

Министерство образования и науки РФ
Севастопольский государственный университет
Кафедра информатики и управления в технических системах

**МАССИВЫ И ФУНКЦИИ
В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PASCAL**

Методические указания
к выполнению лабораторных работ
по дисциплине
" Алгоритмизация и программирование"
для студентов очной формы обучения
по направлениям подготовки
27.03.04 "Управление в технических системах" и
09.03.01 " Информатика и вычислительная техника"

Севастополь
2016

УДК 681.5

Массивы и функции в языке программирования Pascal: Методические указания к выполнению лабораторных работ, входящих в блок № 2 по дисциплине «Алгоритмизация и программирование»/ Сост. Д.Н. Старинская, А.А. Кабанов, В.В. Захаров. – Севастополь: Изд-во СевГУ, 2016. – 20 с.

Целью методических указаний является оказание помощи студентам при выполнении лабораторных работ, целью которых является приобретение навыков составления простейших алгоритмов, изучение основ языка Pascal, освоение базовых приемов работы в среде Turbo Pascal и её модификаций Free Pascal, ABC PASCAL и т.п.

Методические указания предназначены для студентов дневной формы обучения по направлениям подготовки 27.03.04 "Управление в технических системах" и 09.03.01 "Информатика и вычислительная техника"

Методические указания рассмотрены и утверждены на заседании кафедры информатики и управления в технических системах (протокол № 7 от 29.08.2016 г.)

Допущено учебно-методическим центром СГУ в качестве методических указаний.

Рецензент

Кабанов А.А., канд. техн. наук, зав. доцент

СОДЕРЖАНИЕ

Введение	4
1. Лабораторная работа № 4. Программирование методов обработки одномерных массивов на языке Pascal	4
1.1. Цель работы	4
1.2. Задание на работу	4
1.3. Краткие теоретические сведения	6
1.3.1. Массивы	6
1.3.2. Процедуры и функции	7
1.3.3. Пример программы	8
1.4. Содержание отчета и порядок защиты работы	8
1.5. Контрольные вопросы	9
2. Лабораторная работа № 5. Программирование алгоритмов, реализующих операции над матрицами на языке Pascal	10
2.1. Цель работы	10
2.2. Задание на работу	10
2.3. Краткие теоретические сведения	12
2.3.1. Примеры задач обработки матриц	12
2.3.2. Работа с файлами	13
2.3.3. Пример программы	14
2.4. Содержание отчета и порядок защиты работы	15
2.5. Контрольные вопросы	16
Библиографический список	16
Приложение А. Перечень тем блока № 2	17
Приложение Б. Образец задания к контрольной работе № 2	18
Приложение В. Содержимое файла matrix.txt	20

ВВЕДЕНИЕ

Лабораторные работы данного цикла входят в блок № 2 дисциплины «Алгоритмизация и программирование». Для успешной сдачи блок № 2 необходимо выполнить и защитить лабораторные работы №№ 3-4 *не позднее 16-ой недели семестра*, а также написать контрольную работу № 2 на оценку не менее "удовлетворительно"

Примерный перечень тем, входящих в блок №2, приведен в приложении А. Образец задания контрольной работы представлен в приложении Б.

1. ЛАБОРАТОРНАЯ РАБОТА №4. ПРОГРАММИРОВАНИЕ МЕТОДОВ ОБРАБОТКИ ОДНОМЕРНЫХ МАССИВОВ НА ЯЗЫКЕ PASCAL

1.1. Цель работы

Исследование циклических алгоритмов и программ, осуществляющих типичные операции над одномерными массивами. Приобретение навыков программирования ввода и вывода массивов.

1.2. Задание на работу

Работа выполняется в среде Turbo Pascal 7.0. Описание лабораторного стенда приведено в методических указаниях к лабораторной работе № 1.

Создайте программу, реализующую обработку массива вещественных чисел по заданию, указанному в таблице 1.1, в соответствии с номером варианта. Программа должна удовлетворять следующим требованиям:

- В основной программе нужно объявить массив вещественных чисел и заполнить его значениями с клавиатуры.
- Основная программа должна вызывать две вспомогательные подпрограммы.
- Первая подпрограмма-функция должна быть предназначена вычисления значения в соответствии с заданием, указанным в столбце 3 таблицы вариантов. Основная программа должна осуществлять вывод на экран результата работы первой функции;
- Вторая подпрограмма-процедура должна быть предназначена для вывода на экран списка элементов массива, удовлетворяющих условию, указанному в столбце 4 таблицы вариантов.
- Значения A , B , m (в зависимости от варианта) должны быть параметрами подпрограмм и должны вводиться с клавиатуры в основной программе.

Таблица 1.1 – Варианты заданий

№ вар.	Задания и значения исходных данных		
	Набор значений элементов массива и дополнительные исходные данные	Функция должна вычислять	Процедура должна вывести на экран элементы, удовлетворяющие условию
1	$X = 15,6; -0,7; 3,5 \cdot 10^4; 0,9 \cdot 10^{-5}; -5; 1,7; 0$ $A = 0; B = 6,1.$	Сумму положительных элементов	$A < X_i \leq B.$
2	$X = 0,1; -1,5; -4 \cdot 10^{-8}; 3,2 \cdot 10^3; 0; 11; 0,2; 0$ $A = -1,005; B = 2,5.$	Минимальный элемент массива	$X_i > B$ или $X_i < A$
3	$X = 0; -2; 0; 2,5; -11,4; 0,5; 10; 0; 5 \cdot 10^{-6}; 10^4$ $B = 11,4.$	Максимальный квадрат значения элемента массива	$ X_i < B$
4	$X = 0; 0; -1; 10; 13,6; 10^5; 9,6 \cdot 10^{-5}; -0,75$ $A = 15.$	Количество элементов, больших A	$-0,8 < X_i < 0,8$
5	$X = -3,8; 12; 10^3; -4 \cdot 10^{-3}; 4 \cdot 10^{-3}; 0; -1$ $m = 4, A = 0,2, B = 12.$	Сумму первых m элементов	$A \leq X_i^2 < B$
6	$X = 1; -1; 0,2; -2,7 \cdot 10^4; 9 \cdot 10^{-3}; 0,035.$ $A = 0, B = 1.$	Произведение элементов массива	$X_i \geq B$ или $X_i \leq A$
7	$X = -109; 0,2 \cdot 10^{-3}; -1,5 \cdot 10^3; 0; -1; 0.$ $A = 10.$	Сумму модулей элементов	$X_i^2 \geq A$
8	$X = 10^4; -0,1 \cdot 10^{-4}; 0,6 \cdot 10^{-4}; -1; -10,1$ $A = 3 \cdot 10^{-4}.$	Номер максимального элемента массива	$ X_i > A$
9	$X = 1; 15,2; -15; 10^9; -0,8; 0,17; -9 \cdot 10^{-4}; 1$ $B = 1.$	Сумму отрицательных элементов	$\sqrt{ X_i } < B$
10	$X = -1; 2; -1; 0,2; -0,3; 11; 10^5; 8 \cdot 10^{-5}; 0,6$ $A = 1.$	Номер минимального элемента массива	$ X_i < A$
11	$X = 0; -1; 0; 1; 0; 3,2 \cdot 10^6; 9 \cdot 10^{-5}; -0,3; 0; -1$ $m = 3$	Сумму последних m элементов	$X_i \neq 0$
12	$X = -12; 0; 0; -10^{-4}; 3,95 \cdot 10^3; 4 \cdot 10^5.$ $B = 12.$	Сумму элементов массива, больших B	$X_i^3 > B$

Таблица 1.1 – Варианты заданий

№ вар.	Задания и значения исходных данных		
	Набор значений элементов массива и дополнительные исходные данные	Функция должна вычислять	Процедура должна вывести на экран элементы, удовлетворяющие условию
13	$X = 5,6; -0,8; 1,5 \cdot 10^3; -0,9 \cdot 10^{-5}; -8; 1,7; 0$ $B = 5,1.$	Максимальный по модулю элемент массива	$X_i \leq B^2$
14	$X = -4,6; -2; -1; 0,3; -0,3; 11; 1,8 \cdot 10^3; 1$ $A = 1.$	Сумму элементов с четными номерами	$X_i > A^2$

1.3. Краткие теоретические сведения

1.3.1. Массивы

Массивы служат для объединения под одним именем нескольких переменных, имеющих одинаковую природу. Массив – такая структура данных, которая характеризуется:

- фиксированным набором элементов одного и того же типа;
- каждый элемент имеет уникальный набор значений индексов;
- количество индексов определяют размерность массива, например, два индекса – двумерный массив, три индекса – трехмерный массив, один индекс – одномерный массив или вектор;
- обращение к элементу массива выполняется по имени массива и значениям индексов для данного элемента.

Синтаксис описания типа массива:

```
type имя_типа = array[тип_индекса] of тип_элемента;
```

Количество элементов массива определяет тип индекса, а множество значений элементов определяется типом элемента.

Пример:

```
const n=5;
type tMass = array[1..n] of integer;
var a, b: tMass;
```

Для доступа к отдельным элементам массива используется индекс массива. Элемент массива записывается как имя массива, за которым в квадратных скобках указывается индекс нужного элемента.

```
i := 4; a[i] := -3; a[i-2] := 7;
for i:=1 to n do
  read(a[i]);
```

1.3.2. Процедуры и функции

Подпрограмма – процедура или функция – это часть программы, имеющая имя и решающая определенную задачу. Она описывается один раз, а вызываться может столько раз, сколько необходимо.

В Pascal имеется два вида подпрограмм: процедуры и функции. Они имеют незначительные отличия в синтаксисе и правилах вызова.

Синтаксис описания процедуры совпадает с синтаксисом записи программы (за исключением служебного слова `procedure` и правил записи списка формальных параметров):

```
procedure имя [(список формальных параметров)] ; {заголовок}
  разделы описаний
begin
  раздел операторов;
end;
```

Квадратные скобки в данном случае не являются элементом синтаксиса, а означают, что список параметров может отсутствовать. Если в описании процедуры отсутствует список параметров, то такую процедуру называют процедурой без параметров.

Синтаксис вызова процедуры:

```
Имя процедуры [(список фактических параметров)] ;
```

Вызов процедуры можно указывать в любом месте программы, где допустим оператор языка Pascal.

Функцией называется подпрограмма, которая вычисляет единственное значение, относящееся к порядковому, вещественному типам или типу указателей и используемое при подсчете выражений.

Объявление функции:

```
function имя [(список форм. параметров)] : тип ;
  разделы описаний
begin
  раздел операторов;
  Имя := выражение;
end;
```

Список формальных параметров записывается по общим правилам, как и для процедур.

Через двоеточие после списка указывается тип возвращаемого функцией результата. Тип результата должен быть либо простым, либо указателем.

В разделе операторов функции необходимо хотя бы один раз использовать имя функции в левой части оператора присваивания. Т.е. в разделе операторов функции обязательно должно стоять присваивание имени функции какого-либо «возвращаемого» значения.

Для обращения к функции используется вызов функции:

```
Имя функции [(список фактических параметров)]
```

Механизм и правила взаимодействия формальных и фактических параметров такие же, как и в процедурах.

1.3.3. Пример программы

Исходные данные к программе:

$X = -1,3; 5,0; 4; 0; -1,5 \cdot 10^{-4}; 0; 5,3; -9,1; 1,2 \cdot 10^3; 4,2; 0.$

Требуется вычислить максимальный элемент массива.

```
program massiv;
const
    n = 10;
type
    tMass = array [1..n] of real;
var
    m : real;
    i : integer;
    x : tMass;
{*****}
function maxval(var mass : tMass; size : Integer): real;
var
    i : Integer;
    max : Real;
begin
    max := mass[1];
    for i:=2 to size do
        if mass[i]>max then
            max := mass[i];
    maxval := max;
end;
{*****}
begin
    for i := 1 to n do
        begin
            write('x[' , i:2, ']=');
            readln(x[i]);
        end;
    m := maxval(x,n);
    writeln('Max=',m);
    readln;
end.
```

1.4. Содержание отчета и порядок защиты работы

Выполнение и защита лабораторной работы производится каждым студентом индивидуально. Защита результатов лабораторной работы осуществляется при наличии работающей программы и полностью оформленного отчета.

Отчет должен включать в себя титульный лист, цель работы, постановку задачи, схемы основной программы и обеих подпрограмм, текст программы, результаты работы программы, выводы.

Защита работы состоит в следующем:

- представление работающей программы на компьютере;

- предъявление отчета, оформленного в соответствии с приведенными требованиями;
- ответы на вопросы преподавателя касательно теоретической и практической части работы. Примеры возможных вопросов приведены в подразделе 1.5.

1.5. Контрольные вопросы

- 1) Изложите классификацию типов языка Pascal.
- 2) Какие типы данных относятся к простым, а какие – к структурированным?
- 3) Изложите основные особенности порядковых типов данных.
- 4) Охарактеризуйте перечисляемый тип данных.
- 5) Охарактеризуйте интервальный тип данных.
- 6) Что представляет собой массив? Каковы основные характеристики массива?
- 7) Изобразите синтаксическую диаграмму объявления типа-массива.
- 8) Перечислите операции, допустимые для переменных-массивов.
- 9) Как осуществляется доступ к элементам массива?
- 10) Что такое индекс массива? Какие допустимы типы индексов массива?
- 11) Изложите алгоритм поиска максимального элемента в массиве.
- 12) Что собою представляет процедура? Когда целесообразно использовать процедуры?
- 13) Что собою представляет функция? Когда целесообразно использовать функции?
- 14) Какие переменные называются глобальными, какие – локальными?
- 15) Что такое формальные параметры? Что такое фактические параметры? Какие соответствия должны соблюдаться между формальными и фактическими параметрами?
- 16) Что такое параметры-значения и параметры-переменные? Чем они отличаются друг от друга?

2. ЛАБОРАТОРНАЯ РАБОТА №5. ПРОГРАММИРОВАНИЕ АЛГОРИТМОВ, РЕАЛИЗУЮЩИХ ОПЕРАЦИИ НАД МАТРИЦАМИ НА ЯЗЫКЕ PASCAL

2.1. Цель работы

Цель работы заключается в исследовании методов и алгоритмов, реализующих различные матричные операции и манипулирование внешними файлами, а также в приобретении навыков их программирования на языке Pascal.

2.2. Задание на работу

Работа выполняется в среде Turbo Pascal 7.0. Описание лабораторного стенда приведено в методических указаниях к лабораторной работе № 1.

Программа должна удовлетворять следующим требованиям:

- в лабораторной работе необходимо реализовать процедуру и функцию, реализующие вычисления, в соответствии с номером варианта, указанным в таблице 2.1;

- в основной программе необходимо ввести значения элементов заданной матрицы с клавиатуры, а другой матрицы – из текстового файла `matrix.txt`. Содержимое файла `matrix.txt` представлено в приложении В.

- в основной программе вызвать разработанные процедуры и функции для заданных матриц. Вычислить $A*B$ и $B*A$ или $A+B$ и $B+A$, A^T и B^T в зависимости от варианта.

- программа должна осуществлять вывод полученных результатов в отдельный файл.

Таблица 2.1 – Варианты заданий

№ варианта	Задания и значения исходных данных			
	Процедура	Функция	Матрицы	Ввести с клавиатуры
1	Умножение	Максимальная сумма модулей элементов столбца	A, B	A
2	Сложение	Минимальный среди элементов выше главной диагонали	C, D	C
3	Транспонирование	Минимальный среди элементов, расположенных в четных столбцах	E, F	E
4	Умножение	максимальный среди элементов ниже главной диагонали	E, F	E

Таблица 2.1 – Варианты заданий

№ варианта	Задания и значения исходных данных			
	Процедура	Функция	Матрицы	Ввести с клавиатуры
5	Сложение	Сумма модулей элементов выше главной диагонали	C, D	D
6	Транспонирование	Последний положительный среди элементов ниже главной диагонали	A,B	A
7	Умножение	Корень квадратный из суммы квадратов элементов	C, D	D
8	Сложение	Максимальный среди элементов, расположенных в нечетных столбцах	A, B	A
9	Транспонирование	Минимальный элемент среди положительных	E, F	F
10	Умножение	Первый нулевой среди элементов ниже главной диагонали	C, D	C
11	Сложение	Максимальная сумма модулей элементов строки	E,F	E
12	Транспонирование	Сумма квадратов элементов ниже главной диагонали	C,D	C
13	Умножение	Первый положительный среди элементов ниже главной диагонали	E,F	F
14	Сложение	Максимальный элемент среди отрицательных	C, D	D

2.3. Краткие теоретические сведения

2.3.1. Примеры задач обработки матриц

Пример 1. Ввод матрицы с клавиатуры

```
... ..
write('Введите количество строк (не более 10), n=');
read(n);
write('Введите количество столбцов (не более 15), m=');
read(m);
writeln('Введите элементы матрицы');
for i:=1 to n do
  for j:=1 to m do
    read (A[i,j]); { ввод элементов по строкам }
```

Пример 2. Вывод матрицы на экран

```
... ..
writeln('Вывод матрицы');
for i:=1 to n do
  begin
    for j:=1 to m do
      write (A[i,j]:12:3);
    writeln;
```

... ..

Большое количество задач связано с обработкой части элементов **квадратной** матрицы (количество строк совпадает с количеством столбцов), например, рассматриваются только элементы главной диагонали, над дополнительной диагональю (соединяющей элементы $A_{n,1}$ и $A_{1,n}$ и т.д. При решении таких задач возникает проблема, определения границ изменения индексов обрабатываемых элементов. Ниже приведем ряд типичных случаев с указанием соответствующих начальных и конечных значений в заголовках циклов.

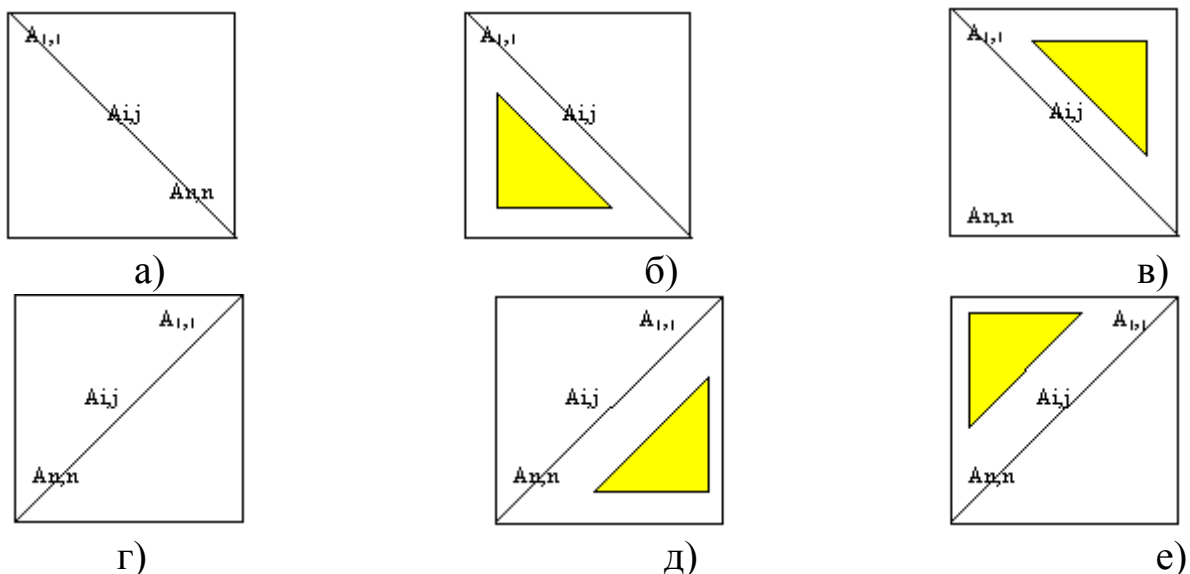


Рисунок 2.1 – Определение индексов элементов квадратной матрицы
а) элементы, принадлежащие главной диагонали

```
for i:= 1 to N do
```

```
  A[i,i]
```

б) элементы под главной диагональю

```
for i:= 2 to N do
```

```
for j:= 1 to i-1 do
```

```
  A[i,j]
```

в) элементы над главной диагональю

```
for i := 1 to N-1 do
```

```
for j := i+1 to N do
```

```
  A[i, j]
```

г) элементы диагонали, соединяющей элементы $A_{n,1}$ и $A_{1,n}$

```
for i := 1 to N do
```

```
  A[i,N+1-i]
```

д) элементы под диагональю, соединяющей элементы $A_{n,1}$ и $A_{1,n}$

```
for i := 2 to N do
```

```
for j:=N-i+2 to N do
```

```
  A[i, j]
```

е) элементы над диагональю, соединяющей элементы $A_{n,1}$ и $A_{1,n}$

```
for i := 1 to N-1 do
```

```
for j := 1 to N-1 do
```

```
  A[i, j]
```

2.3.2. Работа с файлами

Исходные значения элементов векторов и матриц удобно хранить на внешних носителях в виде файлов. Часто исходные числовые данные хранятся в текстовом файле. Текстовыми называются файлы, состоящие из последовательности символов, разбитой на строки произвольной длины. Такие файлы имеют тип *Text*. Каждая строка заканчивается признаком конца строки – специальным символом.

Переменные текстового типа описываются в разделе переменных, например:

```
var f: Text;
```

Для чтения данных из текстового файла используют следующие стандартные процедуры и функции:

Assign (<файловая переменная>, <внешнее имя файла>) - связывает файловую переменную с дисковым файлом;

Reset (<файловая переменная>) - открывает файл для чтения;

Read (<файловая переменная, список переменных>) - процедура присваивает значения из текстового файла переменным списка ввода;

ReadLn (<файловая переменная>) - процедура пропускает все символы в текстовом файле до начала следующей строки;

Eoln (<файловая переменная>) - стандартная функция, возвращающая значение *True*, если из файла считан символ, за которым следует признак конца строки и значение *False* в противном случае;

Eof(<файловая переменная>) - стандартная функция, возвращающая значение *True*, если из файла считан символ за которым следует признак конца файла и значение *False* в противном случае;

Close(<файловая переменная>) - закрывает файл.

В начале работы с текстовым файлом его необходимо открыть с помощью процедуры *Reset*, затем, используя процедуры *Read*, *ReadLn*, *Eoln*, *Eof*, считываются данные, и процедурой *Close* закрывается файл.

Для вывода данных в файл используют процедуры:

Rewrite (<файловая переменная>)- открывает файл для записи;

Write (<файловая переменная, список выражений>) - процедура записывает значения выражений из списка в текстовый файл;

WriteLn (<файловая переменная>) - процедура записывает признак конца строки в текстовый файл.

2.3.3. Пример программы

Ниже приведен пример программы, осуществляющей следующие действия:

- чтение матрицы A из файла matrix.txt,
- вызов процедуры AddConst, прибавляющей заданное значение к каждому элементу квадратной матрицы;
- вывод новой матрицы в файл result.txt.

```

Program lr6;
const N = 4;
type tMatr = array [1..10, 1..10] of real;
var
    f      :   Text;
    i, j   :   Integer;
    Num    :   Real;
    A      :   tMatr;
procedure AddConst( var Matr : tMatr; Size : Integer;
                    C : Real);
begin
    for i := 1 to Size do
        for j := 1 to Size do
            A[i,j] := A[i,j] +C;
end;
begin
{ открытие файла для чтения }
    Assign(f, 'matrix.txt');
    Reset(f);
    {пропуск строки "A=", предшествующей матрице A}
    ReadLn (f);
    { ввод значений элементов матрицы A }
```

```

for i:=1 to N do
for j:=1 to N do
    Read(f, A[i, j]);
{ Заккрытие файла matrix.txt}
Close(f);
write('Прибавить к элементам матрицы число Num=');
readln(Num);
AddConst(A, N, Num);
Assign(f, 'Result.txt');
Rewrite(f); { открытие файла для вывода }
    { вывод матрицы A в файл}
WriteLn(f, 'Матрица A':30);
for i:=1 to N do
    begin
        for j:=1 to N do
            Write(f, A[i, j]:10:3);
        WriteLn(f);
    end;
close(f);
end .

```

2.4. Содержание отчета и порядок защиты работы

Выполнение и защита лабораторной работы производится каждым студентом индивидуально. Защита результатов лабораторной работы осуществляется при наличии работающей программы и полностью оформленного отчета.

Отчет должен включать в себя следующие разделы

- титульный лист;
- цель работы,
- постановка задачи. Этот раздел должен содержать вариант задания в соответствии с таблицей 2.1;
- схема программы. Этот раздел должен содержать схему программы, а также схемы разработанных процедуры и функции.
- текст программы на языке Pascal;
- результаты работы программы. Этот раздел должен содержать результаты работы программы при значениях матриц, взятых в соответствии с файлом matrix.txt.
- выводы.

Защита работы состоит в следующем:

- представление работающей программы на компьютере;
- предъявление отчета, оформленного в соответствии с указанными требованиями;
- ответы на вопросы преподавателя по теоретической и практической части работы. Примеры возможных вопросов приведены в подразделе 2.5.

2.5. Контрольные вопросы

- 1) Что собою представляет двумерный массив?
- 2) Приведите пример фрагмента программы, который бы осуществлял вывод элементов главной диагонали квадратной матрицы на экран.
- 3) Дайте определение операции транспонирования матрицы. Приведите пример фрагмента программы, в котором бы осуществлялось транспонирование квадратной матрицы.
- 4) Дайте определение операции умножения матриц и приведите соответствующую формулу. Как должны быть согласованы размерности матриц? Приведите пример фрагмента программы, в котором бы осуществлялось умножение двух матриц.
- 5) Что собою представляет тип данных – файл? Сопоставьте характеристики типов файл и массив.
- 6) Что собою представляет буферная переменная, связанная с файлом?
- 7) Что такое текстовый файл?
- 8) Для чего предназначены и как работают процедуры Assign и Close?
- 9) Для чего предназначены и как работают процедуры Reset и Rewrite?
- 10) Для чего предназначены и как работают процедуры Read и Readln?
- 11) Для чего предназначены и как работают процедуры Write и Writeln?
- 12) Для чего предназначены и как работают функции Eof и Eoln?
- 13) Что собою представляет процедура? Когда целесообразно использовать процедуры?
- 14) Что собою представляет функция? Когда целесообразно использовать функции?
- 15) Какие переменные называются глобальными, какие – локальными?
- 16) Что такое формальные параметры? Что такое фактические параметры? Какие соответствия должны соблюдаться между формальными и фактическими параметрами?
- 17) Что такое параметры-значения и параметры-переменные? Чем они отличаются друг от друга?
- 18) Каковы особенности передачи параметров-массивов?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Пильщиков В.Н. Сборник упражнений по языку Pascal: Учебное пособие для вузов / В.Н. Пильщиков. – М.: Наука, 1989г. – 160с.
2. Павловская Т. А. Паскаль: программирование на языке высокого уровня: практикум [Текст] : учеб. пособие для студ. вузов / Т. А. Павловская. – М. и др.: Питер, 2006. – 408 с.
3. Павловская Т. А. Паскаль: программирование на языке высокого уровня [Текст] : учеб. для студ. вузов / Т. А. Павловская. – М. и др. : Питер, 2006. – 400 с.

ПРИЛОЖЕНИЕ А. ПЕРЕЧЕНЬ ТЕМ МОДУЛЯ №2

(справочное)

1. Краткая характеристика концепции структурного программирования. Структурное кодирование. Проектирование сверху-вниз. Модульное программирование
2. Модульное программирование. Правила использования подпрограмм.
3. Процедуры в языке Pascal. Объявление процедуры. Оператор процедуры.
4. Локальные и глобальные переменные в языке Pascal. Точка определения, область видимости.
5. Процедуры с параметрами в языке Pascal. Список формальных параметров. Список фактических параметров.
6. Процедуры с параметрами в языке Pascal. Механизмы передачи параметров (параметры-значения, параметры-переменные).
7. Функции в языке Pascal. Примеры.
8. Некоторые правила хорошего стиля программирования.
9. Классификация типов данных в языке Pascal.
10. Общая характеристика порядковых типов в языке Pascal. Символьный тип данных в языке Pascal.
11. Общая характеристика порядковых типов в языке Pascal. Данные перечисляемого типа в языке Pascal.
12. Общая характеристика порядковых типов в языке Pascal. Данные интервального типа в языке Pascal.
13. Тип данных массив в языке Pascal. Тип индекса и тип элементов. Объявление переменных массивового типа. Многомерные массивы.
14. Примеры алгоритмов обработки числовых массивов на языке Pascal. Поиск максимального элемента в массиве. Заполнение матрицы значениями.
15. Примеры алгоритмов обработки числовых массивов на языке Pascal. Транспонирование и умножение матриц.
16. Особенности передачи параметров - массивов в языке Pascal. Пример.
17. Строковый тип данных String в Turbo Pascal.
18. Файловый тип данных в языке Pascal. Стандартные процедуры и функции языка Pascal для работы с типизированными файлами.
19. Реализация операций записи в файл в языке Pascal.
20. Реализация операций чтения из файла в языке Pascal.
21. Текстовые файлы в языке Pascal. Процедуры и функции для работы с текстовыми файлами. Стандартные файлы Input и Output.
22. Текстовые файлы в языке Pascal. Особенности чтения данных из текстовых файлов. Примеры.
23. Текстовые файлы в языке Pascal. Форматированный вывод. Примеры.

ПРИЛОЖЕНИЕ Б.
ОБРАЗЕЦ ЗАДАНИЯ К КОНТРОЛЬНОЙ РАБОТЕ №2
(справочное)

Задание №1

(5 баллов)

Формальный параметр подпрограммы описан как `var a: integer`. **Соответствующий ему фактический параметр может быть:**

- 1) Выражением целого типа;
- 2) Переменной целого типа;
- 3) Выражением любого типа;
- 4) Переменной вещественного типа;
- 5) Константой целого типа с именем a.

Номер правильного ответа: _____

Задание №2

(5 баллов)

Какое из нижеследующих утверждений неверно:

- 1) В разделе операторов функции необходимо хотя бы один раз использовать имя функции в левой части оператора присваивания.
- 2) Глобальные переменные нужно передавать в подпрограмму через список параметров;
- 3) К данным порядкового типа можно применять операции отношения;
- 4) Множеством значений интервального типа является замкнутый интервал значений некоторого порядкового типа.
- 5) Если формальным параметром является параметр-переменная, то соответствующий фактический параметр не может быть выражением.

Номер правильного ответа: _____

Задание №3

(8 баллов)

Что будет выведено в результате работы программы?

```
Program Task3;
Var a,b,c,d : Integer;
Procedure Proc(a: integer; var c : integer; var d : integer);
var b : integer;
begin
    a := 5; c := 7; b := 9;
end;
Begin
    a := 1; b :=1; c :=1;
    Proc(b, a, c);
    Write(a, ' | ', b, ' | ', c);
End.
```

Варианты ответов

- 1) 1|1|1
- 2) 5|7|9
- 3) 9|7|5
- 4) 7|1|1
- 5) 1|7|9

Номер правильного ответа: _____

Задание №4

(Максимальная оценка 12 баллов)

Рассматривается массив целых чисел t размером 3 на 4.

- а) Напишите объявление для переменной t .
- б) Сколько элементов в массиве t ?
- в) Напишите имена всех элементов второй строки массива t .
- г) Напишите имена всех элементов третьего столбца массива t .

Задание №5

(Максимальная оценка 20 баллов)

Дан массив из 15 целых чисел. Вычислить сумму модулей элементов массива. Заменить первый положительный элемент на второй элемент массива.

ПРИЛОЖЕНИЕ В.
СОДЕРЖИМОЕ ФАЙЛА MATRIX.TXT
(справочное)

A =

1.0668	0.2944	-0.6918	-1.4410
0	-1.3362	0.8580	0.5711
-0.0956	0.7143	1.2540	0
-0.8323	1.6236	-1.5937	0.6900

B =

0.8156	1.1908	-1.6041	-0.8051
0.7119	-1.2025	0.2573	0
1.2902	0	-1.0565	0.2193
0.6686	-0.1567	1.4151	-0.9219

C =

-2.1707	0.5077	0.3803	0.0000
-0.0592	1.6924	-1.0091	0
-1.0106	0.5913	-0.0195	1.0950
0.6145	-0.6436	-0.0482	-1.8740

D =

0.4282	0.0403	-0.3775	0.1184
0.8956	0.6771	-0.2959	0.3148
0.7310	0.5689	-1.4751	1.4435
0.5779	-0.2556	-0.2340	-0.3510

E =

0.6232	0.2120	1.0823	-0.6355
0.7990	0.2379	-0.1315	-0.5596
0.9409	-1.0078	0.3899	0.4437
-0.9921	-0.7420	0.0880	-0.9499

F =

0.9501	0	-0.8214	-0.9218
-0.2311	-0.7621	0.4447	0.7382
0.6068	-0.4565	0.6154	0.1763
0.4860	0.0185	0.7919	-0.4057