

Министерство образования и науки Украины
Севастопольский национальный технический университет



ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ PASCAL

Методические указания
к выполнению лабораторных и контрольной работ
по дисциплине
«Алгоритмические языки и программирование»
в первом семестре
для студентов направления подготовки
050201 «Системная инженерия»
заочной формы обучения

Севастополь
2008



Основы языка программирования Pascal : Методические указания к изучению дисциплины «Алгоритмические языки и программирование» в первом семестре для студентов направления подготовки 050201 «Системная инженерия» заочной формы обучения /Разраб. Д.Н. Старинская. – Севастополь: Изд-во СевНТУ, 2008. – 40 с.

Целью методических указаний является оказание помощи студентам заочной формы обучения в изучении дисциплины «Алгоритмические языки и программирование» в первом семестре. Методические указания содержат описание и индивидуальные задания контрольной и трех лабораторных работ. Лабораторные работы предназначены для приобретения навыков составления алгоритмов, изучения основ языка программирования Pascal, освоения базовых приемов работы в среде Turbo Pascal.

Методические указания предназначены для студентов по направлению подготовки 050201 «Системная инженерия» заочной формы обучения.

Методические указания рассмотрены и утверждены на заседании кафедры технической кибернетики, протокол № 5 от 26 марта 2008г.

Допущено учебно-методическим центром СевНТУ в качестве методических указаний.

Рецензент:

Е.А. Шушляпин, д-р техн. наук, профессор.

СОДЕРЖАНИЕ

Введение	5
1. Лабораторная работа №1 Программирование линейных алгоритмов на языке Pascal	6
1.1. Цель работы	6
1.2. Задание на работу	6
1.3. Краткие теоретические сведения	7
1.3.1. Основные приемы работы в среде Turbo Pascal	7
1.3.2. Стандартные арифметические функции	10
1.3.3. Пример программы	10
1.4. Содержание отчета и порядок защиты работы	12
1.5. Контрольные вопросы	12
2. Лабораторная работа №2 Программирование разветвляющихся алгоритмов на языке Pascal	13
2.1. Цель работы	13
2.2. Задание на работу	13
2.3. Пример программы	15
2.4. Содержание отчета и порядок защиты работы	16
2.5. Контрольные вопросы	17
3. Лабораторная работа №3 Программирование циклических алгоритмов на языке Pascal	18
3.1. Цель работы	18
3.2. Задание на работу	18
3.3. Краткие теоретические сведения	20
3.3.1. Массивы	20
3.3.2. Пример программы	20
3.4. Содержание отчета и порядок защиты работы	22
3.5. Контрольные вопросы	23
4. Контрольная работа. Программирование операций над матрицами на языке Pascal	24
4.1. Цель работы	24
4.2. Задание на работу	24
4.3. Краткие теоретические сведения	25
4.3.1. Примеры задач обработки матриц	25
4.3.2. Работа с файлами	27
4.3.3. Процедуры и функции	28
4.3.4. Пример программы	29
4.4. Содержание отчета и порядок защиты работы	30
4.5. Контрольные вопросы	30
Библиографический список	31
Приложение А. Пример оформления титульного листа отчета	32

Приложение Б. Пример оформления основной части отчета	33
Приложение В. Экзаменационные вопросы	35
Приложение Г. Общие требования к оформлению схем программ.....	36
Приложение Д Содержимое файла matrix.txt.....	38

ВВЕДЕНИЕ

Целью преподавания дисциплины «Алгоритмические языки и программирование» является обучение студентов алгоритмизации и программированию задач для их решения на современных ЭВМ.

В результате изучения этой дисциплины в 1-м семестре студент должен усвоить основы языка программирования Pascal.

В 1-м семестре в соответствии с рабочей программой по дисциплине необходимо сдать **экзамен**. Перечень вопросов к экзамену приводится в Приложении В. Для самостоятельной подготовки рекомендуются учебники [1–4].

Для допуска к экзамену студент должен выполнить и защитить отчеты по **трем лабораторным работам**, а также по **контрольной работе**.

При выполнении каждой работы необходимо составить алгоритм в виде схемы программы, записать текст программы на Pascal, осуществить ее отладку и запуск на компьютере. Результаты каждой работы необходимо оформить в виде отдельного **отчета**. Защита работы осуществляется при наличии работающей программы и полностью оформленного отчета.

Каждый отчет должен содержать следующие разделы:

- титульный лист;
- цель работы;
- постановка задачи;
- схема программы;
- текст программы на языке Pascal;
- результаты работы программы;
- выводы.

Отчет оформляется на листах бумаги формата А4 с одной стороны. Пример оформления титульного листа отчета приведен в приложении А. Пример оформления основной части отчета приведен в приложении Б. Общие требования к оформлению схем программ приведены в Приложении Г, для получения более подробной информации см. ГОСТ [5]. Общие требования к оформлению отчетов приведены в ДСТУ [6].

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.

1. ЛАБОРАТОРНАЯ РАБОТА №1

ПРОГРАММИРОВАНИЕ ЛИНЕЙНЫХ АЛГОРИТМОВ НА ЯЗЫКЕ PASCAL

1.1. Цель работы

Целью данной работы является приобретение навыков составления и программирования линейных алгоритмов, реализация вычисления сложных арифметических выражений с помощью операторов языка Pascal.

1.2. Задание на работу

1) Ознакомьтесь с теоретическими сведениями об арифметических функциях языка Pascal, приведенными в пункте 1.3.2 методических указаний.

2) Изучите пример программы, приведенный в разделе 1.3.3. Рекомендуем самостоятельно повторить все стадии разработки программы, приведенные в примере.

3) Создайте программу, реализующую вычисление по формулам, указанным в таблице 1.1 в соответствии с номером варианта. Номер варианта определяется по последней цифре номера зачетной книжки.

При создании программы выполните следующие этапы:

- преобразуйте формулы с целью уменьшения количества операций при вычислениях. Упрощение возможно как за счет математических преобразований, так и за счет введения дополнительных переменных для сохранения значений выражений, неоднократно встречающихся в формуле;

- для двух заданных вариантов исходных данных вычислите с помощью калькулятора значения по формулам (Переведите калькулятор в режим работы с радианами при вычислении тригонометрических функций). При этом окажется, что одна пара исходных значений не входит в область допустимых значений, т.е. приведет к возникновению ошибки;

- составьте алгоритм (схему) программы. В программе должен быть предусмотрен диалоговый ввод значений входных переменных, вывод результатов с пояснениями;

- в соответствии с алгоритмом составьте программу на языке Pascal;

- запустите программу на счет при обоих вариантах значений входных переменных. Проанализируйте полученные результаты и сообщения среды Turbo Pascal.

Таблица 1.1 – Варианты заданий

№ п/п	Типы переменных	Варианты исходных данных	Задания
0	a, b, y, z – вещественные	а) $a = 5,7$; $b = 0,08$ б) $a = -0,3$; $b = 15,1$	$y = \frac{e^{-a} + \frac{z+10^3}{\sin z}}{\cos \pi z + \ln b}$, $z = b-15,1 $
1	b, x, z – вещественные; y – целая	а) $x=0,166$; $y=4$; б) $x=-3,65$; $y = -3$	$b = \frac{x + 3(x-z) + x^2}{ x-z + \sqrt[3]{y}}$; $z = \ln \frac{1}{y}$
2	a, x, k – вещественные; y – целая	а) $x=-3,06$; $y=3$ б) $x=0,215$; $y = -11$	$a = \frac{e^{-tg \pi k} - \ln x }{kx + 10^5}$, $k = \frac{y+11}{2}$
3	k, x, i – вещественные; m – целая	а) $x=0,02$; $m=3$ б) $x = -1,5$; $m = -10$	$k = \frac{\ln i + 8 \cdot 10^{-2} - \ln i}{\cos \pi i + e^x}$, $i = \frac{m-1}{m+1}$
4	m, n, p – вещественные; r – целая	а) $m = 716,2$; $r = 1$ б) $m=0,07$; $r = -11$	$n = e^{\frac{-\sin m + tg \frac{\pi r}{p}}{p}}$, $p = \ln \left \frac{r^3 + 10^3}{r^3 - 1} \right $
5	d, k, z – вещественные; x – целая	а) $k = -0$; $12 x=0$ б) $k = 1,5$; $x = -11$	$d = \ln \frac{z^2 + 10^3}{z^2 + 1,6 \cdot 10^2}$, $z = e^{-k} ctg kx$
6	m, x, z – вещественные; y – целая	а) $x = 1,016$; $y=11$ б) $x = -0,2$; $y=2$	$m = \sin \arctg \frac{z}{2} - \sin \arctg \frac{\pi}{3}$ $z = \frac{ e^{-xy} + 10^{-3} + e^{-xy}}{\pi + \ln xy}$
7	z, x, m – вещественные; c – целая	а) $x = -0,02$; $c = -3$ б) $x = -1,1$; $c = 2$	$z = tg^2 \frac{\pi m}{m + 10^{-3}}$, $m = e^{-c x + \sqrt{cx}}$
8	K – целая r, t, m – вещественные	а) $r = 0,07$; $k = 6$ б) $r = 0,63$; $k = -15$	$t = \sqrt{\frac{\cos^2 \pi r + k \cdot 10^{-2}}{\cos^2 \pi r + \left \frac{m}{k} \right }}$, $m = \ln \cos \pi r$
9	x, z – целые m, k – вещественная	а) $x = -4$; $z = 12$ б) $x = 2$; $z = -6$	$m = \frac{\sqrt{\ln k + 10^3} - \sqrt{10^3 - \ln k}}{x^2 + 18x - 40}$ $k = e^{\pi x} \cos 0,01z$

1.3. Краткие теоретические сведения

1.3.1. Основные приемы работы в среде Turbo Pascal

Пакет Turbo Pascal 7.0 представляет собой единство двух самостоятельных начал [3]:

- языка программирования – расширения языка Pascal;
- инструментальной программной оболочки – среды, предназначенной для написания, отладки и запуска программ.

Среда Turbo Pascal включает в себя текстовый редактор, компилятор, компоновщик и средства прогона (выполнения) программы [2].

Процесс создания работоспособной программы в среде Turbo Pascal включает следующие основные этапы:

1) Написание текста программы в окне редактирования с помощью редактора среды.

2) Компиляция и компоновка программы. Компиляция текста в машинный код – преобразование исходного текста к виду, доступному для вычислительной машины. Компоновка программы – объединение отдельных частей программы, добавление стандартных подпрограмм, установление необходимых связей между отдельными частями программы. В результате будет получена либо программа, находящаяся в оперативной памяти, либо программа, размещенная в файле на диске. Файл будет иметь то же имя, что и файл с исходным текстом программы, но с расширением .exe.

Компилятор также выявляет ошибки, связанные с нарушением правил написания предложений языка (синтаксические ошибки). При этом компиляция прекращается, выводится сообщение об ошибке, а курсор указывает на место в тексте программы, где находилась ошибка.

3) Отладка – выявление и исправление в программе ошибок, связанных с недопустимыми значениями параметров (семантические ошибки), неправильным использованием алгоритмических конструкций (логические ошибки).

4) Запуск отлаженной программы.

После успешного вызова Turbo Pascal на экране появляется окно интегрированной среды. В верхней части окна редактирования размещается имя открытого файла. Если создается новый файл, ему присваивается условное имя NONAME с порядковым номером, начиная с нуля, и расширение .pas.

Все этапы создания программы в интегрированной среде осуществляются с помощью команд меню, расположенного в верхней строке окна. Назначение нескольких важнейших команд меню описано в таблице 1.2. Доступ к наиболее часто используемым пунктам меню может быть осуществлен еще и с помощью нажатия на клавиатуре сочетаний клавиш быстрого управления («горячих» клавиш).

Таблица 1.2 – Некоторые команды меню среды Turbo Pascal 7.0

Пункт меню		Сочетание клавиш	Назначение	Пояснение
File (Работа с файлами)	New		Создать новый файл	Открывает новое окно редактора
	Open	F3	Открыть файл	Выводится окно диалога, с помощью которого задается имя открываемого файла
	Save	F2	Сохранить файл	Сохраняет файл на диске
	Save as...		Сохранить файл под именем...	Запрашивается имя файла, под которым следует записать информацию на диск
	Change dir...		Изменить текущий каталог	Изменяет текущий каталог, в котором производится поиск файлов и в который записываются файлы
	Exit	Alt+X	Выход	Выход из среды Turbo Pascal
Run (Выполнение программы)	Run	Ctrl+F9	Запустить программу	Запускает программу. Если текст программы был изменен, то при запуске происходит ее перекомпиляция
Compile (Компиляция)	Compile	Alt+F9	Компиляция	Компиляция файла. В случае обнаружения синтаксической ошибки выдается сообщение об этой ошибке.
	Destination (Disk, Memory)		Размещение файла.	Определяет, где следует разместить исполняемый файл программы (*.exe): на диске (Disk) или в оперативной памяти (Memory).
Debug (Отладка)	User Screen	Alt+F5	Окно пользователя	Открывается окно вывода результатов. Для выхода из окна следует нажать любую клавишу.
Help (Помощь)	Topic search	Ctrl+F1	Предметный поиск	Выводится информация о слове, на котором находится курсор.
		F1	Сведения о текущей ситуации	Выдаются сведения о текущем окне, выбранной команде меню, расшифровывается сообщение об ошибке и т.д.

1.3.2. Стандартные арифметические функции

Арифметические функции можно использовать только с величинами целого и вещественного типа. Перечень наиболее часто используемых стандартных функций языка Pascal приведен в таблице 1.3.

Таблица 1.3 – Арифметические функции языка Pascal

Функция	Назначение	Тип результата
Abs (x)	Абсолютное значение (модуль) аргумента	Совпадает с типом x
Arctan (x)	Арктангенс аргумента	Вещественный
Cos (x)	Косинус аргумента*	Вещественный
Exp (x)	e^x	Вещественный
Ln (x)	Натуральный логарифм	Вещественный
Sin (x)	Синус аргумента	Вещественный
Sqr (x)	Квадрат аргумента	Совпадает с типом x
Sqrt (x)	Квадратный корень аргумента	Вещественный

В языке Turbo Pascal определена функция Pi, возвращающая значение величины π .

В языке Pascal нет функций вычисления тангенса и возведения в степень. Для выполнения таких операций можно воспользоваться следующими математическими преобразованиями:

$$\operatorname{tg} x = \frac{\sin x}{\cos x}, \quad (1.1)$$

$$x^y = e^{y \ln x}, \text{ для } y > 0, \quad (1.2)$$

в частности,

$$\sqrt[y]{x} = e^{\frac{1}{y} \ln x}. \quad (1.3)$$

1.3.3. Пример программы

Постановка задачи: Вычислить значение a по формуле

$$a = \frac{\sqrt[3]{|y|}}{(x+1) + (x-1)} + \frac{|\sin x|}{\sqrt[4]{|y|}} \quad (1.4)$$

для $x = 1,241$, $y = -0,879$.

Проанализируем формулу (1.4) с целью выявить возможность упростить вычисления. Знаменатель первой дроби можно упростить следующим образом:

$$(x+1) + (x-1) = 2x. \quad (1.5)$$

Чтобы дважды не вычислять $|y|$, введем дополнительную переменную

$$z = |y|. \quad (1.6)$$

Тогда с учетом соотношений (1.5), (1.6) алгоритм программы будет иметь вид, показанный на рисунке 1.1.

* Аргументы функций sin(), cos(), а также значение функции arctan() рассматриваются как углы, заданные в радианах

Предварительный расчет значения a с помощью калькулятора при заданных значениях x и y дает:

$$z = 0,879, \sin x = 0,9461, a = \frac{0,9579}{2,482} + \frac{0,9461}{0,9013} = 0,3859 + 1,0497 = 1,4357. \quad (1.7)$$

Текст программы на языке Pascal будет выглядеть следующим образом:

```
program lr2;
var  x, y, z,  a: real;
begin
  read(x, y);
  z  :=  abs(y); {z = |y| - вспомогательная переменная}
  a  :=  exp(ln(z)/3)/2/x + abs(sin(x))/exp(ln(z)/x);
  writeln('Результат: a =', a:5:3); {вывод значения a в 5 позициях,
                                     из них 3 позиции после запятой}
end.
```

При заданных x, y программа выведет на экран:

Результат: a = 1.436

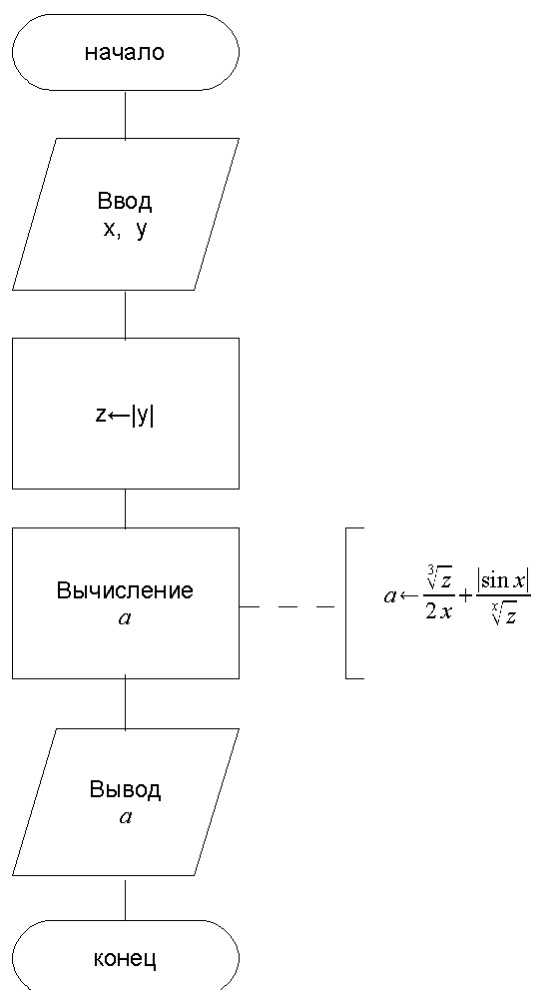


Рисунок 1.1 – Схема программы

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 следующие разделы

- титульный лист;
- цель работы,
- постановка задачи. Этот раздел должен состоять из двух подразделов:
 - «2.1 Исходные данные», где должны быть приведены номер варианта, формула и значения входных переменных;
 - «2.2 Анализ постановки задачи», где должны быть выполнены упрощение вычислений и расчет на калькуляторе с указанием промежуточных результатов;
- схема программы.
- текст программы на языке Pascal;
- результаты работы программы для двух вариантов исходных данных;
- выводы. Этот раздел должен содержать сопоставление результатов вычислений на калькуляторе с результатами работы программы, изложение причины ошибки в случае «плохого» варианта исходных данных, перечень использованных в программе арифметических функций языка Pascal.

Примеры возможных вопросов при защите отчета приведены в подразделе 1.5.

1.5. Контрольные вопросы

- 1) Перечислите состав средств среды Turbo Pascal 7.0.
- 2) Назовите этапы создания программы в среде Turbo Pascal.
- 3) Поясните термин «компиляция программы».
- 4) Поясните термин «синтаксическая ошибка».
- 5) Перечислите разделы программы на языке Pascal.
- 6) Сформулируйте понятие типа данных.
- 7) Приведите основные характеристики типа Integer (типа Real).
- 8) Запишите число 0,000 000 000 043 в экспоненциальном формате по правилам синтаксиса языка Pascal.
- 9) Каковы приоритеты операций при вычислении арифметических выражений?

2. ЛАБОРАТОРНАЯ РАБОТА №2 ПРОГРАММИРОВАНИЕ РАЗВЕТВЛЯЮЩИХСЯ АЛГОРИТМОВ НА ЯЗЫКЕ PASCAL

2.1. Цель работы

Целью данной работы является составление разветвляющихся алгоритмов и их программирование с помощью условного оператора языка Pascal. Закрепление навыков программирования ввода и вывода информации. Приобретение начальных навыков тестирования программ.

2.2. Задание на работу

1) Изучите пример программы, приведенный в разделе 2.3.

2) Создайте программу, реализующую вычисление по формулам, указанным в таблице 2.1 в соответствии с номером Вашего варианта. Для этого выполните следующие этапы:

- преобразуйте формулы с целью уменьшения количества операций при вычислениях. Упрощение возможно как за счет математических преобразований, так и за счет введения дополнительных переменных для сохранения значений выражений, неоднократно встречающихся в формуле;

- составьте алгоритм (схему) программы. В программе должен быть предусмотрен диалоговый ввод значений входных переменных, вывод результатов с пояснениями. Также должен быть организован вывод значений промежуточных переменных, если таковые имеются;

- **самостоятельно подберите** такие **три варианта исходных данных**, чтобы, запустив программу, можно было проверить правильность выбора каждой из ветвей алгоритма;

- в соответствии с алгоритмом составьте программу на языке Pascal;

- запустите программу на счет при трех вариантах значений входных переменных. Проанализируйте полученные результаты и сообщения среды Turbo Pascal.

Таблица 2.1 – Варианты заданий

№ варианта	Задание
0	$m = \begin{cases} \operatorname{ctg} x^2, & \sin x^2 \geq 0,702, \\ \sqrt{\operatorname{tg} x^2}, & -0,2 < \sin x^2 < 0,702 \\ 1,6 \cdot 10^{-8}, & \sin x^2 \leq -0,2 \end{cases}$
1	$i = \begin{cases} 2,3 \cdot 10^{-5}, & \cos^2 x \geq 0,5, \\ 5 \cos^2 x, & 0,25 < \cos^2 x < 0,5, \\ \operatorname{ctg}^2 x, & \cos^2 x \leq 0,25. \end{cases}$
2	$b = \begin{cases} \ln k, & k > \sqrt{2}, \\ -2,6 \cdot 10^3, & -2\sqrt{2} \leq k \leq \sqrt{2} \\ \ln \frac{ k }{\sqrt{2}}, & k < -2\sqrt{2} \end{cases}$
3	$r = \begin{cases} e^{-0,02 m }, & m > \sqrt{ m }, \\ -0,902 \cdot 10^{-12}, & \sqrt{ m } - 12 < m \leq \sqrt{ m }, \\ \sqrt{e^{0,02 m } - \sin \frac{m}{\sqrt{3}}}, & m \leq \sqrt{ m } - 12 \end{cases}$
4	$k = \begin{cases} -0,706 \cdot 10^{-4}, & \operatorname{arctg} z \geq \frac{\pi}{4}, \\ \sin \frac{\operatorname{arctg} z}{2}, & -\frac{\pi}{4} \leq \operatorname{arctg} z < \frac{\pi}{4}, \\ \sin \operatorname{arctg} z , & \operatorname{arctg} z \leq -\frac{\pi}{4} \end{cases}$
5	$a = \begin{cases} \ln \sqrt{ m - 10^{-6} }, & m - 10^{-6} > 0,75, \\ (m - 10^{-6})^4, & 0,2 < m - 10^{-6} \leq 0,75, \\ m - 10^{-6} - 0,3 \cdot 10^{-6}, & m - 10^{-6} \leq 0,2 \end{cases}$
6	$t = \begin{cases} -9,001 \cdot 10^3, & j > 0, \\ \sqrt[4]{-j} + \operatorname{tg} j, & -\sqrt{-j} \leq j \leq 0, \\ \sqrt[4]{-j}, & j < -\sqrt{-j}. \end{cases}$

Таблица 2.1 (продолжение)

№ варианта	Задание
7	$n = \begin{cases} 12,305 \cdot 10^6, & \ln x > 10, \\ \frac{\ln^2 x + \ln x + \sqrt{2}}{\sqrt{2} \ln x + x^2}, & 1 < \ln x \leq 10, \\ \sqrt{\ln x + 1}, & \ln x \leq 1. \end{cases}$
8	$i = \begin{cases} -2 \sin^2 x, & \sin^2 x \geq \frac{1}{2}, \\ -1,6 \cdot 10^{-6}, & \frac{1}{4} < \sin^2 x < \frac{1}{2}, \\ \operatorname{tg}^2 x, & \sin^2 x \leq \frac{1}{4}. \end{cases}$
9	$z = \begin{cases} \sqrt{2} \sin^2 n, & \operatorname{arctg}^2 n > 1, \\ -0,068 \cdot 10^{-10}, & 10^{-5} \leq \operatorname{arctg}^2 n \leq 1, \\ \operatorname{tg} \operatorname{arctg}^2 n, & \operatorname{arctg}^2 n < 10^{-5}. \end{cases}$

2.3. Пример программы

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.

Рассмотрим следующую постановку задачи: составить программу вычисления кусочно-заданной функции

$$y = \begin{cases} 10^{-8} \sqrt{b^2 + 1}, & \text{при } \sqrt{b^2 + 1} > 2, \\ 0,5 \sqrt{b^2 + 1} & \text{при } \sqrt{b^2 + 1} \leq 2. \end{cases} \quad (2.1)$$

Проанализируем формулу (2.1) с целью выявить возможность упростить вычисления. Чтобы многократно не вычислять $\sqrt{b^2 + 1}$, введем дополнительную переменную

$$z = \sqrt{b^2 + 1}. \quad (2.2)$$

Тогда с учетом соотношения (2.2), алгоритм программы будет иметь вид, показанный на рисунке 2.1.

Выберем два таких варианта значений b , чтобы, произведя вычисления на ЭВМ, можно было проверить правильность выбора каждой из ветвей алгоритма. Произведем предварительный расчет значений y с помощью калькулятора:

- 1) $b = 2, \quad z = \sqrt{2^2 + 1} = \sqrt{5} = 2,2361 > 2, \quad y = 2,2361 \cdot 10^{-8};$
- 2) $b = 0, \quad z = \sqrt{0^2 + 1} = 1 \leq 2, \quad y = 0,5.$

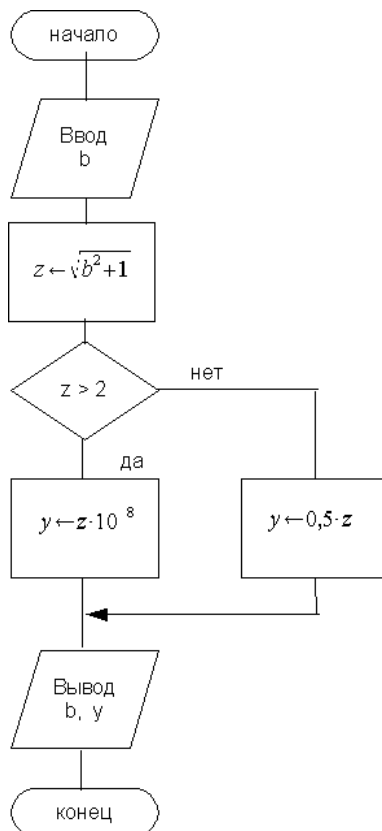


Рисунок 2.1 – Схема программы

Текст программы на языке Pascal будет выглядеть следующим образом:

```

program lr3;
var b, y, z : real;
begin
  write('b='); readln(b);
  z := sqrt(sqr(b)+1.0);
  if z>2 then
    y:=1e-8*z
  else
    y:=0.5*z;
  writeln(' b=',b:5,' y=',y:10)
end.

```

Результаты работы программы для двух вариантов входных данных:

$b = 2.0\text{E}+00$	$y = 2.236\text{E}-08$
$b = 0.0\text{E}+00$	$y = 5.000\text{E}-01$

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 состоять из двух подразделов:
 - «2.1 Вариант задания», где должны быть приведены номер варианта и формула для проведения вычислений;
 - «2.2 Выбор исходных данных», где должны быть приведены три варианта значений переменной и обоснование их выбора;
- схема программы;
- текст программы на языке Pascal;
- результаты работы программы для трех вариантов исходных данных;
- 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) Назовите основные характеристики типа данных Boolean.
- 2) Поясните термин «логическое выражение».
- 3) Перечислите операции отношения.
- 4) Изобразите таблицу истинности для логических операций языка Pascal.
- 5) Перечислите основные блоки, применяемые в схемах разветвляющихся алгоритмов.
- 6) Изобразите синтаксическую диаграмму условного оператора.
- 7) Изложите алгоритм работы условного оператора.
- 8) Что представляет собой составной оператор?
- 9) Каковы синтаксис и алгоритм работы оператора case?
- 10) Сколько будет « $2 \times 2 = 4$ »? :)

3. ЛАБОРАТОРНАЯ РАБОТА №3 ПРОГРАММИРОВАНИЕ ЦИКЛИЧЕСКИХ АЛГОРИТМОВ НА ЯЗЫКЕ PASCAL

3.1. Цель работы

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

3.2. Задание на работу

Варианты заданий приведены в таблице 3.1. В каждом задании определены значения элементов массива и чисел A и B , при которых необходимо осуществить вычисления. До выполнения программы следует определить эти значения без применения ЭВМ, а затем сравнить их с полученными в результате работы программы.

Программа должна удовлетворять следующим требованиям:

- в программе должен быть предусмотрен диалоговый ввод исходных данных: значений элементов массива и чисел A и B . Также должен быть организован вывод элементов массива по пять значений в одной строке;
- затем в программе необходимо осуществить циклический перебор элементов массива и определить три числа в соответствии с заданием. Желательно реализовать все вычисления в рамках одного цикла;
- программа должна осуществлять вывод полученных результатов с комментариями.

Таблица 3.1 – Варианты заданий

№ варианта	Задания и значения исходных данных
0	Вычислить произведение модулей элементов. Найти номер минимального элемента. Вычислить количество элементов, удовлетворяющих условиям: $X_i > B$ или $X_i < A$. $X = 0,2; -3,005; -4,2 \cdot 10^{-3}; 3,2 \cdot 10^3; 0,5; -11; 0,32; -0,9$. $A = -3,5; B = 0,3$.
1	Вычислить сумму модулей элементов. Найти максимальный элемент. Определить количество элементов, лежащих в пределах $A < X_i \leq B$. $X = 15,6; -0,07; 3,5 \cdot 10^4; -0,9 \cdot 10^{-5}; -5; 1,7; 0$. $A = 0; B = 6,1$.

Таблица 3.1 (продолжение)

№ варианта	Задания и значения исходных данных
2	Вычислить сумму модулей элементов. Найти минимальный элемент. Вычислить произведение элементов, удовлетворяющих условиям: $X_i > B$ или $X_i < A$. $X = 0,1; -1,005; -42 \cdot 10^{-8}; 3,2 \cdot 10^3; 0; -11; 0,32; 0$. $A = -1,005; B = 0$.
3	Вычислить сумму модулей элементов. Найти номер максимального элемента. Определить количество элементов, по модулю меньших B. $X = 0; -2; 0; 2,5; -11,04; 0,085; 100; 0; -5 \cdot 10^{-6}; 10^4$. $B = 11,04$.
4	Вычислить сумму модулей элементов. Найти номер минимального элемента. Вычислить сумму элементов, по модулю больших или равны A. $X = 0; 0; 0; -1; 1; 103,6; 10^5; -9,6 \cdot 10^{-5}; -0,75$. $A = 15$.
5	Вычислить сумму квадратов элементов. Найти максимальный по модулю элемент. Определить количество элементов, лежащих в пределах $A \leq X_i < B$. $X = -3,08; 12; 1,5 \cdot 10^3; -4 \cdot 10^{-3}; 4 \cdot 10^{-3}; 0; -1$. $A = 0, B = 12$.
6	Вычислить сумму квадратов элементов. Найти минимальный по модулю элемент. Определить количество элементов, удовлетворяющих условиям $X_i \geq B$ или $X_i \leq A$ $X = 1; -1; 0,2; -2,7 \cdot 10^4; 9 \cdot 10^{-3}; 0,035$. $A = 0, B = 1$.
7	Вычислить сумму квадратов элементов. Найти номер максимального по модулю элемента. Вычислить произведение элементов, меньших или равных A. $X = -109; 0,2 - 10^{-3}; -1,5 \cdot 10^3; 0; -1; 0$. $A = -1$.
8	Вычислить сумму квадратов элементов. Найти номер минимального по модулю элемента. Вычислить сумму элементов, больше или равных A. $X = 10^4; -0,75 \cdot 10^{-4}; 0,76 \cdot 10^{-4}; -1; -10,05$. $A = 10^{-4}$.
9	Вычислить сумму элементов. Найти минимальный из положительных элементов. Определить количество элементов, по модулю меньших B $X = 1; 15,02 - 15; 10^9; -0,88; 0,107; -93 \cdot 10^{-4}; 1$. $B = 1$.

3.3. Краткие теоретические сведения

3.3.1. 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]);
```

3.3.2. Пример программы

Рассмотрим следующую постановку задачи. Задан массив X из 9 чисел и значение переменной A . Все значения – вещественные. Вычислить сумму элементов этого массива, больших A , а также количество и произведение его элементов, меньших $-A$.

В настоящей лабораторной работе необходимо ввести в диалоговом режиме элементы заданного массива X и число A . Ввод простых переменных в диалоговом режиме применялся в предыдущих работах. Ввод элементов массива лучше осуществлять последовательно, по одному значению в каждой строке ввода. Для этого следует организовать цикл, в котором перед вводом каждого очередного элемента на экран выводится его номер. В отличие от ввода, вывод элементов массива по одному значению в строке неудобен. Лучше, чтобы на каждую строку выводилось несколько, например, пять, элементов. Такое расположение значений требует меньше места и более удобно для чтения.

Далее в программе необходимо осуществить перебор элементов массива и определить три числа в соответствии с заданием. Полученный алгоритм будет пред-

ставлять собой цикл, в который вложены несколько идущих последовательно ветвлений. Можно для вычисления каждого числа организовать отдельный цикл и расположить эти циклы последовательно. При этом алгоритм получится более простым и понятным для человека, но программа для выполнения потребует большего машинного времени.

В завершение работы программа должна осуществить вывод полученных значений с комментариями.

Алгоритм решения задачи можно представить с помощью схемы, изображенной на рисунке 3.1.

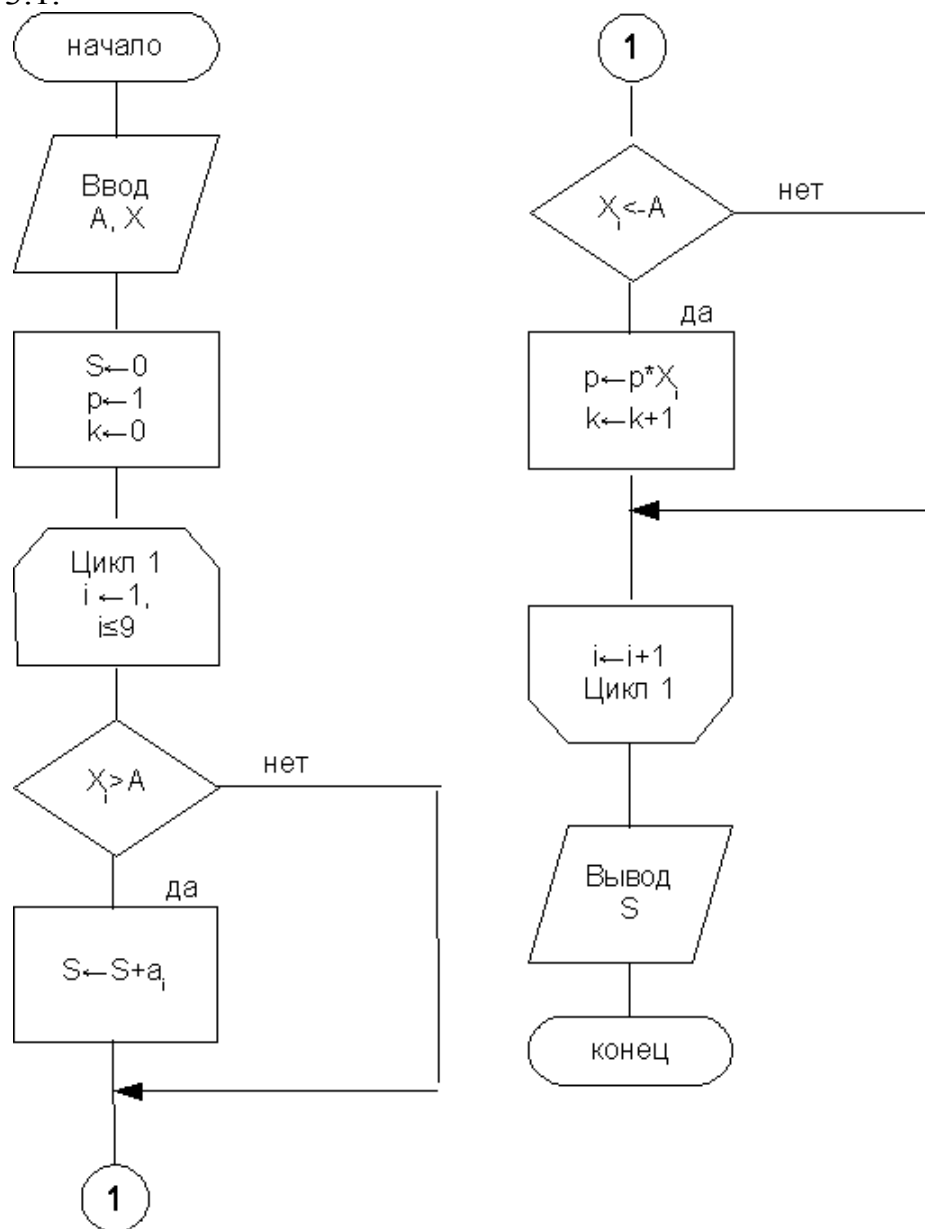


Рисунок 3.1 – Схема программы

Текст программы на языке Pascal будет выглядеть следующим образом:

```

programm massiv;
var s, p, a : real;
    i, k : integer;
    x : array [1..9] of real;
begin
  { Диалоговый ввод a и массива x }

```

```

write('a='); readln(a);
for i := 1 to 9 do
begin
    write('x[',i:1,']='); readln(x[i]);
end;
{ Вывод исходных данных }
writeln('a=', a);
writeln('Массив');
for i:=1 to 9 do
begin
    write(x[i]);
    if i=5 then writeln
end;
{ Вычисление значений s, p, k }
s:=0; p:=1; k:=0;
for i:=1 to 9 do
begin
    if x[i] > a then s := s+x[i];
    if x[i]<-a then
begin
        p := p*x[i]; k := k + 1;
    end;
end;
{ Вывод результатов }
writeln;
writeln ('s=',s, ' p=',p, ' k=', k)
end.

```

3.4. 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 3.1;
- схемы программ. В этом разделе должны быть приведены схемы двух разработанных программ;
- тексты программ на языке Pascal;
- результаты работы программ. Этот раздел должен состоять из двух подразделов. Подраздел «Результаты работы первой программы» должен содержать таблицы значений результатов и графики для трех вариантов x в соответствии с пунктом 4) подраздела 3.2 настоящих методических указаний. Подраздел «Результаты работы второй программы» должен содержать

таблицу значений результатов в соответствии с пунктом 5) подраздела 3.2 настоящих методических указаний.

– 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 3.5.

3.5. Контрольные вопросы

- 1) Опишите синтаксис и алгоритм работы оператора цикла с предусловием в языке Pascal.
- 2) Опишите синтаксис и алгоритм работы оператора цикла с постусловием в языке Pascal.
- 3) Опишите синтаксис и алгоритм работы оператора цикла с параметром в языке Pascal.
- 4) Изложите классификацию типов языка Pascal.
- 5) Какие типы данных относятся к простым, а какие – к структурированным?
- 6) Изложите основные особенности порядковых типов данных.
- 7) Охарактеризуйте перечисляемый тип данных.
- 8) Охарактеризуйте интервальный тип данных.
- 9) Что представляет собой массив? Каковы основные характеристики массива?
- 10) Каков синтаксис объявления типа-массива?
- 11) Перечислите операции, допустимые для переменных-массивов.
- 12) Как осуществляется доступ к элементам массива?
- 13) Что такое индекс массива? Какие допустимы типы индексов массива?
- 14) Изложите алгоритм поиска максимального элемента в массиве.

4. КОНТРОЛЬНАЯ РАБОТА. ПРОГРАММИРОВАНИЕ ОПЕРАЦИЙ НАД МАТРИЦАМИ НА ЯЗЫКЕ PASCAL

4.1. Цель работы

Приобретение навыков алгоритмизации и программирования матричных операций, а также манипулирования внешними файлами.

4.2. Задание на работу

- 1) Изучите примеры программ, приведенных в разделе 4.2.
- 2) Получите у преподавателя номер варианта индивидуального задания на лабораторную работу. Варианты заданий приведены в таблице 4.1.
- 3) Программа должна удовлетворять следующим требованиям:
 - в лабораторной работе необходимо реализовать процедуру и функцию, реализующие вычисления, в соответствии с номером варианта, указанным в таблице 4.1;
 - 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 Д.
 - 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 4.1 – Варианты заданий

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

Таблица 4.1 (продолжение)

№ варианта	Задания и значения исходных данных			
	Процедура	Функция	Матрицы	Ввести с клавиатуры
4	Умножение	максимальный среди элементов ниже главной диагонали	C, D	D
5	Сложение	Сумма модулей элементов выше главной диагонали	A, B	A
6	Транспонирование	Последний положительный среди элементов ниже главной диагонали	C, D	D
7	Умножение	Корень квадратный из суммы квадратов элементов	A, B	A
8	Сложение	Максимальный среди элементов, расположенных в нечетных столбцах	E, F	F
9	Транспонирование	Минимальный элемент среди положительных	C, D	C

4.3. Краткие теоретические сведения

4.3.1. 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;
    end;

```

... ..

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 [4] с указанием соответствующих начальных и конечных значений в заголовках циклов.

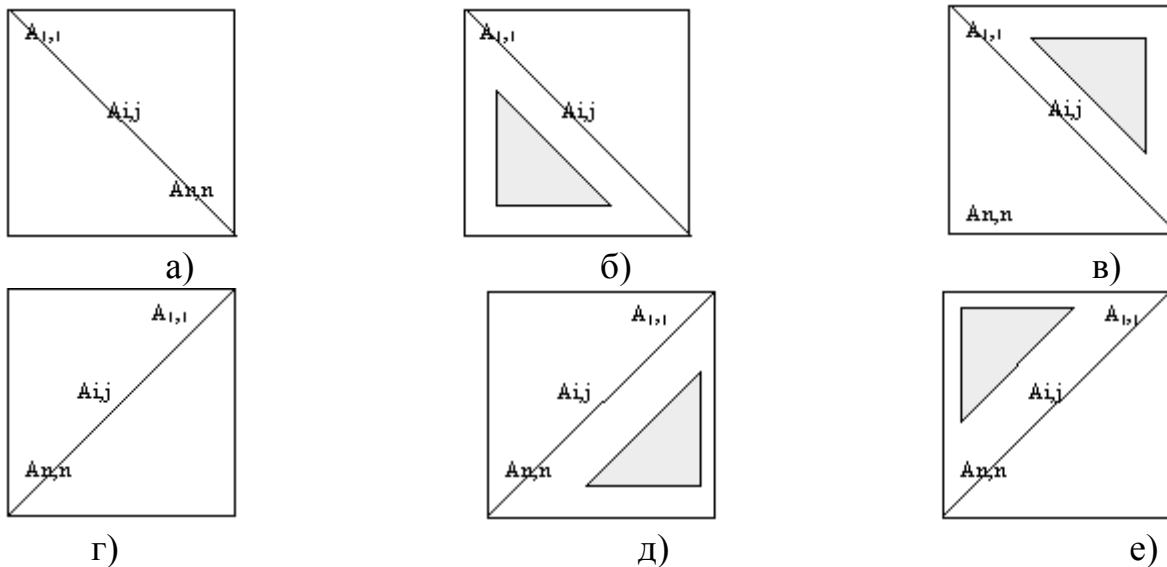


Рисунок 1.1 – Определение индексов элементов квадратной матрицы

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, принадлежащие 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]

```

4.3.2. 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.

4.3.3. 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 Паскале 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.

В Паскале существует ограниченное количество стандартных функций. Например, `sin(x)`, `odd(i)`, `sqrt(x)`.

Итак, функция вычисляет одно значение, которое передается через ее имя.

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)]

Вызов функции включают в некоторое выражение.

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

4.3.4. 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 .

```

4.4. 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 4.1;
- 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 4.5.

4.5. 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 такое текстовый файл?
- 7) Для чего предназначены и как работают процедуры Assign и Close?
- 8) Для чего предназначены и как работают процедуры Reset и Rewrite?

- 9) Для чего предназначены и как работают процедуры Read и Readln?
- 10) Для чего предназначены и как работают процедуры Write и Writeln?
- 11) Для чего предназначены и как работают функции Eof и Eoln?
- 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?
- 17) Каковы особенности передачи параметров-массивов?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Павловская Т.А. Паскаль. Программирование на языке высокого уровня /Т.А. Павловская. – СПб.: Питер, 2007. – 393 с.
2. Епанешников А.М. Программирование в среде Turbo Pascal 7.0 /А.М. Епанешников, В.А. Епанешников. – М.: «Диалог-МИФИ», 1996. – 288с.
3. Фаронов В.В. Турбо Паскаль 7.0. Начальный курс/ В.В. Фаронов – М.: “Номидж”, 1997. - 616 с.
4. Зайцева Л.В. Алгоритмические языки и программирование /Л.В. Зайцева. – [Электронный ресурс]. – http://www.nesterova.ru/bibl/algorithm_lang/Kniga/index.html.
5. ГОСТ 19.701-90 ЕСПД. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения – М.: Изд-во стандартов, 1991. –26с.
6. ДСТУ 3008-95. Документация. Отчеты в сфере науки и техники. Структура и правила оформления. – К.: Госстандарт Украины, 1995. – 37с.

**ПРИЛОЖЕНИЕ А.
ПРИМЕР ОФОРМЛЕНИЯ ТИТУЛЬНОГО ЛИСТА ОТЧЕТА**

(справочное)

Министерство образования и науки Украины
Севастопольский национальный технический университет
Кафедра технической кибернетики

ОТЧЕТ
по лабораторной работе №1
«Программирование линейных алгоритмов на языке Pascal»
по дисциплине
«Алгоритмические языки и программирование»

Выполнил: студент гр. А-11з
Иванов И.И.

Проверил: ст. преп. кафедры ТК
Старинская Д.Н.

Севастополь
2008

ПРИЛОЖЕНИЕ Б. ПРИМЕР ОФОРМЛЕНИЯ ОСНОВНОЙ ЧАСТИ ОТЧЕТА

(справочное)

1 ЦЕЛЬ РАБОТЫ

Целью данной работы является приобретение навыков составления и программирования линейных алгоритмов, реализация вычисления сложных арифметических выражений с помощью операторов языка Pascal.

2 ПОСТАНОВКА ЗАДАЧИ

2.1 Исходные данные

Вариант № 19.

Вычислить значение a по формуле

$$a = \frac{\sqrt[3]{|y|}}{(x+1) + (x-1)} + \frac{|\sin x|}{\sqrt[3]{|y|}}$$

а) для $x = 1,241$, $y = -0,879$;

б) для $x = 2,5$, $y = 0$.

2.2 Анализ постановки задачи

Предварительный расчет значения a с помощью калькулятора при заданных значениях x и y дает:

$$\text{а) } |y| = 0,879, \quad \sin x = 0,9461, \quad a = \frac{0,9579}{2,482} + \frac{0,9461}{0,9013} = 0,3859 + 1,0497 = 1,4357.$$

б) $|y| = 0$, следовательно, знаменатель выражения $\frac{|\sin x|}{\sqrt[3]{|y|}}$ равен нулю. Таким образом, пара значений $x = 2,5$, $y = 0$ не входит в ОДЗ.

3 СХЕМА ПРОГРАММЫ

Схема программы приведена на рисунке 3.1.

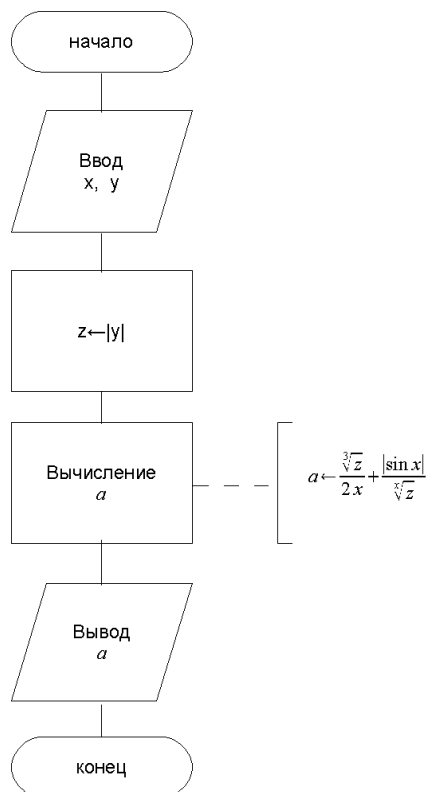


Рисунок 3.1 – Схема программы

4 ТЕКСТ ПРОГРАММЫ

```

program lr2;
var  x, y, z,  a: real;
begin
  read(x, y);
  z  :=  abs(y); {z = |y|- вспомогательная переменная}
  a  :=  exp(ln(z)/3)/2/x + abs(sin(x))/ exp(ln(z)/x);
  writeln('Результат: a =', a:5:3);
end.

```

5 РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ

При $x = 1,241, y = -0,879$ программа выведет на экран:

Результат: a = 1.436

При $x = 2,5, y = 0$ на экран будет выведено сообщение об ошибке:

Runtime error 207 at 0BED:0076.

ВЫВОДЫ

При выполнении лабораторной работы разработана программа, реализующая вычисление сложного арифметического выражения по заданной формуле. При $x = 1,241, y = -0,879$ программа дает результат $a = 1.436$, что согласуется с результатом вычислений на калькуляторе. При $x = 2,5, y = 0$ происходит ошибка вследствие деления на нуль.

ПРИЛОЖЕНИЕ В. ЭКЗАМЕНАЦИОННЫЕ ВОПРОСЫ

(справочное)

1. Переменные и константы в языке Pascal. Разделы описания переменных и констант. Понятие типа данных.
2. Понятие типа данных. Данные целого типа в языке Pascal.
3. Понятие типа данных. Данные вещественного типа в языке Pascal. Особенности машинной арифметики.
4. Арифметические операции. Математические функции языка Pascal.
5. Разветвляющиеся алгоритмы, условный оператор в языке Pascal. Составной оператор.
6. Данные логического типа в языке Pascal. Логические операции и выражения в языке Pascal.
7. Оператор выбора варианта `case`.
8. Циклические алгоритмы. Оператор цикла с предусловием в языке Pascal. Пример.
9. Циклические алгоритмы. Оператор цикла с постусловием в языке Pascal. Пример.
10. Циклические алгоритмы. Оператор цикла с параметром в языке Pascal. Пример.
11. Процедуры в языке Pascal. Объявление процедуры. Оператор процедуры.
12. Локальные и глобальные переменные в языке Pascal. Точка определения, область видимости.
13. Процедуры с параметрами в языке Pascal. Список формальных параметров. Список фактических параметров.
14. Процедуры с параметрами в языке Pascal. Механизмы передачи параметров (параметры-значения, параметры-переменные).
15. Функции в языке Pascal. Примеры.
16. Классификация типов данных в языке Pascal.
17. Общая характеристика порядковых типов в языке Pascal. Символьный тип данных в языке Pascal.
18. Общая характеристика порядковых типов в языке Pascal. Данные перечисляемого типа в языке Pascal. Данные интервального типа в языке Pascal.
19. Тип данных массив в языке Pascal. Тип индекса и тип элементов. Объявление переменных массивового типа. Многомерные массивы.
20. Примеры алгоритмов обработки числовых массивов на языке Pascal. Поиск максимального элемента в массиве. Заполнение матрицы значениями.
21. Примеры алгоритмов обработки числовых массивов на языке Pascal. Транспонирование и умножение матриц.
22. Файловый тип данных в языке Pascal. Стандартные процедуры и функции языка Pascal для работы с типизированными файлами.
23. Текстовые файлы в языке Pascal. Процедуры и функции для работы с текстовыми файлами. Стандартные файлы `Input` и `Output`.
24. Текстовые файлы в языке Pascal. Форматированный вывод. Примеры.

ПРИЛОЖЕНИЕ Г.

ОБЩИЕ ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ СХЕМ ПРОГРАММ

(справочное)

Требования основаны на стандарте ГОСТ 19.701 – 90 («Схемы алгоритмов, программ, данных и систем») [4], который распространяется на условные обозначения в схемах алгоритмов, программ, данных и систем и устанавливает правила выполнения схем, используемых для отображения различных видов задач обработки данных и средств их решения.

Схемы состоят из имеющих заданное значение символов, краткого пояснительного текста и соединяющих линий.

В схемах программ ГОСТ разрешает использовать любые символы процесса, символы линий и специальные символы, а также символ «Данные» (в виде параллелограмма) для отображения операций ввода и вывода

Формы символов определены стандартом. Размеры их должны быть, по возможности, одинаковы. Вычерчивать символы желательно горизонтально, а большинство символов должно содержать внутри минимальное количество текста, необходимого для понимания его функции. Символы в схеме должны быть расположены равномерно и соединены линиями разумной длины, длинных соединительных линий желательно не иметь вовсе.

Направления входного потока слева и сверху, а выходного потока вправо и вниз считаются *стандартными*. На линиях с нестандартными направлениями потока необходимо указывать стрелки, а на линиях со стандартными направлениями – можно указывать стрелки для пояснения.

Линия должна быть направлена к центру символа.

Условные обозначения в схемах программ, данных и систем (ГОСТ 19.701–90) приведены в таблицах Г1 – Г3.

Таблица Г.1 – Символы данных

Данные 	Документ 	Ручной ввод 	Карта 
Запоминаемые данные 	Оперативное ЗУ 	ЗУ с прямым доступом 	ЗУ с последоват. выборкой 
Бумажная лента 	Дисплей 	<i>Рекомендуемое отношение сторон большинства символов – 3/2</i>	

Таблица Г.2 – Символы процесса

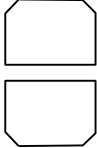
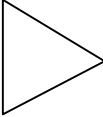
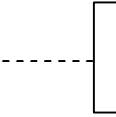
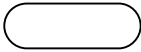
Процесс 	Предопределенный процесс 	Ручная операция 	Подготовка 
Решение 	Границы цикла 	Параллельные действия 	

Таблица Г.3 – Линии и специальные символы

Линия 	Передача управления 	Комментарий 	Канал связи 
Терминатор 	Соединитель 	Пропуск 	

ПРИЛОЖЕНИЕ Д **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

```


Заказ № _____ от « _____ » _____ 2008 Тираж _____ экз.
Изд-во СевНТУ