записываются зарезервированными словами Procedure и Function соответственно. Описание процедур и функций осуществляется в разделе описаний данных.

Описание процедуры:

Procedure <идентификатор процедуры>

(<список формальных параметров с объявлением типа>);

var <раздел описаний>; {локальные данные}

begin {начало локального блока}

<раздел операторов>

end; {конец локального блока}

Допускается описание процедур без параметров.

Описание функции:

```
Function <идентификатор функции> ( <список формальных
параметров с объявлением типа> ) : <тип функции–результата>;
  var <раздел описаний>; {локальные данные}
  begin {начало локального блока}
    <раздел операторов>
  end; {конец локального блока}
```

В разделе операторов должен находиться оператор присваивания:

<идентификатор функции> := <значение–ответ>;

Описание процедуры или функции само по себе никакого действия не вызывает. Чтобы использовать процедуру или функцию, необходимо в нужной точке программы поместить обращение к ней. Записать специальный оператор вызова процедуры в следующем виде:

<идентификатор процедуры> (<список фактических параметров>);

Вызов функции осуществляется по имени функции аналогично обращению, например к математической функции `sin x`:

`x:= 0.5;`

`a:=sin (x);`

или

`a:=sin (0.5);`

Вызов `sin x` осуществляется оператором присваивания.

Между фактическими и формальными параметрами должно быть соответствие: по типу, количеству, порядку следования. В качестве фактических параметров могут выступать константы, выражения, переменные, структурированные данные. Константы и выражения не могут быть формальными параметрами.

Формальные параметры могут быть заданы как:

- параметры–значения (передача параметров по значению);
- параметры–переменные (передача параметров по ссылке);

Передача параметров по значению обеспечивает сохранность величины передаваемого параметра. Изменение их в процедуре никак не отразится на фактических параметрах.

Передача параметров по ссылке при изменении формального параметра обеспечивает изменение фактического параметра. Параметры

переменные записываются в списке формальных параметров после зарезервированного слова **Var**.

Содержание лабораторной работы

В лабораторной работе требуется разработать программу на языке программирования Turbo Pascal использующую процедуры и функции в соответствии с Вашим вариантом. Если в задаче не указан способ задания переменной, то она вводится пользователем с клавиатуры. Отчет о проделанной работе должен содержать: название и цель работы; номер варианта для выполнения задания и условие своего варианта; блок-схему решения задачи; тексты программ; полученные при расчетах численные результаты и выводы по проделанной лабораторной работе.

Варианты заданий

1. Решить уравнение $(ba)!x^2 + 2(dc)!x + (mk)! = 0$, где b, a – сумма модулей и количество элементов, расположенных в матрице $z(6,6)$ ниже побочной диагонали; d, c – сумма модулей и количество элементов, расположенных в матрице $w(11,11)$ ниже побочной диагонали; m, k – сумма модулей и количество элементов, расположенных в матрице $v(12,12)$ ниже побочной диагонали. Знакопеременные массивы сформировать случайным образом.
2. Решить показательное уравнение $e^{\frac{d_1}{t_2}x} - \frac{d_2}{t_1} = 0$, где d_1, t_1 – количество столбцов, не содержащих нулей и сумма модулей членов этих столбцов в матрице $a(8,6)$; b_2, t_2 – количество столбцов, не содержащих нулей и сумма модулей членов этих столбцов в матрице $b(12,8)$. Знакопеременные массивы сформировать случайным образом.

3. Вычислить значения функции $z = \frac{\cos \sum_{i=1}^{20} |a_i| + \sin \sum_{i=1}^{30} |b_i|}{\ln \left(\sum_{i=1}^{20} |a_i| \right) x}$, если x изменяется от x_n до x_k с шагом h . Знакопеременные массивы $a(20)$ и $b(30)$ сформировать случайным образом.
4. Вычислить таблицу значений функции $y = \frac{ne^{-m!} - le^{-s!}}{kl^{-(m!+c!)}} x$, где m, n – количество и сумма отрицательных элементов матрицы $w(11,11)$; s, l – количество и сумма отрицательных элементов матрицы $v(10,10)$; c, k – количество и сумма отрицательных элементов матрицы $t(8,8)$. Знакопеременные массивы сформировать случайным образом.
5. Вычислить таблицу значений функции $z = \frac{s_1 s_2}{(k_1 + k_2) x}$, где s_1, k_1 – сумма и количество отрицательных элементов в матрице $d(10,10)$, стоящих под главной диагональю; s_2, k_2 – сумма и количество отрицательных элементов в матрице $c(8,8)$, стоящих под главной диагональю; x изменяется от x_n до x_k с шагом h . Знакопеременные массивы сформировать случайным образом.
6. Вычислить таблицу значений функции $z = \frac{(x^{s_1} + y^{s_2})}{2k_1 k_2}$, где s_1, k_1 – сумма и количество отрицательных элементов массива $a(50)$; s_2, k_2 – сумма и количество отрицательных элементов массива $b(70)$; x – фиксированное число, вводимое с клавиатуры, а y изменяется от 1 до 3 с шагом 0,5. Знакопеременные массивы сформировать случайным образом.
7. Вычислить таблицу значений функции $z = \frac{q_1 + q_2}{2n_1 n_2} x$, где q_1, n_1 – среднее геометрическое и количество положительных элементов

массива $a(60)$; q_2, n_2 – среднее геометрическое и количество положительных элементов массива $b(40)$; x изменяется от x_n до x_k с шагом h . Знакопеременные массивы сформировать случайным образом.

8. Вычислить таблицу значений функции $y = \frac{k!x^a + (k+m)x^b}{((kn)!x^c)}$, где k, a – количество и среднее арифметическое положительных элементов матрицы $d(11,11)$, расположенных ниже побочной диагонали; m, b – количество и среднее арифметическое положительных элементов матрицы $t(10,10)$, расположенных ниже побочной диагонали; n, c – количество и среднее арифметическое положительных элементов матрицы $l(12,12)$, расположенных ниже побочной диагонали; x изменяется от x_n до x_k с шагом h . Знакопеременные массивы сформировать случайным образом.

9. Вычислить таблицу значений функции $z = \frac{a^{x_{\min}} b^{y_{\min}}}{an_{\min} + bm_{\min}}$, где x_{\min}, n_{\min} – минимальный элемент и его номер в массиве $x(200)$; y_{\min}, m_{\min} – минимальный элемент и его номер в массиве $y(100)$; a – фиксированное число, вводимое с клавиатуры; b изменяется от b_n до b_k , с шагом h . Знакопеременные массивы сформировать случайным образом.

10. Вычислить таблицу значений функции $z = \frac{(a^2 + ab + b^2)x + (\sin^2 c + \sin c * tgd + tg^2 d)x^2}{2\left(\frac{k^2}{4} + \frac{tk}{4} + \frac{k^2}{4}\right)}$, где a, b – ко-

личество и сумма отрицательных элементов в матрице $q(8,10)$; c, d – количество и сумма отрицательных элементов в матрице $m(12,10)$; t, k – количество и сумма отрицательных элементов в

матрице $l(15,13)$; x изменяется от x_n до x_k с шагом h . Знакопеременные массивы сформировать случайным образом.

11. Вычислить значение функции $z = \frac{x^{s_1} + x^{s_2}}{k_1 + k_2}$, где s_1, k_1 – среднее

арифметическое и количество положительных элементов массива $a(100)$; s_2, k_2 – среднее арифметическое и количество положительных элементов массива $b(80)$. x изменяется от x_n до x_k с шагом h . Знакопеременные массивы сформировать случайным образом.

12. Решить тригонометрическое уравнение $\sin\left(\frac{a_1}{b_1}\right)x - \frac{n_1}{n_2} = 0$ на ин-

тервале $[0;10]$, где a_1, n_1 – сумма и количество отрицательных элементов, стоящих в массиве $d(30)$ на местах, номера которых кратны 4; b_1, n_2 – сумма и количество отрицательных элементов, стоящих в массиве $t(40)$ на местах, номера которых кратны 4. Знакопеременные массивы сформировать случайным образом.

13. Решить показательное уравнение $e^{-\frac{d_1}{b_1}x} - e^{\frac{d_2}{b_2}} = 0$, где d_1, b_1 – сумма модулей членов и количество строк в матрице $a(12,10)$, не содержащих отрицательных значений; d_2, b_2 – сумма модулей членов и количество строк в матрице $t(8,16)$, не содержащих отрицательных значений. Знакопеременные массивы сформировать случайным образом.

14. Вычислить таблицу значений функции

$$y = \frac{\sqrt{|a^2 - b^2|x} + \sqrt{|tg^2 d + \sin^2 c|}}{2\sqrt{|t^2 - k^2|}}, \text{ где } a, b \text{ – сумма и количество}$$

положительных элементов в матрице $q(6,6)$, расположенных на главной диагонали; d, c – сумма и количество положительных элементов в матрице $m(11,11)$, расположенных на главной диаго-

нали; t, k – сумма и количество положительных элементов в матрице $n(20, 20)$. Расположенных на главной диагонали; x изменяется от x_n до x_k с шагом h . Знакопеременные массивы сформировать случайным образом.

15. Вычислить таблицу значений функции
$$y = \frac{2|a|!x^k + 2|a-b|!x^m}{|c-a-b|!x^n},$$

где a, k – целая часть модуля суммы и количество отрицательных элементов матрицы $d(11, 11)$, расположенных выше побочной диагонали; b, m – целая часть модуля суммы и количество отрицательных элементов матрицы $t(12, 12)$, расположенных выше побочной диагонали; c, n – целая часть модуля суммы и количество отрицательных элементов матрицы $l(14, 14)$, расположенных выше побочной диагонали; x изменяется от x_n до x_k с шагом h . Знакопеременные массивы сформировать случайным образом.

16. Решить уравнение $a^n x^2 + bmx + am = 0$, где a, n – сумма и количество положительных элементов в матрице $d(4, 4)$, находящихся на главной диагонали; b, m – сумма и количество положительных элементов в матрице $t(8, 8)$, находящихся на главной диагонали. Знакопеременные массивы сформировать случайным образом.

17. Решить уравнение $\frac{m}{a}x^2 + \frac{n}{b}x = 0$, где m, a – количество и сумма отрицательных элементов матрицы $c(7, 7)$, расположенных выше главной диагонали; n, b – количество и сумма отрицательных элементов матрицы $d(8, 8)$, расположенных выше главной диагонали. Знакопеременные массивы сформировать случайным образом.

18. Решить уравнение $\frac{a}{m}x^2 + \frac{b}{d}x + \frac{c}{n} = 0$, где a, n – среднее арифметическое и количество положительных элементов массива $t(15)$, удовлетворяющих условию $k < t_i < z$; b, m – среднее арифмети-

ческое и количество положительных элементов массива $d(12)$, удовлетворяющих условию $k < d_i < I$; c, d – среднее арифметическое и количество положительных элементов массива $r(10)$, удовлетворяющих условию $k < l_i < s$; k, s, z, I – числа, вводимые с клавиатуры. Знакопеременные массивы сформировать случайным образом.

19. Решить квадратное уравнение $\left(\frac{a}{n}\right)x^2 + \left(\frac{b}{m}\right)x + \left(\frac{c}{l}\right) = 0$, где a, n – сумма и количество положительных элементов в матрице $d(10,10)$; b, m – сумма и количество положительных элементов в матрице $x(8,8)$; c, l – сумма и количество положительных элементов в матрице $y(6,6)$. Знакопеременные массивы сформировать случайным образом.

20. Решить уравнение $\frac{c}{n} + \frac{b}{l} = \frac{a}{m}$, где l, c – количество и сумма отрицательных элементов матрицы $d(8,8)$, расположенных выше главной диагонали; n, a – количество и сумма отрицательных элементов матрицы $t(10,10)$, расположенных выше главной диагонали; m, b – количество и сумма отрицательных элементов матрицы $q(7,7)$, расположенных выше главной диагонали. Знакопеременные массивы сформировать случайным образом.

21. Вычислить таблицу значений функции $z = \frac{x^{s_1} + x^{s_2}}{k_1 - k_2}$, где s_1, k_1 – сумма и количество положительных элементов матрицы $t(6,6)$, расположенных ниже главной диагонали; s_2, k_2 – сумма и количество положительных элементов матрицы $t(6,6)$, расположенных ниже главной диагонали; x изменяется от x_n до x_k с шагом h . Знакопеременные массивы сформировать случайным образом.

22. Вычислить таблицу значений функции $z = \frac{ax_{\max} + by_{\max}}{n_{\max} + m_{\max}}$, где

x_{\max}, n_{\max} – максимальный элемент и его номер в массиве $x(70)$;

y_{\max}, m_{\max} – максимальный элемент и его номер в массиве $y(100)$;

a, b – фиксированное число, вводимое с клавиатуры. Знакопеременные массивы сформировать случайным образом.

23. Решить показательное уравнение $e^{\left(\frac{2d_1}{c_1}\right)x} - \frac{d_2}{2c_2} = 0$, где d_1, c_1 –

сумма и количество положительных элементов в матрице $a(7,6)$;

d_2, c_2 – сумма и количество отрицательных элементов в матрице $a(7,6)$. Знакопеременные массивы сформировать случайным образом.

24. Решить тригонометрическое уравнение $n_1 \sin\left(\frac{a_1}{n_2}\right)x - \frac{a_2}{n_1} = 0$ на

интервале $[0;15]$, где n_1, a_1 – количество и среднее геометрическое положительных элементов массива $t(32)$; n_2, a_2 – количество и среднее геометрическое положительных элементов массива $w(20)$. Знакопеременные массивы сформировать случайным образом.

25. Решить тригонометрическое уравнение $2n_2a_1 - n_2 \operatorname{tg}\left(a_2 \frac{x}{n_1}\right) = \frac{n_3}{a_2}$

на интервале $[5;25]$, где n_1, a_1 – количество и среднее арифметическое отрицательных элементов массива $t(30)$, стоящих на чётных местах; n_2, a_2 – количество и среднее арифметическое отрицательных элементов массива $d(25)$, стоящих на чётных местах; n_3, a_3 – количество и среднее арифметическое отрицательных элементов массива $b(28)$, стоящих на чётных местах. Знакопеременные массивы сформировать случайным образом.

26. Решить показательное уравнение $e^{\frac{d_1}{t_2}} - e^{\frac{d_2}{t_1}} = 0$, где t_1, d_1 – количество и сумма положительных элементов массива $a(50)$, удовлетворяющих условию $0 < a_i < m$; t_2, d_2 – количество и сумма положительных элементов массива $b(40)$, удовлетворяющих условию $0 < b_i < n$; m и n – положительные числа, вводимые с клавиатуры. Знакопеременные массивы сформировать случайным образом.

27. Решить показательное уравнение $5e^{\left(\frac{n_1}{a_1}\right)^x} + 2\frac{n_2}{a_2} = 0$, где n_1, a_1 – количество и сумма положительных элементов матрицы $d(8,8)$, стоящих ниже побочной диагонали; n_2, a_2 – количество и сумма положительных элементов матрицы $t(10,10)$, стоящих ниже побочной диагонали. Знакопеременные массивы сформировать случайным образом.

28. Вычислить значение функции $y = bn + 2nma$, где b, n – сумма и количество отрицательных элементов массива $c(30)$; a, m – сумма и количество отрицательных элементов массива $d(10)$. Знакопеременные массивы сформировать случайным образом.

29. Решить уравнение $\frac{a}{m}x^2 + \frac{c}{n}x + \frac{d}{t} = 0$, где a, m – сумма и количество положительных элементов массива $f(30)$; c, n – сумма и количество положительных элементов массива $l(20)$; d, t – сумма и количество положительных элементов массива $k(10)$. Знакопеременные массивы сформировать случайным образом.

30. Вычислить таблицу значений функции $y = \frac{a!x^n + (a+b)!x^k}{(ac)!x^m}$, где a, n – количество и среднее арифметическое положительных элементов матрицы $d(10,6)$; b, k – количество и среднее арифметическое положительных элементов матрицы $t(8,7)$; c, m – количе-

ство и среднее арифметическое положительных элементов матрицы $l(12,10)$; x изменяется от x_n до x_k с шагом h . Знакопеременные массивы сформировать случайным образом.

Пример программы на языке Turbo Pascal

При известных значениях переменных a , b , c и натуральных чисел n , m вычислить значения функции: $y = \frac{a^n + b^{m+1}}{c^{10}}$.

Для вычисления y необходимо три раза определить функцию заданной степени при заданном основании, причем основания и степени разные. Чтобы не выполнять один и тот же алгоритм, оформим его процедурой или функцией.

Схема алгоритма работы программы представлена на рисунке 12.

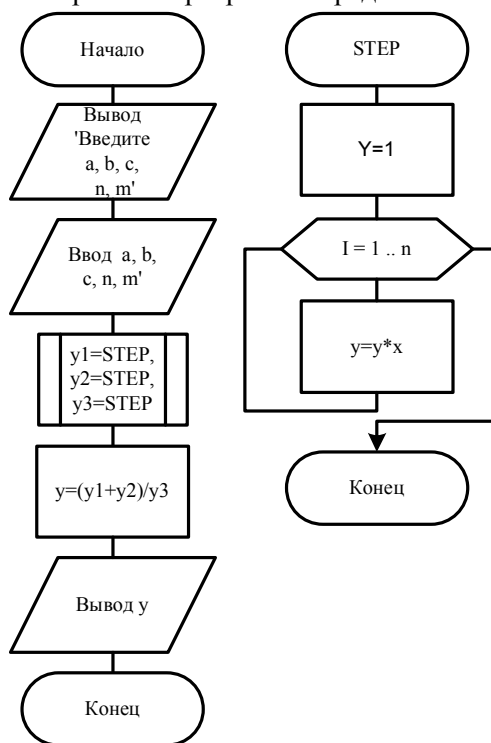


Рис. 12 – Схема алгоритма работы программы

а) программа с процедурой Procedure

```

Program Primer6a;
var a, b, c, y, y1, y2, y3:real;
n, m: byte; {Глобальное описание данных}
  procedure step (x:real; n:byte; var y:real);
    {Объявление процедуры}
  var i:byte; {Локальное описание данных}
  begin {Начало локального блока}
    y:=1;
    for i:=1 to n do y:=y*x;
  end; {Конец локального блока}
begin {Начало глобального блока}
  writeln ('Введите исходные данные a, b, c, n, m');
  readln (a, b, c, n, m);
  step (a, n, y1); {Обращение к процедуре}
  step (b, m+1, y2); {Обращение к процедуре}
  step (c, 10, y3); {Обращение к процедуре}
  y:= (y1+y2)/y3;
  writeln ('y=', y:8:4);
end. {Конец глобального блока}

```

Схема алгоритма работы программы представлена на рисунке 13.

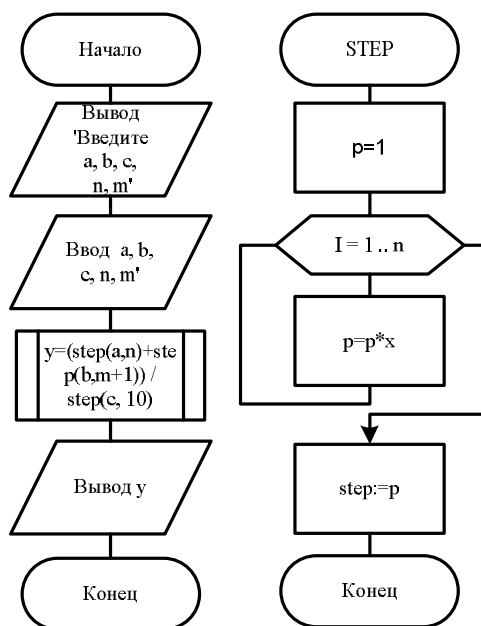


Рис. 13 – Схема алгоритма работы программы

б) программа с процедурой Function

Program primer6a;

var a, b, c, y, y1, y2, y3:real;

n, m: byte; {Глобальное описание данных}

function step (x:real; n:byte) : real; {Объявление функции}

var p:real; i:byte; {Локальное описание данных}

begin {Начало локального блока}

p:=1;

for i:=1 to n do p:=p*x;

step:=p; {Имени функции присваивается
значение результата}

end; {Конец локального блока}

begin {Начало глобального блока}

writeln ('введите исходные данные a, b, c, n, m');

readln (a, b, c, n, m);

y:=(step(a,n)+step(b,m+1))/step(c, 10);

writeln ('y=', y:8:4);

end. {Конец глобального блока}

Контрольные вопросы

1. Какие алгоритмы называют вспомогательными?
2. Какое количество вспомогательных алгоритмов может присутствовать в основном алгоритме?
3. Можно ли вспомогательные алгоритмы, написанные для решения данной задачи, использовать при решении других задач?
4. Для чего нужны процедуры и функции пользователя?
5. Что такое “глобальный блок” и “глобальные объекты” и “локальный блок”, “локальные объекты”?
6. Как описываются в программе процедуры?
7. Как описываются в программе функции?
8. Чем отличается описание процедур от описания функций?
9. В каких случаях целесообразно использовать функции?
10. Какого типа может быть значение функции?
11. Чем отличаются фактические параметры от формальных?
12. Почему, если в функции используются параметры–переменные, необходимо преобразовать её в процедуру?
13. Как вызываются процедуры, функции?
14. Как задаются (передаются) формальные параметры?
15. Какой вариант, представленной в качестве примера программы, более рационален а) или б) и почему?

Лабораторная работа № 7.

Файлы и их применение в Turbo Pascal

Цель работы: ознакомление с разновидностями файлов в Pascal. Принципы организации и порядок работы с типизированными и нетипизированными файлами. Организация текстовых файлов, операции чтения и записи текстовых файлов.

Теоретические сведения

В Pascal под файлом понимается именованная область памяти на долгосрочном запоминающем устройстве, например, на жестком диске или дискете. Выделяют три типа файлов: текстовые, типизированные, нетипизированные.

Текстовый файл – содержит строки текста различной длины, чтение или запись осуществляется (как правило) построчно.

Типизированный файл – с точки зрения Turbo Pascal содержит элементы одного типа, единицей обмена информацией является один элемент.

Нетипизированный файл – содержит любые данные, Turbo Pascal никак не интерпретируемые.

Типизированные файлы

Это файл, содержащий компоненты одного конкретного типа. Это могут быть простые или сложные типы данных Turbo Pascal (числа, строки, символы, массивы, записи и т.д.). Объявление типизированного файла выглядит следующим образом:

```
Type <имя_типа> = file of <тип>;
```

```
Var
```

```
    <имя> : <имя_типа>;
```

или

```
Var
```

```
    <имя> : file of <тип>;
```

Здесь:

<имя_типа> – имя файлового типа; <тип> – любой тип данных Pascal, кроме файлов; <имя> – имя файловой переменной.

После объявления файловой переменной для работы с файлом необходимо связать ее с реальным файлом на диске. Это выполняется процедурой Assign:

```
Assign ( <имя>, ' <путь_к_файлу> ' );
```

Первый параметр – имя файловой переменной, используемой в дальнейшем для обращения к файлу. Может иметь любой файловый тип. Второй параметр – строка, указывающая на путь и имя файла, с которым Turbo Pascal далее будет оперировать.

Для создания нового файла на диске после выполнения процедуры **Assign** необходимо открыть файл в режиме записи: **ReWrite (<имя>);**

Если файл с таким именем не существовал, он создается. Если он существовал, вся информация из этого файла стирается и файл становится пустой. При этом никаких предупреждений не выдается.

Для открытия файла в режиме чтения он должен быть открыт процедурой: **Reset(<имя>);**

После этого можно выполнять чтение или запись, используя соответствующую файловую переменную:

```
Read(F, i, j);
```

```
Write(F, n);
```

В первом примере из файла, связанного с переменной F, считывается два значения, во втором – записывается одно.

Для определения позиции указателя – порядкового номера компонента файла, который будет обрабатываться следующей операцией ввода-вывода, применяется функция **FilePos**. Например:

```
N := FilePos(F);
```

Эта функция возвращает переменной N положение указателя относительно начала файла. Самая первая позиция имеет номер 0. Переместить указатель в требуемую позицию можно, используя функцию **Seek**. Например:

```
Seek(F, 5);
```

```
Seek(F, FilePos(F)-1);
```

Первый оператор устанавливает позицию указателя в файле после 5 элемента (и перед 6-м). Следующий оператор вызывает смещение указателя файла на 1 элемент назад относительно текущего положения указателя.

Дополнительно для проверки положения указателя есть функция **EOF**. Эта функция возвращает логическое значение, показывающее, находится ли указатель файла после последнего элемента файла. Чаще всего эта функция применяется в задачах полного перебора элементов файла:

```
While not EOF(F) do Read(F, i);
```

Этот оператор считывает элементы файла, пока не доберется до конца файла.

Общее число компонент файла можно определить, вызвав функцию **FileSize**. Например:

```
Count := FileSize(F);
```

Эта функция возвращает целое число, равное количеству компонент в указанном файле.

Для закрытие файла следует воспользоваться следующей процедурой:

```
Close ( <имя> );
```

При этом связь файловой переменной, с именем файла установленная процедурой **Assign** сохраняется.

Текстовые файлы

Текстовый файл содержит строки текста различной длины. Каждая строка завершается символами «Возврат каретки» (код 13) и «Перевод строки» (код 10). При обычном просмотре эти символы на экране не отображаются, а интерпретируются компьютером как признак окончания одной строки и начала следующей.

Объявление текстового файла выглядит следующим образом:

```
Var
```

```
<имя> : Text;
```

Как и в случае других типов файлов, перед работой с текстовым файлом, предварительно необходимо связать файловую переменную с файлом на диске процедурой **Assign**.

При работе с текстовым файлом разрешается либо чтение, либо запись. Открытие файла для чтения или записи выполняется аналогично типизированным файлам процедурами **ReSet** или **ReWrite**. Для текстовых файлов, помимо процедур **ReSet/ReWrite**, определена еще одна процедура открытия:

```
Append(<имя>);
```

Данная процедура открывает файл для записи, при этом новые данные добавляются в конец файла, прежнее содержимое файла сохраняется (в отличие от процедуры **ReWrite**).

Далее в зависимости от способа открытия файла можно либо читать из него данные:

```
Read(F, i, j); ReadLn(F, s);
```

Либо записывать в него данные:

```
Write(F, a, b, c); WriteLn(F, Result);
```

При чтении/записи данных применяются те же правила, что и при вводе/выводе с клавиатуры/на экран, т.е. можно использовать форматирование:

```
WriteLn(F, x:5:2);
```

В данном примере *F* – имя файловой переменной; *x* – выводимое выражение; *5* – параметр, указывающий минимальную ширину поля, в которое будет записываться символьное представление; *2* – задает количество десятичных знаков в дробной части вещественного числа.

Нетипизированные файлы

Нетипизированные файлы очень похожи на типизированные, за исключением того, что при работе с ними можно сразу считать или записать несколько элементов файла.

Объявление нетипизированного файла:

```
Var
```

```
  <имя> : File;
```

Открытие нетипизированного файла выполняется процедурами **ReSet** или **ReWrite**. При этом может указываться дополнительный параметр, который указывает размер одного блока данных в байтах:

```
ReSet(F, 25);
```

или

```
ReWrite(F, SizeOf(Integer)*12);
```

В первом примере открывается файл для чтения/записи с размером блока 25 байт. Во втором случае размер блока указывается через размер типа **Integer**. Функция **SizeOf** возвращает количество байт, отводимых под указанный тип или переменную. Если размер блока данных опущен, принимается размер по умолчанию, равный 128 байт.

Чтение данных выполняется следующей процедурой:

```
BlockRead(F, Biffer, Count);
```

```
BlockRead(F, Biffer, Count, Result);
```

В примерах из файла *F* в область памяти, задаваемую переменной **Buffer**, считывается **Count** блоков данных. Во втором случае в переменную **Result** дополнительно заносится число реально считанных блоков данных. Такая ситуация возможна в тех случаях, когда конец файла встречается раньше, чем будет считано заданное число блоков. Если значения **Count** и **Result** различаются, это говорит о неполном чтении.

Аналогичные правила действуют для записи данных в файл:

```
BlockRead(F, Biffer, Count);
```

BlockRead(F, Biffer, Count, Result);

Частичная запись возможна в том случае, если на диске заканчивается свободное место.

Содержание лабораторной работы

В лабораторной работе требуется разработать программу на языке программирования Pascal 7.0 использующую принципы организации и порядка работы с файлами в соответствии с Вашим вариантом. Отчет о проделанной работе должен содержать: название и цель работы; номер варианта для выполнения задания и условие своего варианта; блок–схему решения задачи; тексты программ; полученные при расчетах численные результаты и выводы по проделанной лабораторной работе.

Варианты заданий

1. Организовать простейшую базу данных по студентам группы. Сведения о студенте включают: ФИО, год рождения, пол, средний балл. Обеспечить ввод данных, редактирование, вывод на экран. Информацию хранить в типизированном файле.
2. Запись содержит данные о студенте: фамилию и инициалы, номер курса, номер группы, оценки, полученные в сессию. Получить списки студентов заданной группы, претендующих на повышенную стипендию и студентов-задолжников.
3. Дан текстовый файл. Считая, что количество букв в одном слове не превосходит 20, определить, сколько в файле имеется слов, состоящих из одного, двух, трех и т.д. символов. Результат вывести в другой текстовый файл.
4. Запись содержит данные о студенте: фамилию и инициалы, номер курса, номер группы, оценки, полученные в сессию. Для каждого студента заданной группы определить средний балл, а для всей группы вычислить средний балл по каждому предмету.
5. Сведения об автомобиле состоят из его марки, номера и фамилии владельца. Создать файл, содержащий сведения о нескольких автомобилях, после чего определить фамилии владельцев и номера автомобилей заданной марки. Марка автомобиля вводится пользователем.
6. Запись содержит данные о студенте: фамилию и инициалы, номер курса, номер группы, аттестацию (0 или 1) по каждому предмету.

Получить списки неаттестованных студентов заданной группы по каждому предмету.

7. Сведения об автомобиле состоят из его марки, номера и фамилии владельца. Создать файл, содержащий сведения о нескольких автомобилях, после чего определить количество автомобилей каждой марки.
8. Запись содержит данные о студенте: фамилию и инициалы, номер курса, номер группы, количество часов, пропущенных по уважительной и неуважительной причине. Получить списки студентов заданной группы, пропустивших занятия отдельно по уважительной и неуважительной причинам с указанием часов пропуска.
9. Дан текстовый файл, содержащий программу на языке Паскаль. Проверить эту программу на соответствие числа открывающих и закрывающих круглых скобок.
10. Запись содержит данные о студенте: фамилию и инициалы, номер курса, номер группы, количество часов, пропущенных по уважительной и неуважительной причине. Для каждого студента заданной группы определить суммарное число пропущенных часов и выдать список студентов этой группы, пропустивших более 30 часов.
11. Дан текстовый файл. Записать все строки файла в новый файл, изменив порядок следования букв в каждой строке на противоположный.
12. Запись содержит данные о студенте: фамилию и инициалы, номер курса, номер группы, сведения о месте проживания (дома, в общежитии или на квартире). Получить списки студентов заданной группы отдельно по каждому месту проживания.
13. Создать файл, содержащий не более 100 случайных целых чисел. Выполнить сортировку чисел по возрастанию.
14. Запись содержит данные о студенте: фамилию и инициалы, номер курса, номер группы, экзаменационную оценку и дату сдачи экзамена. Получить списки студентов заданной группы, сдавших экзамен на отлично (5), хорошо (4), удовлетворительно (3), не сдавших экзамен (2) и не явившихся на экзамен (0).
15. Создать файл, содержащий не более 100 случайных целых чисел. Создать новый файл, разместив все нечетные числа в начале файла, а четные – в конце, при этом порядок следования чисел сохраняется.

16. Запись содержит сведения о результатах медосмотра: фамилия и инициалы, рост, вес. Для каждого человека определить индекс Кетле и выдать списки людей с нормальным, малым и избыточным весом.

Указание к решению

Индекс Кетле определяется по формуле $\frac{\text{вес (кг)}}{\text{рост}^2 \text{ (м)}}$

Индекс Кетле	Показатели веса
до 19,5	малый вес
от 19,5 до 24,9	нормальный вес
от 25 и выше	избыточный вес

17. Запись содержит данные о работнике: фамилия и инициалы, табельный номер, номер подразделения, должность, стаж работы, величина зарплаты. Получить списки работников, заданного подразделения, стаж которых не менее 10, 20, 25 лет.
18. Дан текстовый файл, содержащий строки произвольной длины. Отформатировать текст и записать его в новый файл так, чтобы все строки имели одинаковую длину, равную длине самой длинной строки. Форматирование выполняется добавлением пробелов между словами.
19. Запись содержит данные о работнике: фамилия и инициалы, табельный номер, номер подразделения, должность, стаж работы, величина зарплаты. Определить для каждого работника заданного подразделения сумму к выдаче (в простейшем случае это 87 процентов от зарплаты) и общую сумму заработка работников этого подразделения.
20. Дан текстовый файл, содержащий сведения о студентах группы. Сведения включают ФИО, год рождения, оценки по 4 экзаменам за последний семестр. Сведения об одном студенте находятся в одной строке, отделены друг от друга точкой с запятой. Считать сведения, записать в типизированный файл соответствующего типа, вывести на экран ФИО студента с максимальным средним баллом.
21. Запись содержит данные о работнике: фамилия и инициалы, табельный номер, номер подразделения, количество отработанных часов за месяц, стоимость одного часа работы. Определить для каждого работника заданного подразделения сумму заработка и общую сумму для этого подразделения.

22. Запись содержит данные о товаре: наименование, артикул (штрих-код), цена за единицу, количество. Для каждого товара определить общую сумму и получить сумму всех товаров. Выдать списки товаров, стоимость (цена) которых выше и ниже средней.
23. Запись содержит данные о товаре: наименование, артикул (штрих-код), количество единиц товара, количество проданного товара за день. Для каждого товара получить остаток на конец дня. Выдать списки распроданных товаров и товаров, не пользующихся спросом.
24. Запись содержит сведения о перевозках авиапассажиров на рейсах аэропорта: номер рейса, маршрут, марка самолета, общие затраты на рейс, количество пассажиров. Подсчитать стоимость перевозки одного пассажира на рейсе. Для каждой марки самолета выдать список с указанием затрат, количества пассажиров и стоимости перевозки одного пассажира.
25. Запись содержит сведения о результатах медосмотра: фамилия и инициалы, рост, вес. Для каждого человека определить идеальный вес и выдать списки людей, чей вес близок к идеальному (± 3 кг), и тех, у кого он избыточен.

Указание к решению

Идеальный вес определяется в зависимости от роста:

рост (см)	идеальный вес (кг)
до 155	вес – 50
до 165	вес – 60
до 175	вес – 70
свыше 175	вес – 85

26. Запись содержит сведения о перевозках авиапассажиров на рейсах аэропорта: номер рейса, маршрут, марка самолета, общие затраты на рейс, количество пассажиров. Подсчитать стоимость перевозки одного пассажира на рейсе и среднюю стоимость перевозки одного пассажира по аэропорту. Выдать списки номеров рейсов, для которых стоимость перевозки одного пассажира ниже и выше средней по аэропорту.
27. Запись содержит сведения о расходе топлива на автопредприятиях города: название предприятия, количество израсходованного топлива и количество автомашин на предприятии. Подсчитать средний расход топлива на одну машину на каждом предприятии и в

целом по городу. Выдать списки предприятий, у которых расход топлива ниже и выше среднего по городу.

28. Запись содержит данные о книгах: автор, название, издательство, год издания, цена. Получить список книг, изданных в указанный год, и список книг заданного автора.
29. Запись содержит сведения о погоде в городах России: город, дневная и ночная температуры. Для каждого города определить разность температур и выдать списки городов, в которых дневная температура выше нуля, а ночная – ниже и в которых разность температур превышает 5°C .
30. Запись содержит сведения о странах мира: название страны, столица, территория (тыс. кв. км), население (тыс. чел.). Для каждой страны определить плотность населения и выдать списки стран с наименьшей (меньше минимума $+20$ тыс. чел.) и наибольшей (больше максимума -50 тыс. чел.) численностью населения.

Пример программы на языке Turbo Pascal

Создать файл, содержащий 50 случайных целых чисел в диапазоне $[10..30]$, вывести числа на экран в обратном порядке.

Схема алгоритма работы программы представлена на рисунке 14.

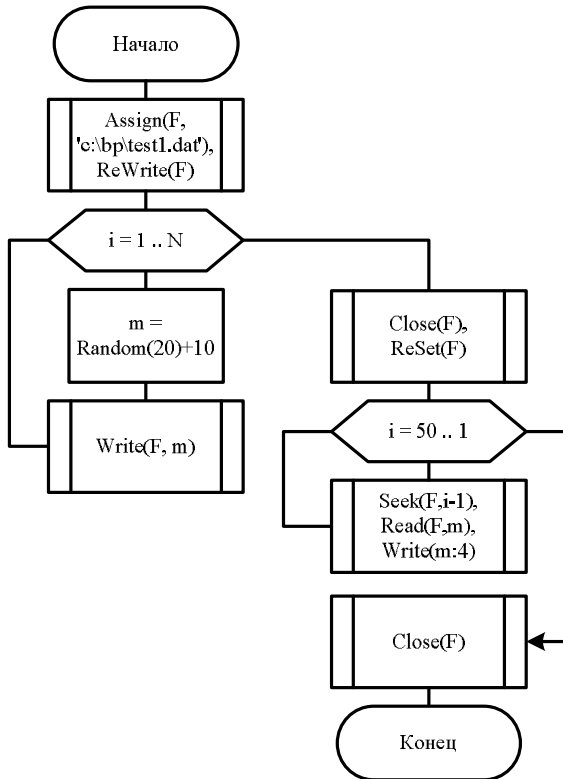


Рис. 14 – Схема алгоритма работы программы

```

Program Primer7;
const
  N=50;
var
  F: file of integer; {Объявление файла с целыми числами}
  i, m : integer;
begin
  Assign(F, 'c:\bp\test1.dat');
    {Связь файловой переменной с файлом}
  ReWrite(F); {Открыть файл для записи}
  for i:=1 to N do
    begin
      m := Random(20)+10; {Получение случайных чисел}
      Write(F, m); {Запись числа в файл}
    end
  end
end
  
```

```

    end;
    Close(F); {Заккрытие файла}
    ReSet(F); {Открыть тот же файл для чтения}
    for i:=50 downto 1 do
        begin
            Seek(F,i-1); {Установка указателя файла}
            Read(F,m); {Считывание числа из файла}
            Write(m:4); {Вывод числа на экран}
        end;
        Close(F); {Заккрытие файла}
        WriteLn;
    end.

```

Контрольные вопросы

1. Что такое «файл» и «файловая переменная»?
2. Что такое «типизированный файл»?
3. Что такое «текстовый файл»?
4. Что такое «нетипизированный файл»?
5. Что такое «указатель файла»?
6. Как установить указатель файла в требуемую позицию?
7. Для каких типов файлов можно устанавливать позицию указателя файла?
8. Какие процедуры предназначены для открытия типизированного файла?
9. Какие процедуры предназначены для открытия текстового файла?
10. Какие процедуры и в какой последовательности надо вызывать для чтения данных из типизированного файла?
11. Какими процедурами выполняется чтение и запись для нетипизированных файлов?
12. Какими процедурами выполняется чтение и запись для текстовых файлов?
13. Как задать размер блока для нетипизированного файла?
14. Какие функции позволяют узнать текущее положение указателя в файле?
15. Как определить возникновение какой-либо ошибки при работе с файлами?

Рекомендованная литература

1. Рапаков, Г. Г. Программирование на языке Pascal / Г. Г. Рапаков, С. Ю. Ржеуцкая. – СПб. : БХВ-Петербург, 2005. – 470 с.
2. Потопахин, В. В. Turbo Pascal. Освой на примерах / В. В. Потопахин. – СПб. : BHV, 2005. – 240 с.
3. Малыхина, М. П. Программирование на языке высокого уровня Turbo Pascal / М. П. Малыхина. – СПб. : БХВ-Петербург, 2006. – 523 с.
4. Потопахин, В. В. Turbo Pascal. Решение сложных задач / В. В. Потопахин. – СПб. : BHV, 2006. – 208 с.
5. Марченко, А. И. Программирование в среде Turbo Pascal 7.0 / А. И. Марченко, Л. А. Марченко. – М. : Век, 2007. – 464 с.
6. Фаронов, В. В. Turbo Pascal 7.0. Начальный курс. / В. В. Фаронов. – М. : КноРус, 2007. – 576 с.
7. Немнюгин, С. А. Turbo Pascal: практикум / С. А. Немнюгин. – СПб. : Питер, 2007. – 267 с.
8. Фаронов, В. В. Turbo Pascal / В. В. Фаронов. – СПб. : БХВ-Петербург, 2007. – 1037 с.
9. Вольский, С. В. Turbo Pascal 7.0 для студентов и школьников / С. В. Вольский, П. А. Дмитриев. – М. : Наука и техника, 2007. – 214 с.
10. Культин, Н. Б. Turbo Pascal в задачах и примерах / Н. Б. Культин. – СПб. : БХВ-Петербург, 2008. – 256 с.
11. Фаронов, В. В. Turbo Pascal 7.0. Практика программирования / В. В. Фаронов. – М. : КноРус, 2008. – 576 с.
12. Немнюгин, С. А. Изучаем Turbo Pascal / С. А. Немнюгин, Л. В. Перколаб. – СПб. : Питер, 2008. – 312 с.
13. Шпак, Ю. А. Turbo Pascal. Просто как дважды два / Ю. А. Шпак. – М. : ЭКСМО, 2008. – 396 с.
14. Меженный, О. А. Turbo Pascal: самоучитель / О. А. Меженный. – М. : Диалектика, 2008. – 335 с.
15. Немнюгин, С. А. Программирование на языке высокого уровня / С. А. Немнюгин. – СПб. : Питер, 2008. – 543 с.
16. Культин, Н. Б. Программирование в Turbo Pascal 7.0 и Delphi / Н. Б. Культин. – СПб. : БХВ-Петербург, 2008. – 380 с.
17. Ковтанюк, Ю. С. Программирование на Turbo Pascal: шаг за шагом / Ю. С. Ковтанюк. – М. : ЭКСМО, 2008. – 587 с.