МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«ВИТЕБСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Информатика. Основы программирования

Методические указания и задания к лабораторным работам
для студентов ХТФ специальности 1-50 01 01
«Технология пряжи, тканей, трикотажа и нетканых материалов»

Витебск
2010

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
«ВИТЕБСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

СОГЛАСОВАНО:	**УТВЕРЖДАЮ:**
Зам. председателя редакционно-издательского Совета УО «ВГТУ»	Первый проректор УО «ВГТУ»
_____ **В.В. ПЯТОВ**	_____**С.И. Малашенков**
“____”_____ 2010 г.	“____”_____ 2010 г.

Информатика. Основы программирования

Методические указания и задания к лабораторным работам

для студентов ХТФ специальности 1-50 01 01

«Технология пряжи, тканей, трикотажа и нетканых материалов»

Витебск

2010

УДК 004.43(075)

Информатика. Основы программирования: методические указания и задания к лабораторным работам для студентов ХТФ специальности 1-50 01 01 «Технология пряжи, тканей, трикотажа и нетканых материалов»

Витебск: Министерство образования Республики Беларусь, УО «ВГТУ», 2010

Составитель: ст.пр. Стасеня Т.П.

В методических указаниях рассмотрены вопросы программирования на языке Паскаль. Методические указания предназначены для использования при выполнении лабораторных работ студентами ХТФ специальности 1-50 01 01 «Технология пряжи, тканей, трикотажа и нетканых материалов» по курсу «Информатика» и будут полезны для студентов других специальностей при изучении основ программирования.

Одобрено кафедрой информатики УО «ВГТУ»
1 сентября 2010 г., протокол № 1

Рецензент: Казаков В.Е.
Редактор: Терентьев В.П.

Рекомендовано к опубликованию редакционно-издательским советом
УО «ВГТУ» __ _____ 2010 г., протокол № ___

Ответственный за выпуск: Соколов И.В.

Учреждение образования «Витебский государственный технологический университет»

Подписано к печати _____ Формат_____ Уч-изд. лист. _____
Печать ризографическая. Тираж ____ экз. Заказ № _____ Цена _____

Отпечатано на ризографе учреждения образования «Витебский государственный технологический университет».
Лицензия 02330/0494384 от 16 марта 2009 года
210035, Витебск, Московский пр., 72.

СОДЕРЖАНИЕ

Лабораторная работа № 1

Тема. Разработка блок-схем алгоритмов различных вычислительных процессов

Цель работы: приобрести практические навыки разработки блок-схем линейных, разветвляющихся и циклических процессов

Теоретическая часть

Алгоритм − это точное предписание о порядке выполнения действий над исходными данными для получения требуемого результата.

Наиболее распространенным способом представления алгоритма является ***графический***. В графическом представлении алгоритмы изображаются в виде блок-схемы, дополненной элементами словесной или математической записи.

Схема алгоритма включает геометрические фигуры (блочные символы), соединенные между собой стрелками (линиями), указывающими порядок выполнения операций.

Блочные символы стандартизированы и различаются по типу выполняемых действий (ГОСТ 19.002−80 и 19.003−80, международные стандарты ИСО 2636-73 или ИСО 1028−73).

Единственное ограничение накладывается на последовательность записей − они должны читаться слева направо и сверху вниз независимо от направления потоков информации.

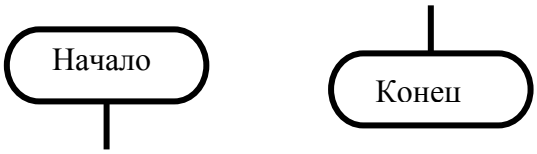
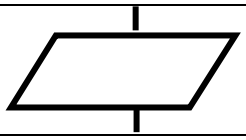
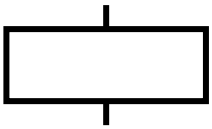
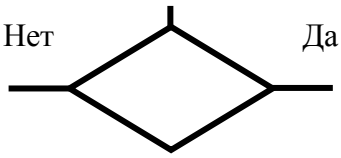
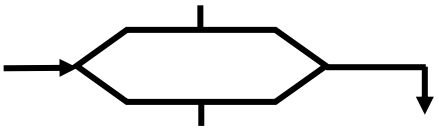
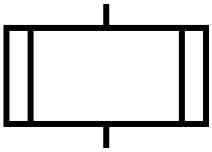
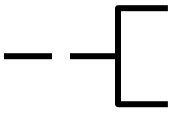
Логика алгоритма должна опираться на минимальное число достаточно простых управляющих базовых структур. При разработке схем алгоритмов необходимо соблюдать некоторые требования:

- В схеме алгоритма все линии от блока «начало» до блока «конец» не должны иметь разрывов, не помеченных соединителями. Все линии, указывающие последовательность выполнения действий, должны быть замкнутыми.
- В схеме должны четко прослеживаться потоки информации. Блоки следует размещать таким образом, чтобы избегать пересечения линий. При передаче управления в схеме «снизу−вверх» или «справа−налево» линии обязательно помечают стрелками.
- Блоки следует размещать таким образом, чтобы избегать пересечения линий. При передаче управления в схеме «снизу−вверх» или «справа−налево» линии обязательно помечают стрелками.
- Не допускается передача управления в никуда. «Источник» передачи управления и «получатель» должны быть четко обозначены.

Виды структур алгоритмов

По структуре алгоритмы разделяют на линейные, алгоритмы с ветвлением и циклические.

Таблица 1 – Виды и значения алгоритмических блоков

№ п/п	Изображение	Значение
1		Блоки начала и конца программы
2		Блок ввода или вывода информации
3		Блок вычислений
4		Логический блок
5		Блок модификации (начало цикла)
6		Блок подпрограммы
7		Внутристраничные соединительные блоки
8		Межстраничные соединительные блоки
9		Блок комментариев

Линейные алгоритмы

Линейными называют алгоритмы, в которых операции выполняются последовательно одна за другой, в естественном и единственном порядке следования. В таких алгоритмах все блоки имеют последовательное соединение логической связью передачи информационных потоков. В них могут использоваться все блоки, за исключением блоков проверки условия и модификации. Линейные алгоритмы, как правило, являются составной частью любого алгоритмического процесса.

Используя общепринятые символы блоков, на рис.1 изображена блок-схема линейного алгоритма.

Алгоритмы с ветвлением

При составлении схем алгоритмов часто возникает необходимость проведения анализа исходных данных или промежуточных результатов вычислений и определения дальнейшего порядка выполнения вычислительного процесса в зависимости от результатов этого анализа. Алгоритмы, в которых в зависимости от выполнения некоторого логического условия происходит разветвление вычислений по одному из нескольких возможных направлений, называют ***разветвляющимися, или алгоритмы с ветвлением***. Подобные алгоритмы предусматривают выбор одного из альтернативных путей продолжения вычислений. Каждое возможное направление вычислений называется ветвью. Логическое условие называют ***простым***, если разветвляющийся процесс имеет две ветви, и ***сложным,*** если процесс разветвляется на три и более ветви.

На рис. 2 приводится пример разветвляющегося алгоритма с простым логическим условием. Для определения ветви, по которой необходимо производить процесс вычисления значения ***x,*** достаточно проверить выполнение одного условия. Если условие $x > 3$ не выполняется, то очевидно и без дополнительной проверки, что будет выполнено условие $3 \le x$.

На рис. 3 приводится пример сложного алгоритма с ветвлением.

Циклический алгоритм

Алгоритм ***циклической*** структуры предусматривает многократное повторение действий в одной и той же последовательности по одним и тем же математическим зависимостям, но при разных значениях некоторой специально изменяемой величины. Циклические алгоритмы позволяют существенно сократить объем программы за счет многократного выполнения группы повторяющихся вычислений, так называемого ***тела цикла.***

Специально изменяемый по заданному закону параметр, входящий в тело цикла, называется ***переменной цикла.*** Переменная цикла используется для подготовки очередного повторения цикла и отслеживания условий его окончания. В качестве переменной цикла используют любые переменные, индексы массивов, аргументы вычисляемых функций и тому подобные величины цикла.

Циклы, в теле которых нет разветвлений и других встроенных в них циклов, называют ***простыми***. *В* противном случае их относят к ***сложным.*** Циклические алгоритмы разделяют на ***детерминированные*** и ***итерационные.***

Циклы, в которых число повторений заранее известно из исходных данных или определено в ходе решения задачи, называют *детерминированными.* Для организации детерминированных циклов наиболее целесообразно использовать блок модификации, внутри которого указывается переменная цикла, ее начальное и конечное значения. Организовать подобный цикл возможно и при использовании блока проверки условия вместо блока модификации, однако при этом несколько усложняется алгоритм и теряется его рациональность.

На рис. 4 пример циклического алгоритма с использованием блока модификации. Операция нахождения суммы, при предварительном обнулении значения переменной S (блок 5), повторяется 10 раз в теле цикла.

Использована операция присваивания **S := S + X * I,** по которой и осуществляется вычисление суммы путем прибавления к предыдущему значению переменной S всё новых значений элементов ряда.

Цикл является детерминированным, и количество его повторений заранее определено (**10** раз). В качестве переменной цикла **i** принято текущее значение членов натурального ряда.

На рис. 8 блок-схема алгоритма для расчета суммы бесконечного ряда. Вычисление суммы прекратить, как только значение очередного элемента ряда станет меньше или равно **e (e = 0,00001),** и значениями остальных элементов ряда можно пренебречь.

На рис. 5 для примера 9.1 организован цикл в виде итерационного, т.к. число повторений заранее неизвестно. В алгоритме выход из цикла или его продолжение определяется выполнением условия delta <= e в блоке 5. Если условие не выполняется, то вычисление суммы продолжается путем прибавления к предыдущему значению суммы (переменная S) значения очередного члена ряда, отслеживаемого переменной цикла i.

На рис. 7 разработана блок-схема табулирования функции, заданной на отрезке **[a, b],** где **h** шаг приращения аргумента x, значение константы **d** вводится с клавиатуры.

Вводятся исходные данные (**a и b** – границы интервала табулирования функции ***y,*** **h** – приращение аргумента, **d** – постоянная величина), затем задается начальное значение аргумента функции (***x*** **= a**) и вычисляется значение функции ***y***, которая задана системой равенств. Значения аргумента ***x*** и функции **y** выводятся на экран. Для расчета следующего значения функции аргумент ***x*** получает приращение **h** (блок 12).

После выполнения первой итерации управление передается на начало цикла, где проверяется условие повторения цикла. Если оно выполняется, то расчет функции повторяется, в противном случае происходит выход из цикла.

Логические блоки 6 и 7 определяют функцию расчета ***y***. При $a \le x < 2$ значение ***y*** рассчитывается по формуле $y = 12d + x + \cos(x^3)$ (блок 10), при $2 \le x \le 4$ – по формуле $y = \ln|10 - x^2|$ (блок 9), при $4 < x \le b$ – по формуле $y = 4 + tg(x^2)$ (блок 8).

Вид итерационного цикла (с пост- или предусловием) определяется условием задачи и допустимыми или возможными значениями исходных данных.

При организации цикла с постусловием необходимо помнить, что при любых начальных значениях исходных данных тело цикла обязательно будет выполнено хотя бы один раз. Если же организовать цикл с предусловием, то необходимо быть уверенным в том, что начальные значения исходных данных позволяют проверить условие выхода из цикла без его выполнения.

Задание

Разработать блок-схемы алгоритмов для индивидуальных заданий из лабораторных работ 3, 4.

Лабораторная работа № 2

Тема. Запись математических и логических выражений на языке Паскаль

Цель работы: научиться записывать математические и логические выражения на языке Паскаль

Теоретическая часть

Для записи математических формул в языке Паскаль используются выражения. Выражение задает порядок выполнения действий над элементами данных и состоит из операндов (констант, переменных, обращений к функциям), круглых скобок и знаков операций. Операции определяют действия, которые надо выполнить над операндами. Круглые скобки ставятся как и в обычных арифметических выражениях для управления порядком выполнения операций. В языке Паскаль принят приоритет операций. Вначале выполняются операции: умножения, деления (***, /, MOD, DIV**), логические операции **AND, OR**, выражения в круглых скобках.

Таблица 2 – Запись некоторых математических функций на языке Паскаль

Вычисляемая функция	Математическая запись	Запись на языке Паскаль
x^y	$e^{y*\ln x}$	exp(y*ln(x))

Вычисляемая функция	Математическая запись	Запись на языке Паскаль
$\sqrt[y]{x}$	$x^{1/y} = e^{1/y * \ln x}$	exp(1/y*ln(x))
arcsin(x)	$arctg \frac{x}{\sqrt{1-x^2}}$	arctan(x/sqrt(1-sqr(x)))
arccos(x)	$arctg \frac{\sqrt{1-x^2}}{x}$	arctan(sqrt(1-sqr(x))/x)
$log_y(x)$	lnx/lny	ln(x)/ln(y)
tg(x)	sinx/cosx	sin(x)/cos(x)

В языке Паскаль есть встроенные функции, которые используются в арифметических выражениях. Однако многих часто используемых математических функций в Паскале не существует, и программист должен самостоятельно вычислить их, используя стандартные функции.

Существует ряд констант и функций, к значениям которых можно обращаться без предварительного определения: функция **π** – **Pi;** константа логическая «истина» – **True;** константа логическая «ложь» – **False**.

Все значения углов, используемые в функциях языка Паскаль, представляются в радианах, для преобразования значения угла из радианной меры в градусную необходимо значение угла умножить на число **180/Pi.**

Таблица 3 - Примеры записи математических и логических выражений на языке Паскаль

Математическая запись	Запись на языке Паскаль
$y = \frac{a + tg(b/4 - 1)}{4 * c - \lg(b + 1)}$	y:=(a+sin(b/4-1)/cos(b/4-1))/(4*c-ln(b + 1)/ln(10));
$y = \frac{b^{a+4} + c/2}{c/3 - a * b}$	y:=(exp((a+4)*ln(b))+c/2)/(c/3-a*b);
$0 \le x \le 5$	(0<=x) and (x<=5)
$5 \ge x > 0$	(5>=x) and (x>0)
$0 \le x \le 5$ или $10 \le x \le 25$	((0<=x) and (x<=5)) or ((10<=x) or (x<=25))

Программы в текстовом редакторе можно писать строчными или прописными буквами, символы кириллицы (русские буквы) используются только для вывода текста и в комментариях.

Задание

Записать 2 математических выражения на языке Паскаль (из заданий к лабораторной работе № 3, табл. 4, 5).

Лабораторная работа № 3

Тема. Реализация линейных вычислительных процессов на языке Паскаль

Цель работы: научиться создавать программы на языке Паскаль с линейной структурой

Теоретическая часть

Программы с линейной структурой являются простейшими и используются для реализации обычных вычислений по формулам (рис. 1). В программах с линейной структурой инструкции выполняются последовательно, одна за другой.

Пример3.1. Написать программу вычисления функции **Y(a,c,d).** Значения **a, c, d** вводятся с клавиатуры.

$$y = \frac{tgc - d * 23}{a^{d-2} - 1}$$

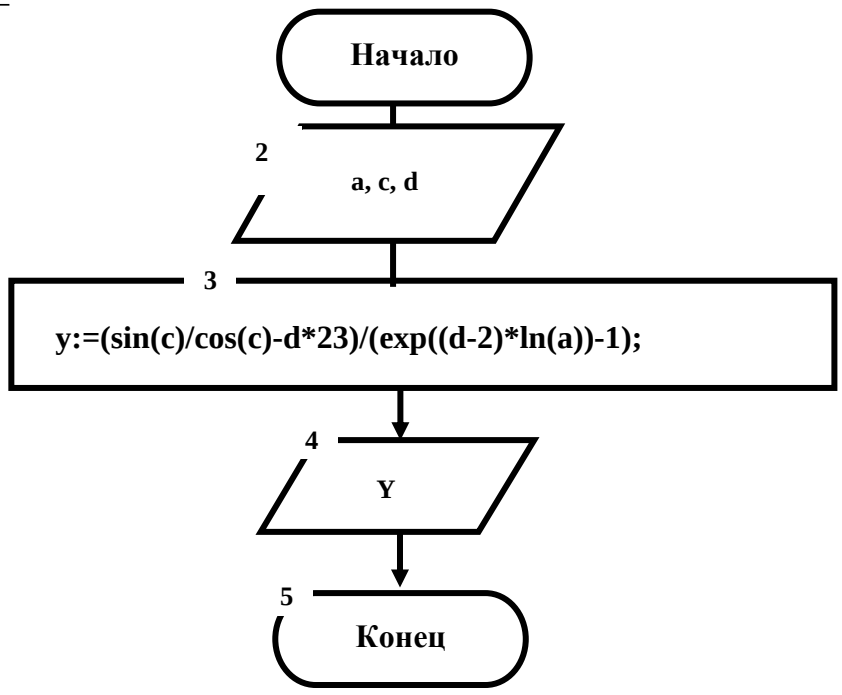


Рисунок 1

Текст программы на языке Pascal

```
Program Project1;                          //заголовок программы
var a,c,d,y: real;                         //раздел описания переменных
begin                                      //начало раздела операторов
writeln('Введите значения a, c, d ');      //вывод текста на экран
readln(a,c,d);                             // ввод значений a,c,d
                                  //вычисление значения y
                          //и присваивание полученного значения переменной y
y:=(sin(c)/cos(c) – d * 23)/(exp((d - 2) * ln(a)) - 1);
                                  //вывод полученного значения на экран
writeln ('Значение функции y= ', y:5:2);
readln;                                    //задержка экрана для просмотра результата
end.                                       //конец программы
```

Содержание отчета

1. Название, цель работы и задание.
2. Блок-схема алгоритма и текст программы.

Варианты заданий (2 задания) в табл. 4, 5.

Таблица 4 - Задание 1

№	Функция	Проверочное значение y(x) при x = 2 (или y(x1, x2) при x1 = 2 и x2 = 2)
1	$y = \frac{5x + \sin x^2}{2x + tgx} + \lvert \sin x^2 \rvert$	5,8496
2	$y = \frac{\log_3 \lvert x - 10 \rvert}{2x - 7} - \cos(3x + 5)$	-0,6354
3	$y = \frac{\cos(x_1 + 5) + x_2}{3x_1 + 6x_2} * \lvert \sin \chi_1^{\,2} \rvert$	0,1158
4	$y = \frac{\cos(3x + 5) + 5^x}{4.35x} + \lvert \cos x^2 \rvert + \frac{5}{x}$	6,027716

№	Функция	Проверочное значение y(x) при x = 2 (или y(x1, x2) при x1 = 2 и x2 = 2)
5	$y = \frac{3x_1 + 2^{x_2}}{\sin(x_1 + x_2)} + \frac{5}{\lvert \sin x2 \rvert} + 6$	-1,7147
6	$y = \frac{\sin^2(x + 3.5) + \lg x}{3x + e^{x+1.35}}$	0,0232
7	$y = \frac{x_1 + 5.5x_2}{\cos x_1 + \ln x_2} + tg(x_1 + 4.3)$	46,9482
8	$y = \frac{3^{x-1.4} + e^x}{4.5 + x} + tg\,3x$	1,1432
9	$y = \frac{x_1^{x_2} + 3.5tgx_1}{3x_1 + 5x_2} + \frac{5\,x_1}{x_2 + 6} + 5$	6,022
10	$y = \frac{\cos^3(x + 2.5) + 1.5x}{x^2 + 3.5} + \frac{3x^2}{8x} + 8.5$	9,6488
11	$y = \frac{\ln(x + 1.3) + 5}{1.35x^2 + 6} + \sqrt{\frac{x + 1.3}{(35x)^2 + 6}}$	0,5693
12	$y = \frac{2x_1 + 1.4^{x_2}}{tgx_1 + 2x_2} + \sqrt[3]{6x_1 + 8^{x2}}$	7,5196
13	$y = \frac{3x + tgx}{2.36x + 6} + 2x + 1.4^x$	6,3159

№	Функция	Проверочное значение y(x) при x = 2 (или y(x1, x2) при x1 = 2 и x2 = 2)
14	$y=\frac{e^{x-1.3}+\sin x}{x+3.5}+2x+1.4\sin x$	5,8045
15	$y=\frac{x+3\cos(x^2+1.5)}{tgx+4.56}+\sin x\frac{2+x}{\sqrt{5^x+56x}}$	2,0480
16	$y=\frac{x+3\cos(x^2+1.5)}{tgx+4.56}+\sin(2x)+\cos(x^2+5)$	0,0694
17	$y=\frac{\cos^2(x_1+1.3)+x_2}{2x_1+e^{x_2}}+\cos({x_2}^2+1.5)+\frac{x_2}{\cos({x_1}^2+1.5)}$	3,7921
18	$y=\frac{5x_1+1.3^{x_2}}{\cos(x_1+x_2)}+\cos({x_2}^2+1.5)+\frac{5+x_2}{\sqrt{x_1}}$	-12,2259

Таблица 5 – Задание 2

№	Функция и проверочные данные
1	$t=\frac{2\cos\left(x-\frac{\pi}{6}\right)}{0.5+\sin^2 y}\left(1+\frac{z^2}{3-z^2/5}\right).$
	При x=14.26, y=-1.22, z=0.035 t=0.564849
2	$u=\frac{\sqrt[3]{8+\lvert x-y\rvert^2+1}}{x^2+y^2+2}-e^{\lvert x-y\rvert}\left(tg^2z+1\right)^x.$
	При x=-4.5, y=0.000075,z=84.5 u=-55.6848
3	$v=\frac{1+\sin^2(x+y)}{\left\lvert x-\frac{2y}{1+x^2y^2}\right\rvert}x^{\lvert y\rvert}+\cos^2\left(arctg\frac{1}{z}\right).$
	При x=0.0374, y=-0.825, z=16, *v*=1.0553

№	Функция и проверочные данные
4	$w = \left\|\cos x - \cos y\right\|^{\left(1+2\sin^2 y\right)}\left(1 + z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4}\right).$
	При x=4000 ,y=-0.875, z=-0.000475 *w*=1.9873
5	$\alpha = \ln\left(y^{-\sqrt{\left\|x\right\|}}\right)\left(x - \frac{y}{2}\right) + \sin^2 arctg(z).$
	При x=-15.246, y=0.04642, z=2000.1 α=-182.036
6	$\beta = \sqrt{10\left(\sqrt[3]{x} + x^{y+2}\right)}\left(\arcsin^2 z - \left\|x - y\right\|\right).$
	При x=0.01655, y=-2.75, z=0.15 β=-40.6307
7	$\gamma = 5arctg(x) - \frac{1}{4}\arccos(x)\frac{x + 3\left\|x - y\right\| + x^2}{\left\|x - y\right\| z + x^2}.$
	При x=0.1722, y=6.33, z=0.000325 γ=-205.3056
8	$\varphi = \frac{e^{\left\|x-y\right\|}\left\|x - y\right\|^{x+y}}{arctg(x) + arctg(z)} + \sqrt[3]{x^6 + \ln^2 y}.$
	При x=-0.02235, y=2.23, z=15.221 φ=39.374
9	$\psi = \left\| x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}} \right\| + (y - x)\frac{\cos y - \frac{z}{(y - x)}}{1 + (y - x)^2}.$
	При x=182.5, y=18.225, z=-0.03.298 ψ=1.2131
10	$a = 2^{-x}\sqrt{x + \sqrt[4]{\left\|y\right\|}}\sqrt[3]{e^{x - 1/\sin z}}.$
	При x=0.03981, y=-1625, z=0.512 *a*=1.26185.
11	$b = y^{\sqrt[3]{\left\|x\right\|}} + \cos^3(y)\frac{\left\|x - y\right\|\left(1 + \frac{\sin^2 z}{\sqrt{x + y}}\right)}{e^{\left\|x-y\right\|} + \frac{x}{2}}.$
	При x=6.251, y=0.827, z=25.001 *b*=0.7121
12	$c = 2^{\left(y^x\right)} + \left(3^x\right)^y - \frac{y\left(arctg(z) - \frac{\pi}{6}\right)}{\left\|x\right\| + \frac{1}{y^2 + 1}}.$
	При x=3.251, y=0.325, z=0.0000466 *c*=4.2514

№	Функция и проверочные данные
13	$f = \dfrac{\sqrt[4]{y+\sqrt[3]{x-1}}}{\lvert x-y\rvert\left(\sin^2 z + tgz\right)}.$
	При x=17.421, y=0.010365, z=82800 *f*=0.33056
14	$g = \dfrac{y^{x+1}}{\sqrt[3]{\lvert y-2\rvert}+3} + \dfrac{x+\dfrac{y}{2}}{2\lvert x+y\rvert}(x+1)^{-1/\sin z}.$
	При x=1.23, y=15.4, z=252 *g*=82.8257.
15	$h = \dfrac{x^{y+1}+e^{y-1}}{1+x\lvert y-tgz\rvert}\left(1+\lvert y-x\rvert\right) + \dfrac{\lvert y-x\rvert^2}{2} - \dfrac{\lvert y-x\rvert^3}{3}$
	При x=2.444, y=0.00869, z=-130 h = -0.49871

Контрольные вопросы

1. Из каких разделов состоит программа на языке Паскаль?
2. Что такое оператор?
3. Какие операторы языка Паскаль вам известны?
4. Зачем нужен оператор присваивания? Какой вид он имеет?
5. Что может быть записано в правой части оператора присваивания?
6. Что такое переменная?
7. Что такое константа?
8. Какие правила применяются для создания имен переменных?
9. Что такое идентификатор?
10. Почему знак умножения всегда выписывают явно (например, пишут a*t, а не at)?
11. Как описываются переменные в Паскале?
12. Какие стандартные числовые типы языка Паскаль вам известны?
13. Что вам известно о соответствии типов переменных в языке Паскаль?
14. Какие арифметические операции можно выполнять в Паскале?
15. Что вам известно о приоритете арифметических действий в Паскале?
16. Какие математические функции есть в Паскале?
17. Какая команда служит для ввода данных?
18. Какой формат записи имеет команда ввода?
19. Чем команда ввода READ отличается от READLN?
20. Какая команда служит для вывода данных?
21. Какой формат записи имеет команда вывода?
22. Чем команда вывода WRITE отличается от WRITELN?
23. Как распечатать текст?
24. Что такое форматный вывод?

25. Как напечатать значение переменной на фиксированном количестве позиций?
26. Как напечатать значение переменной с фиксированным количеством знаков после запятой?
27. Как разместить комментарии в программе?

Лабораторная работа № 4

Тема. Реализация разветвляющихся вычислительных процессов. Оператор IF

Цель работы: научиться создавать программы для алгоритмов с ветвлением, в программе использовать оператор IF

Теоретическая часть

При разработке вычислительных алгоритмов часто возникает необходимость выбора направления дальнейшего решения задачи в зависимости от некоторого заданного условия. Алгоритмы такого типа называются разветвляющимися. В языке **Паскаль** их можно реализовать с помощью условных операторов.

Условные операторы обеспечивают выполнение некоторого оператора или группы операторов в зависимости от заданных условий. Для программирования разветвляющихся алгоритмов используются условные операторы **IF** или **CASE**.

Оператор IF

Оператор условного перехода IF в программах представляется в одном из двух форматов:

IF *логическое выражение* **THEN** *1 блок инструкций*
ELSE *2 блок инструкций;*

или

IF *логическое выражение* **THEN** *1 блок инструкций;* .

Здесь **IF, THEN и ELSE** - ключевые слова; *логическое выражение* - оператор сравнения или сложное логическое выражение; *блок инструкций* (1 и 2) – любые исполняемые операторы языка или блоки операторов, заключенные в операторные скобки **(BEGIN ... END).**

При выполнении условного оператора **IF** вначале анализируется результат логического выражения и в зависимости от его значения управление передается одному из операторов (блоку операторов), следующему за ключевым словом **THEN** или **ELSE**. Если значение результата **TRUE** «истина», то выполняется оператор (блок операторов), следующий за ключевым словом **THEN**. В противном случае (значение результата **FALSE** «ложь») выполняется оператор (блок операторов), следующий за ключевым словом **ELSE**. Далее в любом случае выполняется оператор, следующий в программе непосредственно за оператором **IF**. Ниже

приведен фрагмент программы, поясняющий действие рассматриваемого оператора:

IF A <= B **THEN** D := 2*D **ELSE** D := ABS(D);

В укороченном операторе **IF** отсутствует ключевое слово **ELSE** и блок инструкций, следующий за ним. Действия такой конструкции аналогичны рассмотренной выше. Если значение результата логического выражения **TRUE,** то выполняется оператор (блок операторов), следующий за ключевым словом **THEN.** При значении результата **FALSE** блок инструкций – 1 не выполняется, а сразу же выполняется блок инструкций, непосредственно следующий за оператором **IF**. Пример использования усеченного логического оператора **IF** :

IF A <= B THEN BEGIN D := 2*D; GOTO 10 END;

В последнем фрагменте пришлось после **THEN** вводить не один оператор, а так называемый ***составной оператор*** (блок операторов), который заключен в операторные скобки **BEGIN...END;** Формат такого составного оператора имеет вид:

BEGIN
1 оператор;
2 оператор;
N оператор
END;

В операторе условного перехода любой из операторов блоков инструкций, в свою очередь, может быть оператором условного перехода, образуя так называемую вложенную конструкцию операторов **IF**. Создавая подобную конструкцию, следует руководствоваться принятым в Паскале соглашением о том, что инструкция **ELSE** всегда относится к ближайшему предшествующему **IF**. С учетом этого правила вложенная конструкция **IF** должна иметь следующий вид:

IF *1 логическое выражение* **THEN** *1 блок инструкций*
ELSE IF 2 *логическое выражение*
THEN *2 блок инструкций*
ELSE 3 *блок инструкций.*

Пример 4.1. Определить значение функции ***У(х).***

$$y = \begin{cases} \sqrt{|\, x + 35 + e^{x-1} \,|}, & \textit{если} \quad x < 0{,}25 \\ x + \sin^2(2{,}6x), & \textit{если} \quad 0{,}25 \le x \end{cases}$$

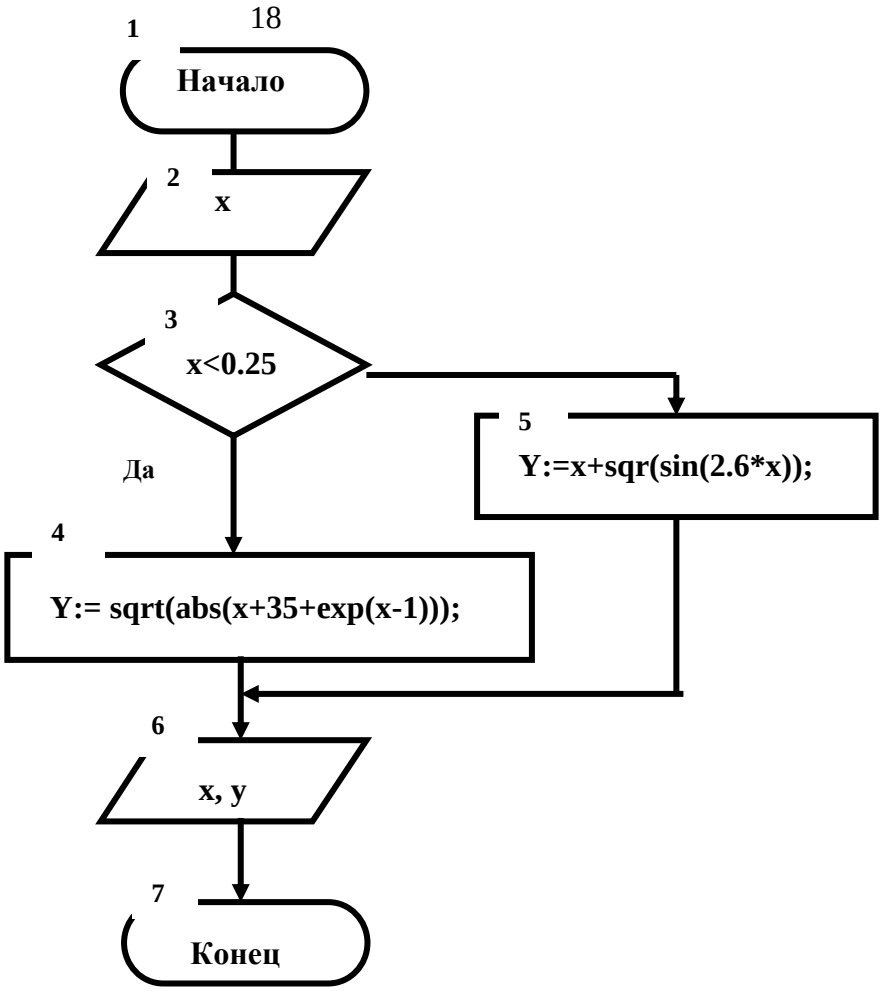


Рисунок 2

```
Program Project1;                                  //заголовок программы
var x,y: real;                                     //раздел описания переменных
begin                                               //начало раздела операторов
writeln('Введите значение x');                     //вывод текста на экран
write('x=');
readln(x);                                          // ввод значения x
if x < 0.25 then y := sqrt(abs(x+35+exp(x-1)))
                    else  y := x + sqr(sin(2.6*x));
writeln('при x = ',x : 6 : 2 , '    y = ', y : 6 : 2); //вывод результата
readln;                                             //задержка экрана
end.
```

Пример 4.2. Рассчитать значение функции ***У(х),*** значение ***х*** вводится с клавиатуры. При вводе значения ***х,*** для которого функция не определена, должно выводиться сообщение «Функция не определена».

$$y=\begin{cases} tg(2x+4{,}2)-2x, & x<1 \\ \sin x+\sqrt{6x}, & 2\le x\le 5 \\ 3{,}56x+\dfrac{2+x}{1+\sqrt{x}}, & x>7 \end{cases}$$

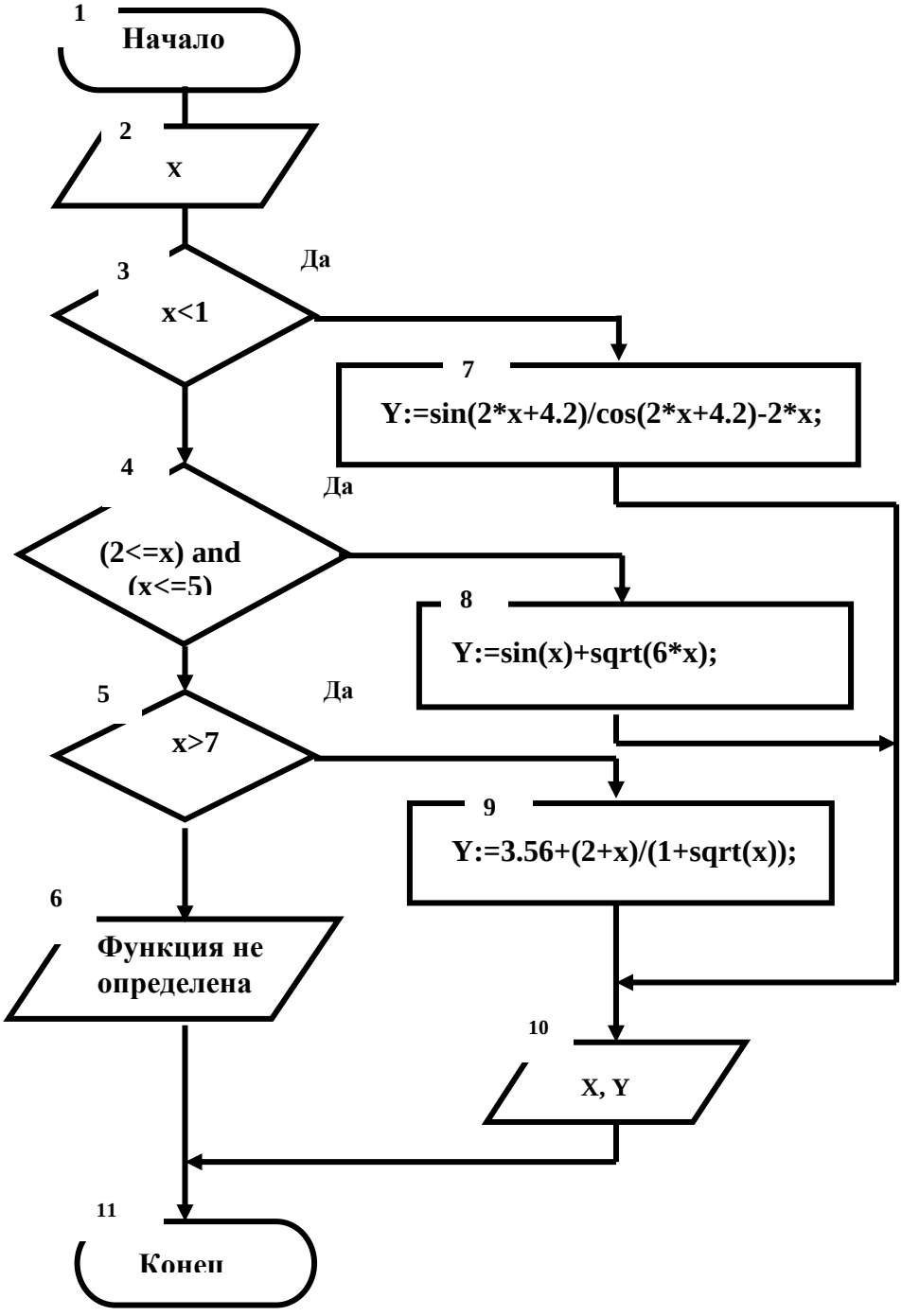


Рисунок 3

```
Program Project2;                          //заголовок программы
var x,y: real;                             //раздел описания переменных
begin                                      //начало раздела операторов
writeln('Введите значение x');             //вывод текста на экран
write('x=');
readln(x);                                 // ввод значения x
            //1 логическое условие
if x < 1 then y := sin(2*x+4.2)/cos(2*x+4.2)-2*x    else
                        //2 логическое условие
      if (2<=x) and (x<=5) then y := sin(x)+sqrt(6*x)  else
            //3 логическое условие
            if x>7 then y:= 3.56+(2+x)/(1+sqrt(x))   else
                  begin writeln('При  x=', x, 'функция не определена');
                        readln;
                  exit;
                  end;
writeln('при x = ',x : 6: 3, '    y = ', y : 6 : 2);          //вывод результата
readln;                                                      //задержка экрана
end.
```

Варианты заданий из табл. 6 (2 задания).

Рассчитать значение функции ***У(х).*** Значение аргумента вводится с клавиатуры. При вводе значения ***х***, для которого функция не определена, должно выводиться сообщение «Функция не определена».

Таблица 6

1.1	$y=\begin{cases}\sin(x+2), & x<1{,}35\\ x+3{,}5tgx, & 1{,}35\le x\end{cases}$	**1.2**	$y=\begin{cases}\sin(x+2), & x<1{,}35\\ x+3{,}5tgx, & 2\le x\le 4\\ \sqrt{\lvert 2{,}56x-0{,}35\rvert}, & x>5\end{cases}$
2.1	$y=\begin{cases}tg^2(x+2{,}5)+e^x, & x<0{,}25\\ 3{,}5\sin x+\cos^2 x, & 0{,}25\le x\end{cases}$	**2.2**	$y=\begin{cases}-\dfrac{tg(\lvert x\rvert^{2x})+x^2}{\cos(x+2)}, & x<-5\\ 12+5x, & 0\le x<5\\ \sqrt{\lvert x^{2-x}-x^2\rvert}*tg\,2x, & x\ge 10\end{cases}$
3.1	$y=\begin{cases}3x+5, & x\le 1\\ \sqrt{7x-6{,}35}, & 1\le x\end{cases}$	**3.2**	$y=\begin{cases}5{,}6(1+tgx), & x<0{,}5\pi\\ \sin x+6, & \pi\le x\le 3\pi\\ 2{,}56\sqrt{x^{2x}+1}, & x>5\pi\end{cases}$

4.1	$y=\begin{cases}3^{x+3,5}+tg(2x), & x<1,45\\ \sin^2 x+e^x, & 1,45\le x\end{cases}$	**4.2**	$y=\begin{cases}3e^{\sin x}*tg\,2x, & x\le 2\\ 2.47\lg x+x^{2x}, & 3<x<4\\ \sqrt{\lvert\cos^2 x\rvert+6}+4, & x\ge 6\end{cases}$
5.1	$y=\begin{cases}\cos x+3, & x<2,5\\ x+1,35, & 2,5\le x\end{cases}$	**5.2**	$y=\begin{cases}6(\cos^2 x-\sin^2 x), & x<-4\\ \sqrt{3,2x^2+tg^2 x}, & -1\le x<2\\ e*\cos(2,58x), & x\ge 5\end{cases}$
6.1	$y=\begin{cases}\log_3(x+2), & x\ge 1\\ \dfrac{x}{\sin(x-6)}, & 1>x\end{cases}$	**6.2**	$y=\begin{cases}tg(\lvert 2x+4,2\rvert)-\lg\lvert x\rvert, & x<1\\ \sin x+\sqrt{6x}, & 2\le x\le 5\\ 6+arctg\left(\dfrac{2x}{1+\sqrt{x}}\right), & x>7\end{cases}$
7.1	$y=\begin{cases}tg(x+3)-\dfrac{1}{x}, & x<2,5\\ x^2+6, & 2,5<x\end{cases}$	**7.2**	$y=\begin{cases}2+x^{2+x}, & x<0,1\pi\\ \sin^2(x^2+0,5), & \pi\le x\le 1,5\pi\\ \sin^2(2,45\sqrt{x-1,2}), & x>2,5\pi\end{cases}$
8.1	$y=\begin{cases}x^3+2, & 0\quad x\le 2,5\\ \sqrt[3]{x+3,5}+\sin x^2, & 2,5<x\end{cases}$	**8.2**	$y=\begin{cases}\sqrt{1+2,4x^2}, & x<1\\ \lg 5,9x, & 2\le x\le 3\\ \sqrt{x^2+1}+2x, & x>5\end{cases}$
9.1	$y=\begin{cases}x^2+3x\sin(x+6), & x\le 1,3\\ \sqrt{\lvert x-10,5\rvert+tg^2 x}, & 1,3<x\end{cases}$	**9.2**	$y=\begin{cases}e^{\lvert\sin(x)\rvert}tg(2,3+x), & x<-0,1\pi\\ arctgx, & -1\le x\le 0,5\\ -1+\sqrt{23,5x}, & x>1,5\pi\end{cases}$
10.1	$y=\begin{cases}x^2+5,35, & 4>x\\ \dfrac{\sqrt{3x-6,6}}{tgx}, & 4\le x\end{cases}$	**10.2**	$y=\begin{cases}\dfrac{e}{tg(2+x)}, & x<1\\ \ln(\lvert(x+2)^2-x^2\rvert), & 2\le x\le 4\\ \sin^2(x-6), & x>5\end{cases}$

11.1	$y=\begin{cases}2^{x+3}+\sin x, & x>2\\ \sin^2(x-6), & x\le 2\end{cases}$	**11.2**	$y=\begin{cases}\ln(\lvert 5{,}3x^{3x}-x^2\rvert), & x<1\\ \dfrac{0{,}025}{tg(2{,}6+x)}, & 2\le x\le 4\\ \sin^2(x-6), & x>5\end{cases}$
12.1	$y=\begin{cases}\sin(x+1), & x<3{,}5\\ 2{,}5x+3, & 3{,}5\le x\end{cases}$	**12.2**	$y=\begin{cases}\dfrac{e}{tg(2{,}9+3x)}, & x<1\\ 0{,}6\ln(\lvert 5-x^2\rvert), & 2\le x\le 4\\ \sin 3x+\lg(x+0{,}3), & x>6\end{cases}$
13.1	$y=\begin{cases}x^2+8, & 6>x\\ \dfrac{\sqrt{3x-3{,}6}}{tgx+\lg(x+2)}, & 6\le x\end{cases}$	**13.2**	$y=\begin{cases}\dfrac{2{,}3x+3{,}56}{tg(x+1)}, & x<2\\ \sin(5{,}4x^2-x^{2-x}), & 3\le x\le 4\\ tg(x+3)-\dfrac{1}{x}, & x>8\end{cases}$
14.1	$y=\begin{cases}24x^{2-x}, & 3>x\\ \dfrac{\ln x}{tgx}x^2+5{,}35, & 3\le x\end{cases}$	**14.2**	$y=\begin{cases}6{,}25+7x, & x<2\\ 5\ln(\lvert 2-x^2\rvert), & 6\le x\le 8\\ \dfrac{25x}{tgx}, & x>10\end{cases}$
15.1	$y=\begin{cases}x^2+6\sin(x+6), & x\le 4\\ \sqrt{\lvert x-10{,}5\rvert+tg^2x}, & 4<x\end{cases}$	**15.2**	$y=\begin{cases}\dfrac{2{,}56x+2}{tg(2+x)}, & x<2\\ \ln(x^2-1), & 2\le x\le 4\\ tg(2{,}78x+2), & x>10\end{cases}$

Контрольные вопросы

1. Что такое алгоритм с ветвлением?
2. Как записывается условный оператор (оператор ветвления) в Паскале?
3. Что такое полная и сокращенная записи условного оператора?
4. Что используется в качестве условий в операторе ветвления?
5. Какие знаки отношений можно использовать при составлении условий?
6. Что такое составное условие?
7. Каковы правила записи составных условий?

8. Какие вы знаете логические операции?
9. Какие действуют правила старшинства логических операций?
10. Что располагается после служебных слов **THEN** и **ELSE?**
11. Что такое составной оператор? Какую структуру он имеет?
12. В каких случаях используется составной оператор?

Лабораторная работа № 5

Тема. Программирование вычислительных процессов с ветвлением. Оператор CASE

Цель работы: научиться создавать программы для алгоритмов с ветвлением, в программе использовать оператор CASE

Оператор выбора CASE

Инструкция **Case** используется для выбора одного из нескольких направлений дальнейшего хода программы (последовательности инструкций, которые могут быть выполнены). Эта структура обеспечивает выбор одного из ряда возможных действий, в зависимости от значения, которое принимает переменная-селектор (условие). Выбор варианта действия осуществляется во время выполнения программы в зависимости от равенства значения переменной-селектора константе, указанной перед группой инструкций. В качестве переменной-селектора можно использовать переменную скалярного типа (integer или char). Конструкция имеет следующий формат:

CASE переменная-селектор **OF**
1 константа: 1 оператор;
2 константа: 2 оператор;
- - - - - - - - - - - - - - - - - - - -
N константа: N оператор
ELSE
оператор;
END;

Эта конструкция имеет и укороченный формат:
CASE переменная-селектор **OF**
1 константа: 1 оператор;
2 константа: 2 оператор;
- -
K константа: K оператор
END;

Пример 5.1. Вводится число от 0 до 21. Вывести сообщение: число меньше 10, число больше 10, число равно 10. При вводе числа больше 20 или меньше 0 должно выводиться сообщение «Ошибка ввода».

```
Program Project1;                                   //заголовок программы
var a: integer;                                    //раздел описания переменных
begin                                               //начало раздела операторов
write('Введите число a=');                        //вывод текста на экран
readln(a);                                          //ввод значения a
  case a of
        0..9: writeln(‘число меньше 10’);
          11..20: writeln(‘число больше 10’);
          10: writeln(‘число равно 10’);
        else
        writeln(‘Ошибка ввода’)
   end;
readln;
end.
```

Варианты заданий

При некорректном воде исходных данных программа должна выводить сообщение «Ошибка ввода».

1. Вводится число от 1 до 4, определяющее пору года. Дать название этой поры года (1 — зима, 2 — весна, 3 — лето, 4 — осень).
2. Вводится число от 1 до 7, определяющее день недели. Дать название этого дня (1 — понедельник, 2 — вторник, ..., 7 — воскресенье).
3. Вводятся числа 12, 1, 2, определяющие зимний месяц года. Дать название этого месяца года (1 — январь, 2 — февраль, 12 — декабрь).
4. Вводится число от 1 до 10. Дать название этого числа (1 — один, 2 — два,…, 10 — десять).
5. Дано натуральное число A (A<20), определяющее сумму денег в рублях. Дать для этого числа наименование: "рубль", "рубля", "рублей".
6. Дано натуральное число N (N<100), определяющее возраст человека в годах. Дать для этого числа наименование: "год", "года", "лет".
7. Вводится число от 1 до 12, определяющее месяц года. Дать название этого месяца года (1 — январь, 2 — февраль, ..., 12 — декабрь).
8. Вводится число от 1 до 20. Дать название этого числа (1 — один, 2 — два, ..., 20 — двадцать).
9. Вводятся числа 3, 4, 5, определяющие весенний месяц года. Дать название этого месяца года (3 — март, 4 — апрель, 5 — май).
10. Вводится число от 1 до 10. Дать название этого числа (1 — один, 2 — два, ..., 10 — десять).
11. Вводится число от 2 до 10. Вывести сообщение: четное или нечетное введенное число.
12. Вводится число от – 10 до 10. Вывести сообщение: введенное число больше 0, меньше 0 или равно 0.

13. Написать программу, которая запрашивает у пользователя номер дня недели и выводит одно из сообщений: «Рабочий день», «Суббота», «Воскресенье».
14. Вводится номер месяца (1, 2, …, 12). Вывести количество дней в указанном месяце.
15. Вводится значение года в укороченной форме (от 0 по 10). Вывести значение года текущего столетия в полном формате (0 – 2000, 1 – 2001 и т.д.).

Контрольные вопросы

1. Как в Паскале записывается оператор выбора?
2. Для чего предназначен оператор выбора?
3. Что такое полная и сокращенная записи оператора выбора?
4. Какого типа должна быть переменная, значения которой выбирают с помощью оператора **CASE**?

Лабораторные работы № 6 – 7

Тема. Вычисление сумм (произведений) конечного числа элементов ряда. Оператор FOR …DO (FOR ... DOWNTO)

Цель работы: научиться разрабатывать блок-схему алгоритма и программу с использованием операторов цикла FOR …DO на примере вычисления сумм (произведений) элементов конечного ряда

Теоретическая часть

Оператор FOR …DO

Циклические вычисления в Паскале реализуются с помощью операторов **FOR...TO(DOWNTO)...DO, WHILE...DO** или **REPEAT...UNTIL**.

Особенностью операторов **FOR...TO(DOWNTO)...DO** является встроенный внутренний счетчик цикла, фиксирующий количество повторений. Форматы записи оператора:

FOR ***переменная цикла := выражение 1*** **TO** ***выражение 2*** **DO**
{операторы тела цикла};

или

FOR ***переменная цикла*** **:=** ***выражение 1*** **DOWNTO** ***выражение 2*** **DO**
{операторы тела цикла};

В формате: ***переменная цикла*** (управляющая переменная, счетчик цикла) – имя переменной целочисленного типа; ***выражения*** 1 и 2 — выражения, тип результата которых совпадает с типом переменной цикла; ***операторы тела цикла*** — любой исполнимый оператор или блок операторов, заключенных в операторные скобки; **FOR, TO, DOWNTO** и **DO** – ключевые слова.

Использование оператора **FOR** регламентируется следующими правилами.

- Переменная цикла, её начальное и конечное значения должны быть

одного типа.

- При выполнении оператора очередное значение параметра цикла вычисляется автоматически. В частности, для целого типа шаг изменения значения переменной цикла равен 1 при конструкции с **TO** и -1 при конструкции с **DOWNTO**.
- После служебного слова **DO** может стоять только один оператор. Если в цикле необходимо выполнить группу операторов, то их заключают в операторные скобки **BEGIN**...**END**, образуя составной оператор.
- Цикл не выполняется вообще, если начальное значение переменной цикла больше (при **DOWNTO** меньше), чем конечное. В этом случае управление передается оператору, следующему непосредственно за конструкцией **FOR**..**TO**(**DOWNTO**)...**DO**.
- В конструкции **FOR**..**TO**(**DOWNTO**)...**DO** используется только одна переменная цикла. При наличии вложенных циклов у каждой конструкции **FOR**..**TO**(**DOWNTO**)...**DO** должна быть своя, отличающаяся от других переменная.
- При организации вложенных циклов внутренний и внешний циклы не должны пересекаться. Цикл, который начинается последним, должен завершаться первым:

```
FOR N := 1 TO 5 DO
BEGIN WRITELN (N:2,'.');
FOR J := 1 TO 5 DO WRITELN (N:2,'.', J:2,'.') END;
```

Пример работы циклов

Пример 6.1. Найти сумму конечного ряда. Значение x вводится с клавиатуры.

$$S = \sum_{i=1}^{10} x * i$$

```
program Project1;
var  x,s:real;  i:integer;
begin
Write('x='); Readln(x);
s:=0;
for i:=1 to 10 do                    //начало цикла For
 s:=s+x*i;              //конец цикла For
Writeln('s=',s:5:3);    //вывод результата
Readln;
end.
```

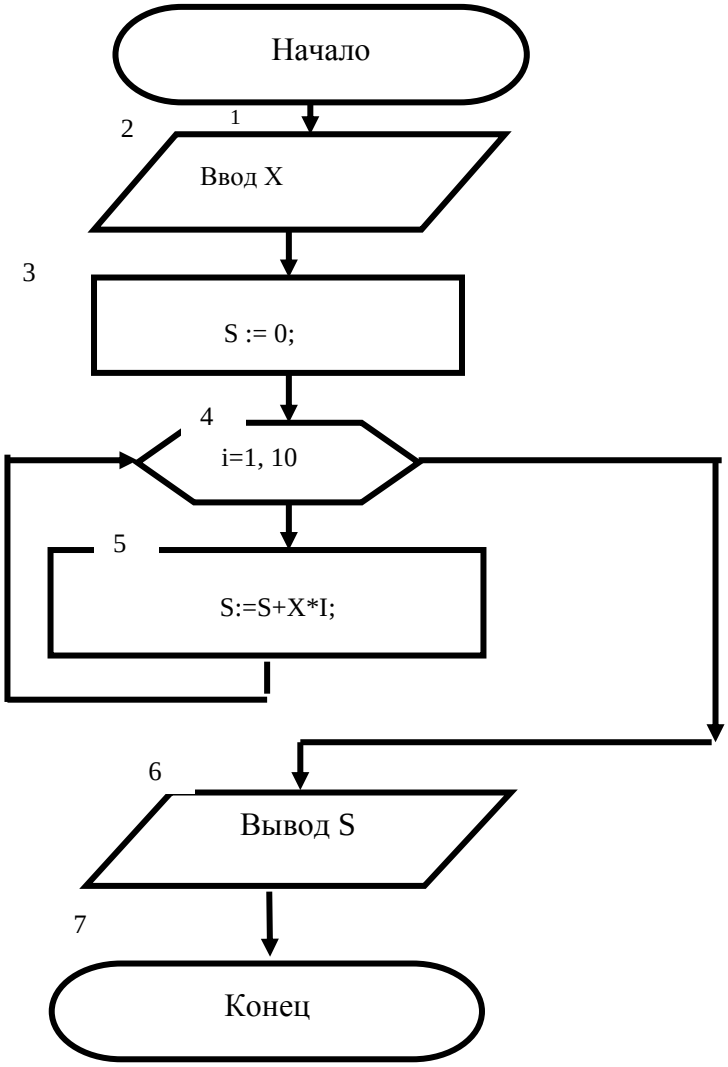


Рисунок 4

Варианты заданий в табл. 7, 8.

Вычислить значение суммы или произведения членов ряда. Значение ***x*** задать с клавиатуры.

Таблица 7 – Задание 1

1	$P=\prod_{k=2}^{10}\frac{x+\ln x}{2xk}$	**9**	$S=\sum_{n=3}^{8}\frac{x*\sin nx}{2n+1}$
2	$S=\sum_{n=1}^{8}\frac{x*n}{2n}$	**10**	$P=\prod_{n=1}^{4}\frac{n+2+\lg nx}{2n}$
3	$P=\prod_{n=1}^{4}\frac{n+\cos nx}{e^n}$	**11**	$P=\prod_{n=3}^{8}\frac{n+\cos(nx+5)}{2n}$

4	$S=\sum_{n=3}^{8}\frac{x\ *\sin(8x)}{2n+1}$	**12**	$P=\prod_{n=1}^{4}\frac{2n+1+\lg nx}{2n}$
5	$S=\sum_{n=2}^{6}\frac{x\ *\sin nx}{6+n}$	**13**	$P=\prod_{n=1}^{6}\frac{n+3+\cos 2x}{2n-1}$
6	$P=\prod_{n=3}^{8}\frac{2n+2+\lg nx}{n+3+x}$	**14**	$P=\prod_{n=3}^{9}\frac{n+2+nx}{2n+2}$
7	$S=\sum_{n=3}^{8}\frac{x\ *\sin(x+5)}{2n+3}$	**15**	$P=\prod_{n=1}^{4}\frac{2n+3+\lg nx}{2n+4}$
8	$S=\sum_{n=1}^{8}\frac{x^2*n}{2+n}$	**16**	$P=\prod_{n=3}^{7}\frac{n+3+\lg nx}{2+n}$

Пример 6.2. Дано действительное число ***x***. Вычислить сумму элементов ряда. Число суммируемых элементов ряда (***N***) задается во время работы программы с клавиатуры.

$$S=\sum_{i=1}^{n}(-1)^{i+1}\frac{x^{2i-1}}{(2i-1)!}.$$

```
Program Project2; //вычисление суммы заданного числа элементов ряда
var
   x,s,a:real;
   f:longint;
   i,n,k:integer;
begin                                          //начало раздела операторов
write('введите значение x=');
readln(x);
write('введите значение n=');
readln(n);
//значение первого слагаемого ряда вычисляется до цикла
a:=x;                                          //значение x в степени (2*1-1)
f:=1;                                          //значение f=(2*1-1)!=1
s:=a;                                          //a - значение 1-го слагаемого, s=a
//к - переменная для изменения знака очередного слагаемого с + на -
k:=1;
for i:=2 to n do                               //начало цикла
begin
k:=- k;                                        //значение (-1) для i-го слагаемого
a:=a * sqr(x);              //вычисление x в степени (2i+1)   i-го слагаемого
```

```
f:=f * (2*i-2)*(2*i-1);          //вычисление факториала i-го слагаемого
s:=s+k*a/f;                      //добавление i-го слагаемого к общей сумме
end;                             //конец цикла
                  //вывод результата на экран
writeln('при x=',x:4:2,'  сумма', n,' элементов =',s:7:5);
readln;
end.
```

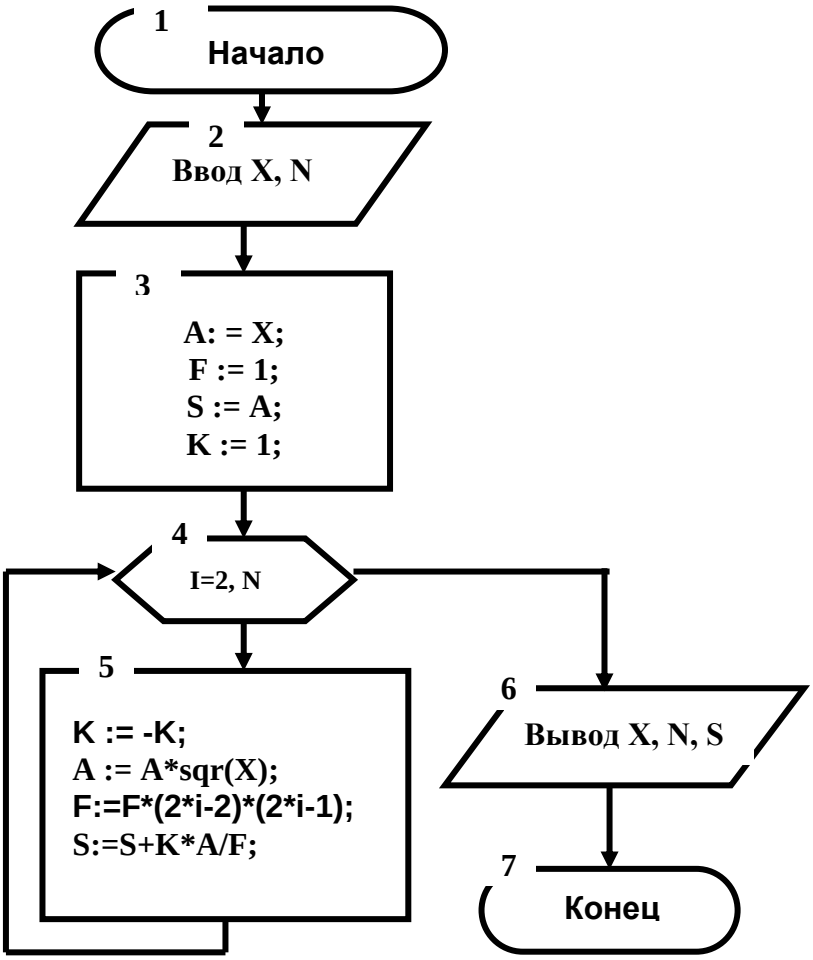


Рисунок 5

Пример 6.3. Задано число ***x*** и натуральное число ***N***. Вычислить произведение ***N*** элементов ряда

$$P = \left(2 + \frac{1}{x}\right) * \left(2 + \frac{1*2}{x^2}\right) * \left(2 + \frac{1*2*3}{x^3}\right) * \ldots * \left(2 + \frac{1*2*\ldots*N}{x^N}\right).$$

Для вычисления произведения целесообразно использовать рекуррентные соотношения, т.е. каждую дробь в скобках, начиная со второй, выражать через предыдущую. Это позволит существенно сократить объем

вычислений. Расчет N! в числителе дроби каждого множителя выполняется в программе в теле цикла по формуле **f:=f*n. p:=p*(2+f/a)** - формула умножения элементов ряда $\mathbf{P=P_1*P_2*\ldots*P_{i-1}}$ на очередной $\mathbf{P_i}$ элемент ряда.

Схема алгоритма и текст программы

```
Program Project3;
//вычисление произведения заданного числа элементов ряда

var       x,p,a : real;     f : longint;     i,n : integer;
begin
write('введите x=');
readln(x);
write('введите n=');
readln(n);
p:=1;                                   //начальное значение произведения
a:=1;  f:=1;              //начальные значение числителя и знаменателя дроби
for i:=1 to n do                        //начало цикла
begin
a:= a*x;
f:=f*i
p:=p*(2+f/a);
end;                          //конец цикла
                              //вывод результата на экран
writeln('при x=',x:4:2,'  произведение', n:2,' элементов=',p:7:5);
readln;
end.
```

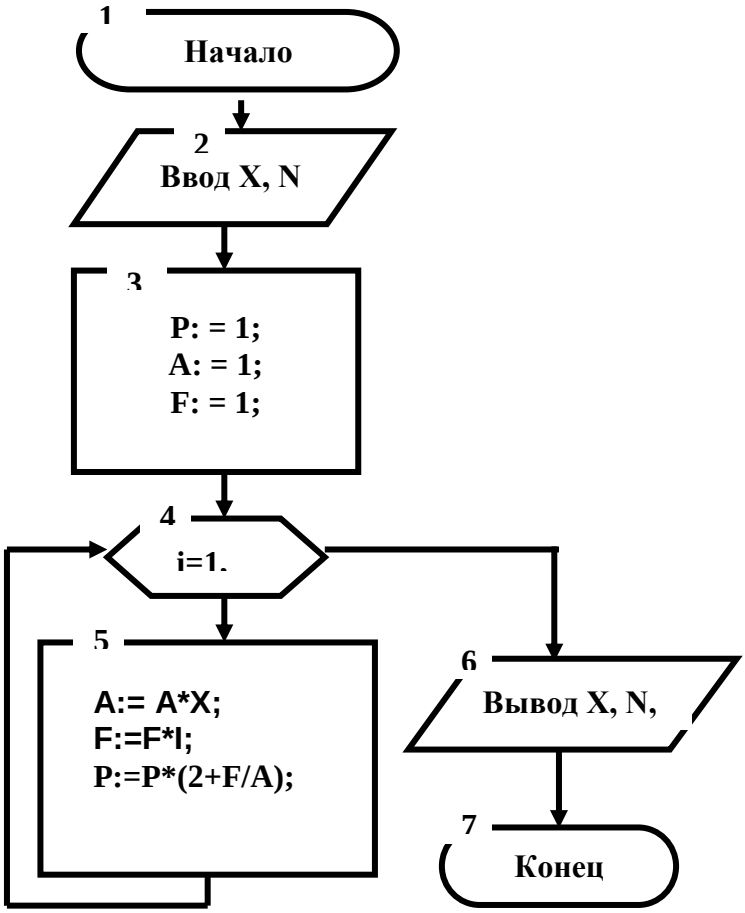


Рисунок 6

Таблица 8 – Задание 2

1	$P=\prod_{k=2}^{10}(-1)^{k+1}\frac{x^{2k-1}+\ln x}{2xk*(k+2)!}$	**9**	$S=\sum_{n=3}^{8}(-1)^{n}*\frac{x^{2n}*\sin nx}{(2n+1)!}$
2	$S=\sum_{n=1}^{8}(-1)^{n+1}\frac{x^{n+2}*(n+4)!}{2n}$	**10**	$P=\prod_{n=1}^{4}\frac{n+x^{n+3}+\lg nx}{(2n)!}$
3	$P=\prod_{n=1}^{4}(-1)^{n+1}\frac{(n!+\cos nx)}{e^{n}}$	**11**	$P=\prod_{n=3}^{8}\frac{(n+2)!+nx}{(2n)!}$
4	$S=\sum_{n=3}^{8}(-1)^{n}*\frac{x^{2n}*\sin nx}{(2n+1)!}$	**12**	$P=\prod_{n=1}^{4}\frac{(2n+1)!+\lg nx}{(2n)!}$
5	$S=\sum_{n=2}^{6}(-1)^{n+1}\frac{x^{2n}*\sin nx^{n+2}}{(6+n)!}$	**13**	$P=\prod_{n=1}^{6}\frac{(n+3)!+8x^{2+n}}{(2n-1)!}$

6	$P=\prod_{n=3}^{8}\frac{(2n+2)!+\lg nx}{x^{n+4}}$	**14**	$P=\prod_{n=2}^{9}\frac{(n+3)!+nx^{n+1}}{(2n+1)}$
7	$S=\sum_{n=3}^{8}(-1)^n*\frac{x^{2n}*\sin nx}{(2n+1)!}$	**15**	$P=\prod_{n=3}^{9}\frac{(n+1)!+5x^{n+4}}{(n+4)}$
8	$S=\sum_{n=3}^{8}(-1)^n*\frac{x^{2n}*\sin nx}{(2n+1)!}$	**16**	$P=\prod_{n=1}^{6}\frac{8x+6}{(3n+1)!}$

Контрольные вопросы

1. Для чего предназначен оператор цикла?
2. Какие виды циклов есть в Паскале?
3. Какой формат записи имеет оператор **FOR***?* Какие существуют варианты этого цикла?
4. Как работает оператор **FOR***?* В каких случаях применяется?
5. Что является телом цикла?
6. Как в теле цикла выполнить несколько операторов?
7. Почему перед выполнением цикла некоторым переменным нужно задавать начальные значения?
8. При каких условиях оператор for не выполнится ни разу?

Лабораторная работа № 8

Тема. Расчет значений функции на заданном промежутке (табулирование функции). Оператор WHILE … DO

Цель работы: научиться разрабатывать схемы и программы с использованием оператора цикла WHILE … DO

Теоретическая часть

Для организации итерационных циклов, когда заранее не известно число повторений, применяют операторы **WHILE...DO и REPEAT...UNTIL.** Запись оператора **WHILE...DO** начинается с ключевого слова **WHILE,** за которым следует условие выхода из цикла, ключевое слово DO и тело цикла:

WHILE ***логическое условие*** **DO**
{операторы тела цикла};

В формате: ***логическое условие*** - оператор отношения (логическое выражение), определяющий условие завершения цикла; ***операторы тела цикла*** — любой исполнимый оператор или блок операторов, заключенных в операторные скобки.

Операторы, входящие в тело цикла, выполняются до тех пор, пока при проверке условия выхода из цикла получаем результат **TRUE** («истина»). Как только оно принимает значение **FALSE** («ложь»), управление передается оператору, следующему в программе за конструкцией **WHILE...DO**. Если условие имеет значение **FALSE** с самого начала выполнения оператора, то операторы тела цикла не выполняются ни разу. Оператор **WHILE...DO** реализует алгоритм циклической структуры с предусловием, так как условие выхода из цикла вычисляется и анализируется перед каждым выполнением тела цикла.

Пример 8.1. Разработать схему и программу табулирования функции, заданной на отрезке **[a, b], h –** шаг приращения аргумента ***x***

$$y = \begin{cases} 12d + x + \cos x^3, & x < 2 \\ \ln(|10 - x^2|), & 2 \le x \le 4 \\ 4 + tgx^2, & x > 4 \end{cases}$$

a = 1, b = 6, h = 0,5. Значение **d** задать с клавиатуры.

Схема алгоритма и программа решения задачи

```
Program Project1;
var a,b,h,d,x,y:real;
begin
writeln('введите границы интервала [a,b],шаг h-');
readln(a,b,h);                    //после ввода каждого значения нажать ENTER
write('введите значение d=');
readln(d);
//вывод заголовка таблицы значений функции на экран
writeln('таблица значений функции');
writeln('   x  ', '         y');
x:=a;                             //начальное значение x
while x <= b do
        begin
//вычисление y в зависимости от условия
       if x<2 then y:=12*d+x+cos(x*sqr(x))
       else     if x<=4 then y:=ln(abs(10 – sqr(x)))
                else y:= 4 + sin(sqr(x))/cos(sqr(x));
//вывод значений аргумента и функции на экран
 writeln (x:5:2,' ':5,y:5:2);
x:=x+h;                           //приращение аргумента на h=0,5
end;
readln;
end.
```

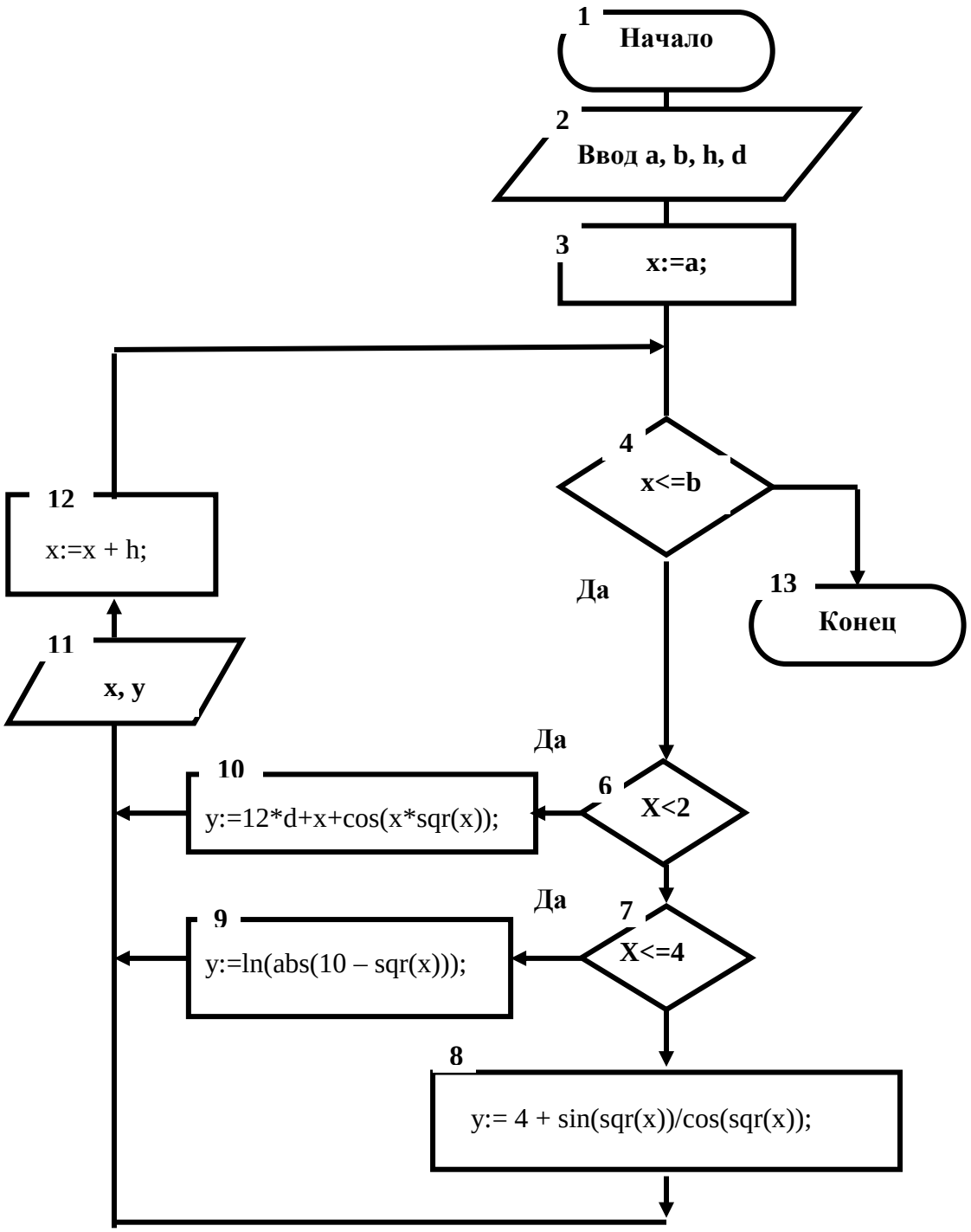


Рисунок 7

Варианты заданий в табл. 9.

Составить блок-схему и программу вычисления значений функции ***Y(x),*** где **a <= *x* <= b, h** – шаг изменения ***x***. Значения **c, d** задать самостоятельно как константы.

Таблица 9

1	$y=\begin{cases}\dfrac{e}{\lg(d+x)}, & x<2\\ \ln(\lvert 10-x^2\rvert), & 2\le x\le 4\\ 4+tgd^2, & x>4\end{cases}$ a=1, b=6, h=0,5	**9**	$y=\begin{cases}-\dfrac{\lg(c^2+x^2+4)}{\cos 2d}, & x<d\\ 12+\ln x, & d\le x<(d+4{,}5)\\ \sqrt{\lvert c^2-x^2\rvert}, & x\ge(d+4{,}5)\end{cases}$ a=-3, b=5, h=0,5
2	$y=\begin{cases}c*(1+\lg x), & x<0{,}5\pi\\ \sin x+5x, & 0{,}5\pi\le x\le 1{,}5\pi\\ c*\sqrt{x^2+1}, & x>1{,}5\pi\end{cases}$ a=1, b=4π, h=$0{,}2\pi$	**10**	$y=\begin{cases}c*e^{\sin x}, & x\le 3\\ c*\lg x, & 3<x<4\\ \sqrt{a*\cos^2 x}, & x\ge 4\end{cases}$ a=1, b=5, h=0,25
3	$y=\begin{cases}c*(\cos^2 x-\sin^2 2x), & x<-1\\ \sqrt{3c^2+x^2}, & -1\le x<2\\ tg^2 x+\cos(c*x), & x\ge 2\end{cases}$ a=-2, b=3, h=0,5	**11**	$y=\begin{cases}d-\lg(2*x), & x<1\\ \sin x+c, & 1\le x\le 1{,}5\\ c+arctg\left(\dfrac{x}{1+\sqrt{x}}\right), & x>1{,}5\end{cases}$ a=0,5, b=2, h=0,2
4	$y=\begin{cases}d*\ln(\lvert 10-x^2\rvert), & x<2\\ \dfrac{tg(3+x)}{2x+2}, & 2\le x\le 4\\ 4+d^2, & x>4\end{cases}$ a=3, b=9, h=0,5	**12**	$y=\begin{cases}\dfrac{2*e}{x+d}, & x<2\\ d*\cos(x^2-2), & 2\le x\le 4\\ 4+d^2, & x>4\end{cases}$ a=2, b=10, h=0,5
5	$y=\begin{cases}\sqrt{\ln(1+c)}, & x<2\\ \lg x+2c, & 2\le x\le 3\\ \sqrt{c^2+1}, & x>3\end{cases}$ a=1, b=5, h=0,5	**13**	$y=\begin{cases}e^x*arctg(b+x), & x<0{,}2\pi\\ ctgx+3, & 0{,}2\pi\le x\le 1{,}5\pi\\ -1+\lg x, & x>1{,}5\pi\end{cases}$ a=1, b=5, h=0,25
6	$y=\begin{cases}\dfrac{e}{tg(d+x)}, & x<2\\ d*\ln(\lvert 10-x^2\rvert), & 2\le x\le 4\\ 4+d^2, & x>4\end{cases}$ a=1, b=6, h=0,5	**14**	$y=\begin{cases}1+2\sin 2x, & x<0{,}5\pi\\ \sin x, & 0{,}5\pi\le x\le \pi\\ \lg(4x+1), & \pi<x\le 1{,}5\pi\\ -1, & x>1{,}5\pi\end{cases}$ a=0, b=6, h=0,5

7	$y=\begin{cases}4+d^2, & x<2\\ d*\ln(\lvert 0{,}47-x^2\rvert), & 2\le x\le 4\\ \dfrac{4}{tg(d+x)}, & x>4\end{cases}$ a=2, b=8, h=0,5	**15**	$y=\begin{cases}\dfrac{e}{tg(d+x)}, & x<2\\ d*\ln(\lvert 10-x^2\rvert), & 2\le x\le 4\\ 3x+d^2+23, & x>4\end{cases}$ a=1, b=6, h=0,5
8	$y=\begin{cases}2+\lg(4x), & x<0{,}5\pi\\ \sin^2(x^2), & 0{,}5\pi\le x\le 1{,}5\pi\\ \sin^2(4*\sqrt{x}), & x>1{,}5\pi\end{cases}$ a=0,5, b=2π, h=0,1π.	**16**	$y=\begin{cases}tgx*e^{\sin x}, & x\le 3\\ c*\lg x, & 3<x<4\\ \sqrt{a*(\lg 10x)^2 x}, & x\ge 4\end{cases}$ a=1, b=5, h=0,25

Контрольные вопросы

1. Для чего предназначен оператор цикла?
2. Какие виды циклов есть в Паскале?
3. Какой формат записи имеет оператор **WHILE**? Как он работает? В каких случаях применяется?
4. Чем отличается оператор **WHILE** от оператора **REPEAT**?
5. Что является телом цикла?
6. Как в теле цикла выполнить несколько операторов?
7. Почему перед выполнением цикла некоторым переменным нужно задавать начальные значения?
8. Что такое зацикливание? Как его избегать?

Лабораторная работа № 9

Тема. Вычисление сумм (произведений) элементов бесконечного ряда с заданной точностью. Оператор Repeat …Until

Цель работы: научиться разрабатывать схемы и программы с использованием оператора цикла Repeat … Until на примере вычисления суммы (произведения) элементов бесконечного ряда с заданной точностью

Теоретическая часть

Алгоритм циклической структуры с *постусловием* реализуется оператором **REPEAT...UNTIL,** структура которого имеет вид:

REPEAT
1 *оператор тела цикла;*
2 *оператор тела цикла*;
N *оператор тела цикла*

UNTIL *логическое условие****;***

Здесь: ***логическое условие*** — оператор отношения (логическое выражение), определяющий условие завершения цикла; ***операторы тела цикла*** – любой исполнимый оператор или группа операторов.

В операторе **REPEAT...UNTIL** проверка условия производится после очередного выполнения операторов тела цикла, следовательно, в нем операторы тела цикла выполняются как минимум один раз.

Операторы тела цикла выполняются до тех пор, пока значение логического условия есть **FALSE** («ложь»). Как только оно принимает значение **TRUE** («истина»), управление передается оператору, следующему в программе за конструкцией **REPEAT UNTIL.**

Операторы цикла в **REPEAT...UNTIL** не заключаются в операторные скобки, а отделяются друг от друга точкой с запятой.

Применение операторов **WHILE...DO и REPEAT...UNTIL** регламентируется некоторыми правилами.

- Перед каждым выполнением цикла (в том числе и первым) операнды, входящие в логическое условие, должны иметь конкретные значения и однозначно определять условие окончания или продолжения цикла.
- Тело цикла должно содержать хотя бы один оператор, влияющий на операнды логического условия (изменяющий переменную, которая анализируется в условии), что исключает «зацикливание» при выполнении и бесконечное продолжение цикла.
- Изменение значения переменной, влияющей на логическое условие, через некоторое конечное число операций должно привести к удовлетворению условия окончания цикла.

Пример 9.1. Разработать блок-схему и программу вычисления суммы элементов ряда с заданной точностью e = 0,00001. Условием окончания вычислений считать $|Sn| <= e$. Значение x задать самостоятельно.

$$S = \sum_{n=1}^{\infty} (-1)^n * \frac{x^{2n+1} * \sin nx}{(2n+3)!}$$

Блок-схема и программа задачи.

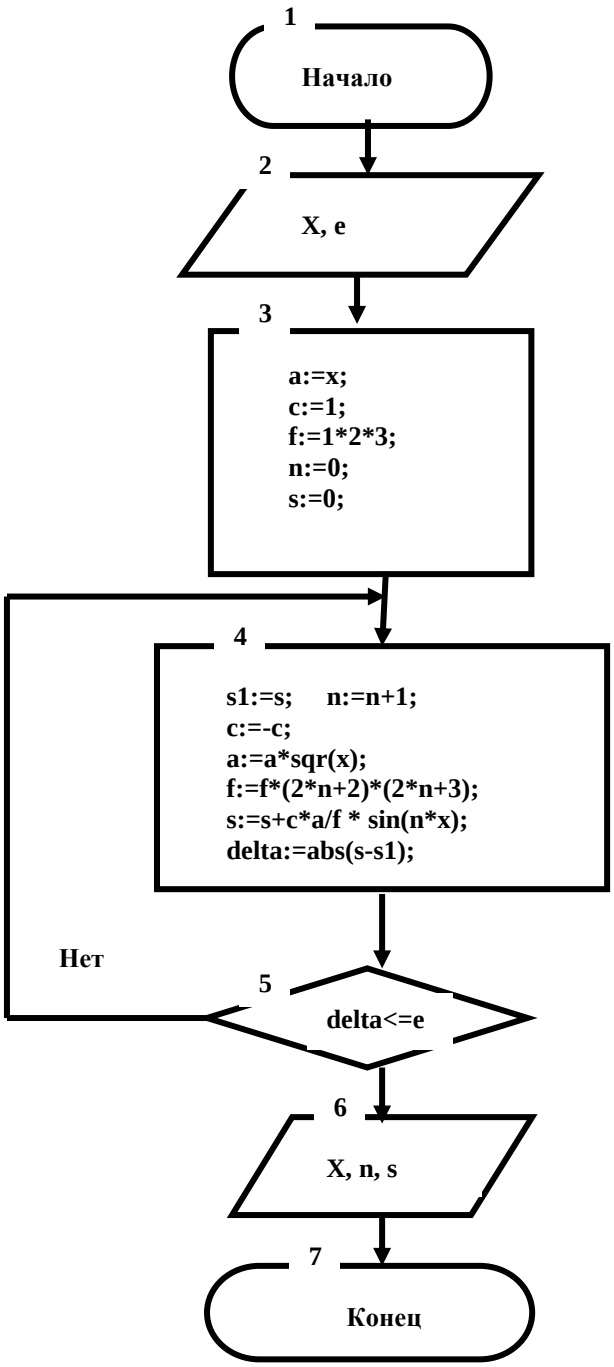


Рисунок 8

```
Program Project1;
var
x,e,a,s,s1,delta:real;
n,c:integer;
f:longint;
begin                                   //начало раздела операторов
```

```
write('введите x=');
readln(x);
write('введите точность вычислений e=');
readln(e);
    //начальные значения a, f, c, n, s
a:=x;   c:=1;
f:=1*2*3;
n:=0;
s:=0;
repeat                                  //начало суммирования
       s1:=s;                           //в s1 хранится значение s для сравнения
       n:=n+1;                          //счетчик слагаемых
       c:=-c;                             //значение (-1) для n-го слагаемого
       a:=a*sqr(x);                   //вычисление x в степени (2n+1)   n-го слагаемого
       f:=f*(2*n+2)*(2*n+3);          // факториала n-го слагаемого
       s:=s+c*a/f * sin(n*x);         //добавление n-го слагаемого к общей сумме
       delta:=abs(s-s1);                //проверка условия окончания цикла
until delta<=e;
       //вывод результата на экран
writeln('при x=',x:4:2,'  сумма', n:2,' элементов',s:7:5);
readln;
end.
```

Особенностью итерационного цикла является то, что заранее невозможно определить число необходимых итераций или приближений. Поэтому при программировании циклов в таких задачах применяется не цикл со счетчиком (или с параметром), а цикл с постусловием.

Программа выполняет расчет каждого элемента ряда по общим формулам. Для уменьшения количества вычислений при расчете $\mathbf{x^{2n+1}}$ и факториала (2n+3)! используются их значения из предыдущих слагаемых (переменные a, f). Для правильного расчета первого элемента ряда задаются начальные условия. Им нужно уделить особое внимание.

Программа снабжена комментариями и дополнительных объяснений не требует.

Варианты заданий в табл. 10

Вычислить сумму элементов бесконечного ряда с заданной точностью e = 0,00001. Условием окончания вычислений считать | Sn | <= e.
Значение x задать самостоятельно.

Таблица 10

1	$S=\sum_{n=3}^{\infty}(-1)^n * \frac{\sin^{n+3}(2n+x)}{(2n-1)!}$	**9**	$S=\sum_{n=1}^{\infty}\frac{(0{,}3x+5)^{2n-1}}{(2n+3)!}$

2	$S=\sum_{n=1}^{\infty}(-1)^{n+1}*\frac{x^{2+n}\sin n}{(3n+2)!}$	**10**	$S=\sum_{n=1}^{\infty}(-1)^{n+2}*\frac{x^{n+2}}{(2n)!}*\sin nx$
3	$S=\sum_{n=1}^{\infty}(-1)^{n}*\frac{02x^{2n-1}}{\cos(2n+1)}$	**11**	$*\frac{1}{n!}$ $S=\sum_{n=1}^{\infty}(-1)^{2n+1}\frac{(x+2)^{2*n}\sin 2nx}{(2n+3)!}$
4	$S=\sum_{n=1}^{\infty}(-1)^{n+1}\frac{6+(0,2x)^{3+n}}{(2n+2)!}$	**12**	$S=\sum_{n=1}^{\infty}(-1)^{n}\frac{6+(0,3x)^{3+n}}{(2n+2)!}$
5	$S=\sum_{n=1}^{\infty}(-1)^{n}\frac{(0,1x+6)^{2n-1}}{(2n+1)!}$	**13**	$S=\sum_{n=1}^{\infty}(-1)^{n}*\frac{x^{2n+1}}{n!*(2n+2)!}$
6	$S=\sum_{n=1}^{\infty}(-1)^{n+1}\frac{4+(0,6x)^{3+n}}{(2n+2)!}$	**14**	$S=\sum_{n=1}^{\infty}(-1)^{n+1}*\frac{\sin 2nx+n}{(2n+2)!+(x+1)^{2+n}}$
7	$S=\sum_{n=1}^{\infty}(-1)^{n+1}*\frac{2nx+16+x^{4+n}}{(n+3)!}$	**15**	$S=\sum_{n=1}^{\infty}(-1)^{n+1}\frac{n-(3-x)^{3+n}}{(2n+3)!}$
8	$S=\sum_{n=1}^{\infty}(-1)^{n}*\frac{x^{2n+1}\sin 2nx+2n}{(n+3)!}$	**16**	$S=\sum_{n=1}^{\infty}(-1)^{n}*\frac{x^{n+3}}{(2n+3)!*(4n+3)}$

Контрольные вопросы

1. Для чего предназначен оператор цикла?
2. Какие виды циклов есть в Паскале?
3. Какой формат записи имеет оператор **WHILEL**? Как он работает? В каких случаях применяется?
4. Какой формат записи имеет оператор **REPEAT**? Как он работает? В каких случаях применяется?
5. Чем отличается оператор **WHILE** от оператора **REPEAT**?
6. Какой формат записи имеет оператор **FOR**? Какие существуют варианты этого цикла?
7. Как работает оператор **FOR**? В каких случаях применяется?
8. Что является телом цикла?
9. Как в теле цикла выполнить несколько операторов?
10. Почему перед выполнением цикла некоторым переменным нужно задавать начальные значения?
11. Что такое зацикливание? Как его избегать?

Лабораторные работы № 10 – 11

Тема. Процедуры и функции пользователя в языке Паскаль

Цель работы: приобретение навыков написания программ с подпрограммами пользователя на языке Паскаль

Теоретическая часть

В языке Паскаль имеется два вида подпрограмм – процедуры и функции.

Имея один и тот же смысл и аналогичную структуру, процедуры и функции различаются назначением и способом их использования.

Все процедуры в языке Паскаль делятся на 2 вида: встроенные и пользовательские.

Встроенные определены заранее и могут вызываться без предварительного описания.

Пользовательские – именованная группа операторов, реализующая определенную последовательность действий.

Вызов пользовательских процедур и функций осуществляется по имени.

Описание процедуры:

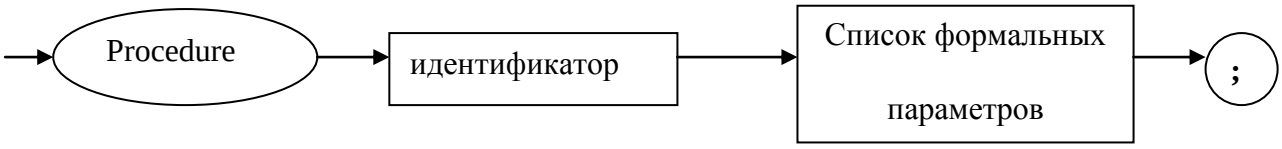


Рисунок 9 - Синтаксическая диаграмма описания процедуры

Описание функции:

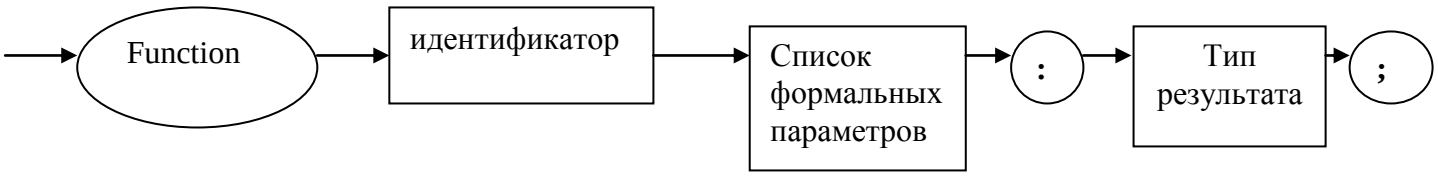


Рисунок 10 - Синтаксическая диаграмма описания функции

Пример 10.1. Рассчитать значение y(x) по формуле

$$y = tg^2(x+1) + tg(x+0.5) + 7*tg(x)$$

Для расчета значения tg(x) используется **подпрограмма-функция** tg(a) с одним параметром.

```
Program Project1;
  var x,y:real;
function tg(a:real):real;
begin
tg:=cos(a)/sin(a);
end;
```

```
begin
   write('x='); readln(x);
   y:=sqr(tg(x+1))+tg(x+0.5)+7*tg(x);
   writeln('y=',y:6:2);
   readln;
end.
```

Пример 10.2. Условие из предыдущего примера. В программе для расчета значения tg(x) используется **подпрограмма-процедура** tg(a,b) с двумя параметрами.

```
Program Project2;
  var x,y:real; x1,x2,x3:real;

procedure tg(a:real; var b:real);
begin
b:=cos(a)/sin(a);
end;

begin
   write('x='); readln(x);
   tg(x+1,x1);    tg(x+0.5,x2);   tg(x,x3);
   y:=sqr(x1)+x2+7*x3;
   writeln('y=',y:6:2);
   readln;
end.
```

Пример 10.3. Рассчитать периметр (в м) и площадь прямоугольника (в $м^2$). Длины сторон (x, y) вводятся с клавиатуры в см. В программе используется **подпрограмма-процедура** parametr(a, b, p, s).

```
Program Project3;
  var x,y:integer; s1,p1:real;

 procedure    parametr(a,b:integer; var p,s:real);
 begin
 p:=2*(a+b)/100;
 s:=a*b/10000;
 end;

begin
x:=30; y:=40;
```

```
parametr(x,y,p1,s1);
writeln('p1=',p1:6:2, '  s1=',s1:6:2);
readln;
end.
```

Пример 10.4. Условие из предыдущего примера. В программе используются две подпрограммы-функции.

```
Program Project4;
 var x,y:integer; s1,p1:real;

 function    p(a,b:integer):real;
 begin
 p:=2*(a+b)/100;
  end;

  function    s(a,b:integer):real;
 begin
  s:=a*b/10000;
 end;

begin
x:=30; y:=40;
p1:=p(x,y);
s1:=s(x,y);
writeln('p1=',p1:6:2, '  s1=',s1:6:2);
readln;
end.
```

Пример 10.5. Условие из предыдущего примера. В программе используется процедура parametr без параметров.

```
Program Project5;
  var a,b:integer; p,s:real;
  procedure parametr;
  begin
  p:=2*(a+b)/100;
  s:=(a*b)/10000;
  end;
begin
write('a='); readln(a);  write('b='); readln(b);
 parametr;
 writeln('p=',p:6:2, '  s=',s:6:2);
 a:=50; b:=80;
```

```
  parametr;
 writeln('p=',p:6:2, '  s=',s:6:2);
readln;
 end.
```

Пример 10.6. Условие из предыдущего примера. В программе используются две функции (s и p) без параметров.

```
Program Project6;
  var a,b:integer; p1,s1:real;
 function    p:real;
 begin
 p:=2*(a+b)/100;
  end;

  function    s:real;
 begin
  s:=a*b/10000;
 end;

begin
a:=30; b:=40;
p1:=p;
s1:=s;
writeln('p1=',p1:6:2, '  s1=',s1:6:2);
readln;
end.
```

Варианты заданий (2 задания)

Задание 1. Написать программу с подпрограммой-функцией.
Задание 2. Написать программу с подпрограммой-процедурой (с параметром или параметрами).

1. Даны действительные s и t. Рассчитать f(t,-2s,1.17)+f(2.2,t,s-t), где $f(a,b,c)=\dfrac{2a-b-\sin(c)}{5+|c|}$ подпрограмма.
2. Даны действительные a, b, c. Получить $\dfrac{\max(a-b,a,a+b)+\max(a,b+c,a-c)}{1+\max(a+bc,1.15,a/c)}$, где max(x,y) подпрограмма.
3. Даны действительные числа S и t. Рассчитать f(t,-3*s,2.5-t)+f(5.2,3*t,s-t), где

$$f(a,b,c)=\frac{2a-b-\cos(c)}{3.5-|c|}$$ подпрограмма.

4. Даны действительные числа S и t. Рассчитать
f(1.5*t,2*s)+f(t,3-s)-f(2*s-3,t+s), где

$$f(a,b)=\frac{a^3-b^2+\sin(ab)}{2ab+5b^2}$$ подпрограмма.

5. Даны действительные числа S и t. Рассчитать
h(s,t)+h(h2 (s-t,s*t),h4 (s-t,s+t))+h(1,1), где

$$h(a,b)=\frac{a}{1+b^2}+\frac{b}{1+a^2}-(a-b)^2$$ подпрограммы.

6. Даны действительные числа a, b. Рассчитать Y=min(a,b), Y1=min(min(a*b,a+b), min(a2 –b*a,b3-3*a)), Y2=min(Y12+Y,5.25),
где min(x,x1) подпрограмма.

7. Даны действительные числа S и t. Рассчитать
f(t,4*s,2.5*t)+f(5,2*t,s+t),

$$f(a,b,c)=\frac{a^2+b^3+\sqrt{|c|}}{3*|c|}$$
где подпрограмма.

8. Даны действительные числа S и t. Рассчитать
h(s,t)+h^3 (s-t,s*t)+h^2 (s-t,s+t)+h(1,1), где

$$h(a,b)=\frac{a^2}{1+b}+\frac{b}{1+a^3}+(a-b)^3$$ подпрограмма.

9. Даны действительные числа a, b, c. Рассчитать

$$y=\frac{\max(a,a+b)+\max(a,b+c)}{1+\max(a+b*c,2*a,c*b)}$$, где max(x,x1) описать как подпрограмму.

10. Вычислить К = (x + y + z) / 3, где x – наибольшее значение из параметров x1, x2, y - из y1, y2, z – из z1, z2. Использовать подпрограмму для нахождения наибольшего значения из двух параметров.

11. В порт в среднем приходят 3 корабля в день. Какова вероятность того, что в день придет 2 корабля, 4 корабля? Вероятность вычислять по формуле
P = 3 * e - 3 / к!. Использовать подпрограмму для расчета к!.

12. Вычислить Z = (v1 + v2 + v3) / 3, где v1,v2,v3 – объемы шаров с радиусами r1, r2, r3 соответственно. Использовать подпрограмму для расчета объёмов шаров. Объем шара вычислять по формуле V = 4 / 3*Пi*R3.

13. Определить число сочетаний из n по m (n>m), по формуле C=n!/m!(n-m). Использовать одну подпрограмму для расчета n! и m!.

14. Вычислить Z = (n + m) / 2, где n – наименьшее значение из параметров n1, n2, m – из m1, m2. Использовать подпрограмму для нахождения наименьшего значения из двух параметров.

15. Составить программу вычисления значения функции

S = x2 + y2 + sin(2 * x2 * y2) + x + z + sin(2 * x * z) + y2 + z2 + sin(2 * y2 * z2) , используя подпрограмму для расчета a + b + sin(2 * a * b).

Лабораторные работы № 12 – 13

Тема. Обработка одномерных массивов

Цель работы: приобретение навыков работы с одномерными массивами

Теоретическая часть

Понятие массива

Массив, в отличие от простой переменной, представляет собой не одно значение, а ***множество значений, объединенных одним именем***. В языке Turbo Pascal все значения из этого множества должны иметь один и тот же тип.

Каждое из значений массива называется ***элементом массива.***

Доступ к элементам массива производится посредством указания имени массива и номера элемента массива, заключенного в квадратные скобки.

Номер элемента массива называется ***индексом элемента массива.***

Использование элемента массива не отличается от использования простой переменной, имеющей тот же тип, что и элемент массива.

В Turbo Pascal'е массив объявляется при помощи ключевого слова **array,** после которого в квадратных скобках указываются границы индексов – верхняя, а после двух точек нижняя. После квадратных скобок после ключевого слова **of** указывается тип элементов массива.

Пример определения массивов:

```
Var
  A: Array [1..10] of integer; //массив A, состоящий из 10 элементов
                               целого типа с индексами от 1 до 10
  B: Array [5..8] of real;     //массив B, состоящий из 4 элементов
                               вещественного типа с индексами от 5 до 8
```

Прежде, чем работать с массивом, его нужно описать. Описание массива в программе осуществляется двумя способами:

Вариант описания массива с использованием раздела TYPE

TYPE <имя типа>=array[тип индекса] of <тип компонент>;
VAR <имя массива1, имя массива2, …, имя массива n>:<имя типа>;

Пример:

```
TYPE MAS=array[1..20] of real;
VAR A,B,C: MAS;
```

Вариант описания массива в разделе VAR

```
VAR
```

<имя массива1,имя массива2,…,имя массиваn>: array [тип индекса] of <тип компонент>;

Пример: VAR A,B,C: array [1..20] of real;

Решение типового примера

Пример 12.1. Дан массив вещественных чисел x1,x2,…,xn. Найти максимальный элемент массива.

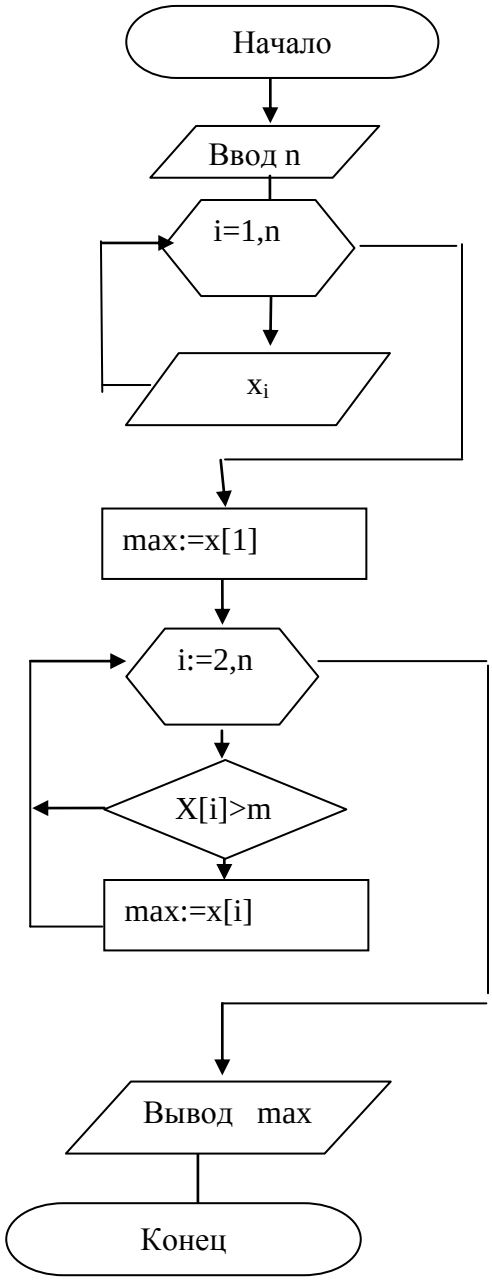


Рисунок 11

Программа

```
Program Project1;
Var max:real;
x: array[ 1..40] of real;
i,n:integer;
Begin

//ввод размера массива
Write('n=');
Readln(n);

//ввод значений элементов массива
for i:=1 to n do

read(x[i]);

max:=x[1];

//нахождение максимального
//элемента массива
for i:=2 to n do
if x[i]>max then max:=x[i];

writeln('max=',max:6:2);
readln;
end.
```

Пример 11.2. Элементам массива b присвоить значения элементов массива a.

```
program Project2;
```

```
const m=20;
type mas=array[1..m] of real;   // описание типа одномерного массива
var a,b:mas;                    // описание переменных одномерного массива,
                                // массивы a и b – идентичные массивы
        i:integer;
begin

for i:=1 to m do
a[i]:=0;                        // инициализация массива a

b:=a;                           // присваивание значений массива a массиву b

   for i:=1 to m do
     write(b[i]:3:1,'  ');   // вывод значений массива b

readln;
end.
```

Пример 12.3. Найти сумму элементов одномерного массива.

```
program Project3;
const m=10;
type m1=array[1..m] of real;
var a:m1;  i:integer;   s:real;
begin
s:=0;
for i:=1 to m do
  begin
  write('a[',i,']=');
  readln(a[i]);  // ввод значений массива a
  s:=s+a[i];
  end;
write('s=',s:3:1); // вывод результата
readln;
end.
```

Пример 12.4. Найти сумму и количество положительных и отрицательных элементов одномерного массива a.

```
program Project4;
const m=10;
type m1=array[1..m] of real;
```

```
var a:m1;  i,kp,ko:integer;   sp,so:real;
begin
sp:=0; kp:=0; so:=0; ko:=0;
for i:=1 to m do
  begin
  write('a[',i,']=');
  readln(a[i]);  // ввод значений массива a
   if a[i]>0 then begin sp:=sp+a[i]; kp:=kp+1; end;
   if a[i]<0 then begin so:=so+a[i]; ko:=ko+1; end;
  end;
 writeln('sp=',sp:3:1,' kp=',kp);   // вывод суммы и количества положительных
                                     //элементов
write('so=',so:3:1,' ko=',ko); // вывод суммы и количества отрицательных
                                  // элементов
readln;
end.
```

Пример 12.5. Найти значение максимального элемента одномерного массива a.

```
program Project5;
const m=10;
type m1=array[1..m] of real;
var a:m1;  i,kp,ko:integer;   max:real;
begin

for i:=1 to m do
  begin
  write('a[',i,']=');
  readln(a[i]);  // ввод значений массива a
  end;

max:=a[1];
for i:=1 to m do
   if a[i]>max then max:=a[i];
writeln('max=',max:3:1);   // вывод значения максимального элемента
одномерного массива
readln;
end.
```

Варианты заданий (2 задания)

Задание 1

1. Дан массив вещественных чисел **A(A_1,A_2,…,A_n)** и **B(B_1,B_2,…,B_n).** Уменьшить положительные элементы массива **A** на величину **K**, а положительные элементы массива **B** увеличить на величину **K**, где **K** – заданное число.
2. Дан массив вещественных чисел A(A1,A2,…,An). Заполнить массив B(B1,B2,…,Bn), элементы которого рассчитываются следующим образом:

$$Bi = \sum_{i=1}^{n} A_i .$$

3. Дан массив вещественных чисел X(X1,X2,…,Xn) . Найти сумму элементов массива X, удовлетворяющих условию G ≤Xn ≤ E, где E и G – заданные числа.
4. Составить программу для вычисления длины n-мерного вещественного вектора X = (X1, X2,…, Xn) по формуле L = $\sqrt{x_1^2 + x_2^2 + ...x_n^2}$
5. Дан массив вещественных чисел X(X1, X2,…, Xn) . Найти произведение положительных элементов массива и подсчитать их число.
6. Дан массив вещественных чисел A(A1,A2,…,An) и B(B1,B2,…,Bn). Найти скалярное произведение векторов A и B по формуле $S = \sum_{i=1}^{n} a_i * b_i$.
7. Дан массив вещественных чисел A(A1,A2,…,An). Найти среднее арифметическое положительных и среднее арифметическое отрицательных элементов массива.
8. Дан массив вещественных чисел B(B1,B2,…,Bn). Найти сумму положительных и сумму отрицательных элементов массива.
9. Дан массив вещественных чисел A(A1,A2,…,An) и B(B1,B2,…,Bn). Найти сумму отрицательных элементов массива A и B.
10. Дан массив вещественных чисел A(A1,A2,…,An) и C(C1,C2,…,Cn). Подсчитать число положительных элементов массива A и отрицательных элементов массива C.
11. Дано целое число N и набор из N ненулевых целых чисел. Вывести в том же порядке номера всех нечетных чисел из данного набора и количество K таких чисел.
12. Дан массив вещественных чисел из N элементов и целое число K. Если в массиве присутствует число, меньшее K, то вывести сообщение True; в противном случае вывести False.
13. Дан набор ненулевых целых чисел; признак его завершения — число 0. Вывести количество элементов в наборе.
14. Дан набор ненулевых целых чисел; признак его завершения — число 0. Вывести сумму всех положительных четных чисел из данного набора. Если требуемые числа в наборе отсутствуют, то вывести 0.
15. Дано целое число K и набор ненулевых целых чисел; признак его завершения — число 0. Вывести количество чисел в наборе, меньших K.

16. Дано целое число K и набор ненулевых целых чисел. Вывести номер последнего числа в наборе, меньшего K. Если таких чисел в наборе нет, то вывести 0.
17. Дано целое число N и набор из N целых чисел. Вывести номера тех чисел в наборе, которые меньше своего левого соседа, и количество K таких чисел.
18. Дано целое число N и набор из N целых чисел. Вывести номера тех чисел в наборе, которые больше своего правого соседа, и количество K таких чисел.
19. Дано целое число N и набор из N вещественных чисел. Проверить, образует ли данный набор возрастающую последовательность. Если образует, то вывести True, если нет — вывести False.
20. Дано целое число N и набор из N вещественных чисел. Если данный набор образует убывающую последовательность, то вывести 0; в противном случае вывести номер первого числа, нарушающего закономерность.
21. Дано целое число N и набор из N целых чисел, содержащий по крайней мере два нуля. Вывести сумму чисел из данного набора, расположенных между первыми двумя нулями (если первые нули идут подряд, то вывести 0).
22. Дано целое число N и набор из N целых чисел, содержащий по крайней мере два нуля. Вывести сумму чисел из данного набора, расположенных между последними двумя нулями (если последние нули идут подряд, то вывести 0).
23. Даны целые числа K, N и набор из N вещественных чисел: A_1, A_2, ..., A_n. Вывести K-е степени чисел из данного набора: A_1^K, A_2^K, ..., A_n^K.
24. Дано целое число n и набор из N вещественных чисел: A_1, A_2, ..., An. Вывести следующие числа: A_1, A_2^2, ..., A_{n-1}^{n-1}, A_n^n.
25. Дано целое число n и набор из n вещественных чисел: A_1, A_2, ..., An. Вывести следующие числа: A_1^n, A_2^{n-1}, ..., A_{n-1}^2, A_n.

Задание 2 (использовать подпрограмму)

1. Найти минимальный и максимальный из данных десяти элементов.
2. Найти номера максимального из данных десяти элементов.
3. Найти номера последнего максимального из данных десяти целочисленных элементов.
4. Найти количество минимальных из данных десяти целочисленных элементов.
5. Найти максимальный четный из данных десяти ненулевых целочисленных элементов. Если требуемые элементы отсутствуют, то вывести 0.
6. Найти минимальный положительный из данных десяти элементов. Если требуемые элементы отсутствуют, то вывести 0.
7. Даны числа a, b ($0 < a < b$) и набор из десяти элементов. Найти минимальный из элементов, содержащихся в интервале (a, b). Если требуемые элементы отсутствуют, то вывести –1.
8. Дан набор из десяти целочисленных элементов. Найти количество элементов, расположенных после первого минимального.
9. Найти номер последнего экстремального (то есть минимального или максимального) из данных десяти целочисленных элементов.

10. Дан набор из десяти целочисленных элементов. Найти количество элементов, содержащихся между первым и последним минимальным. Если в наборе имеется единственный минимальный элемент, то вывести 0.
11. Найти два наибольших из данных десяти элементов.
12. Дан набор из десяти целочисленных элементов. Найти максимальное количество подряд идущих минимальных элементов.
13. Дан массив размера N. Расположить значения его элементов в обратном порядке: значение a[n] поместить в a[1], a[n-1] в a[2] и т.д. Дополнительный массив не использовать.
14. Дан массив размера N. Вывести вначале его элементы с четными индексами, а затем — с нечетными.
15. Дан целочисленный массив A размера 10. Вывести номер первого из тех его элементов A[i], которые удовлетворяют двойному неравенству: A[1] < A[i] < A[10]. Если таких элементов нет, то вывести 0.
16. Дан целочисленный вектор T(m). Изменить знак всех элементов с четными индексами на противоположный.
17. Массив K(n) заполнен случайными числами от -15 до 15.Определить количество отрицательных элементов и их индексы.
18. В заданном массиве M(15) заменить нулевые элементы квадратами их индексов.
19. В массиве A(20) поменять местами соседние четные и нечетные по номеру элементы. Дополнительные массивы не использовать.
20. Задан массив A(n). Вычислить сумму произведений всех пар соседних чисел.

Контрольные вопросы

1. Что такое массив?
2. В каких случаях необходимо использовать массивы?
3. Что такое размерность массива?
4. Что такое размер массива?
5. Что такое элемент массива, индекс массива?
6. Какие типы данных могут использоваться в качестве индексов для массивов?
7. Как ввести массив чисел?

Лабораторные работы № 14 – 15

Тема. Работа с двумерными массивами

Цель работы: приобретение навыков работы с двумерными массивами

Теоретическая часть

Двумерный массив можно представить в виде прямоугольной таблицы, например:

2 3 4 5
0 4 8 3
7 1 5 3

Такой массив в программе определяется следующим образом:

Var A: array[1..3,1..4] of integer;

Здесь в массиве A первый интервал индексов обозначает индекс номера строки – 1..3, второй интервал индексов обозначает индекс номера столбца – 1..4.

Для обращения к элементу двумерного массива необходимо в квадратных скобках сначала указать номер строки, а затем номер столбца.

Например:

```
Writeln(A[2,3]); //будет выведено число 8
Writeln(A[3,1]); //будет выведено число 7
Writeln(A[1,1]); //будет выведено число 2
```

Аналогично одномерному массиву, двумерный также описывается двумя способами:

1. **В разделе TYPE**

TYPE <имя типа>=array[тип индекса] of <тип компонент>;
VAR <имя массива1,имя массива2,…,имя массива n>:<имя типа>;
Пример:

```
TYPE MAS1=array[1..5,1..6] of real;
VAR A,B,C: MAS1;
```

2. **В разделе VAR даётся полное описание массива:**

VAR <имя массива1,имя массива2,…,имя массиван>: array [тип индекса] of <тип компонент>;

Пример:

```
VAR A,B,C: array[1..5,1..6] of real; // массивы A, B, C – идентичные
```

Или

```
VAR
A: array[1..5,1..6] of real;
B: array[1..5,1..6] of real;
C: array[1..5,1..6] of real;  // массивы A, B, C – неидентичные
```

Решение типовой задачи

Пример 14.1. Дана матрица A(n*n), найти произведение элементов каждого столбца матрицы.

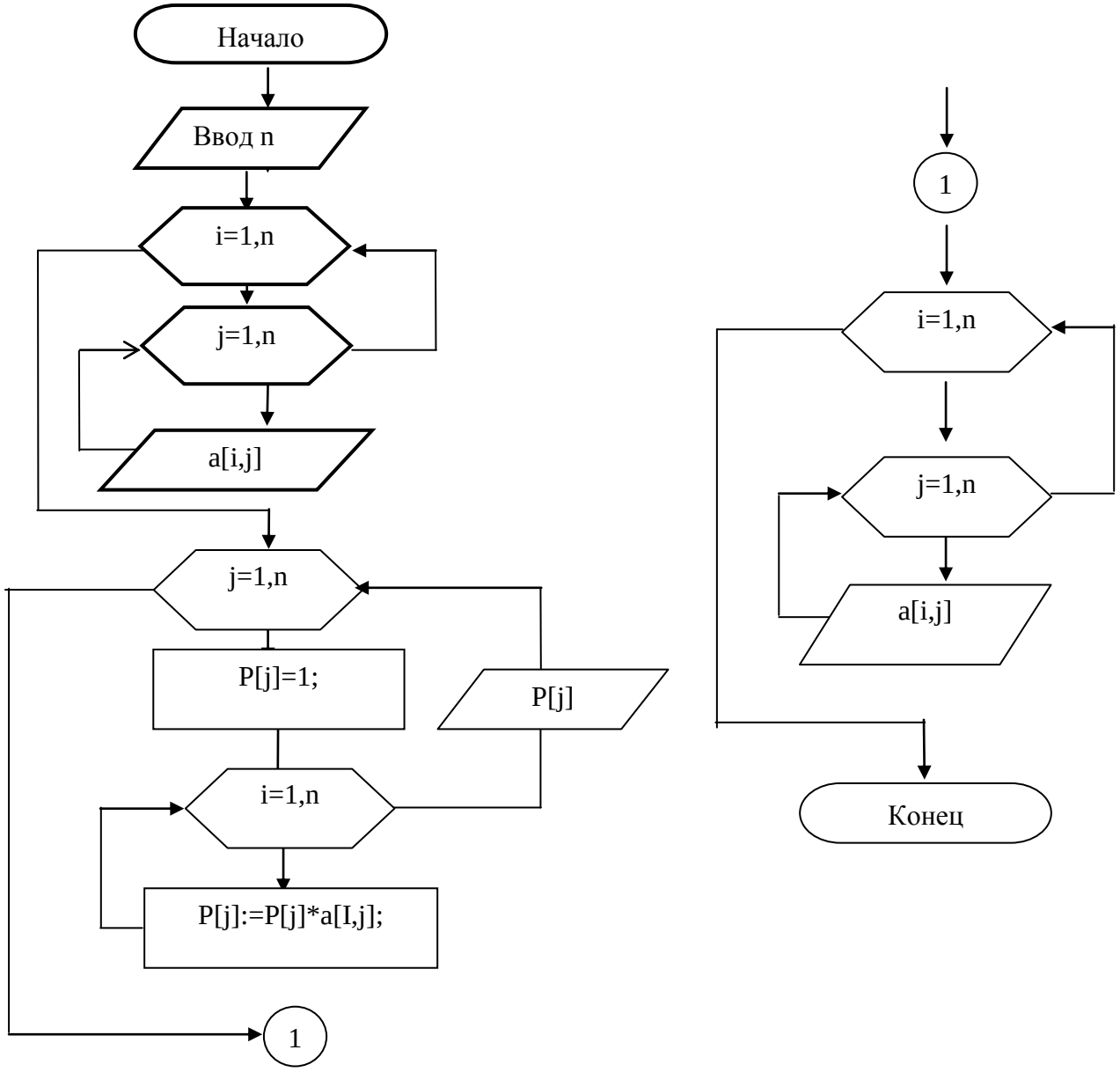


Рисунок 12 – Блок-схема алгоритма решения задачи

Программа:

```
Program Project1;
Type Mas=array [1..8] of real;
Mas1=array [1..8,1..8] of real;
Var
A: Mas1;
```

```
I,j,n:integer; P:Mas;
Begin

//ввод размера массива
Write('n=');
Readln(n);

//ввод значений элементов массива
//значения элементов каждой строки вводятся в одну строку через пробел
//в конце каждой строки нажимается клавиша Enter
For i:=1 to  n do
For j:=1 to n do
Read(a[I,j]);

//расчет произведений элементов каждого столбца
For j:=1 to  n do   //начало внешнего цикла
Begin
  P[j]:=1;
     For i:=1 to n do        //начало вложенного цикла
     P[j]:=P[j]*a[i,j];      //конец вложенного цикла

   Writeln('P[',j:1, ']= ',P[j]:6:2);
End;                                    //конец внешнего цикла

//вывод матрицы A
Writeln('Матрица A : ');

For i:=1 to  n do                //начало внешнего цикла
Begin
     For j:=1 to n do        //начало вложенного цикла
     Write(a[i,j]:5:1);      //конец вложенного цикла
   Writeln;
End;                         //конец внешнего цикла
End.
```

Пример 14.2. Поиск максимального значения и его индексов в массиве a.

```
program Project2;
const n= 3; m=4;
type m1=array[1..n,1..m] of real;
var a:m1;  i,j,n1,n2:integer;  max:real;
begin
for i:=1 to n do
```

```
    for j:=1 to m do
        read(a[i,j]);  // ввод значений массива a

max:=a[1,1];  n1:=1;  n2:=1;
for i:=1 to n do
    for j:=1 to m do
        if a[i,j]>max then begin
                          max:=a[i,j];  n1:=i;
                          n2:=j;
                          end;

writeln('a_max=',max:3:1);   // вывод значения максимального элемента
writeln('i=',n1,' j=',n2);   // вывод номера максимального элемента
 readln;
 readln;
end.
```

При исходных данных

```
1   2.5    5.2   8
7    8      9.3   6
9   15.5    6     9
```

Результат работы программы:

```
a_max=15.5
i=3 j=2
```

Пояснения к программе

Массив a – двумерный массив размерностью 3 x 4 (n*m). Значения n и m определены в разделе констант (CONST).

Переменные i, j используются для работы с элементами массива. I – индекс строки, j – индекс столбца.

Переменные n1,n2 используются для фиксирования индексов максимального элемента массива: n1 – индекс строки, n2 – индекс столбца.

Переменная max используется для хранения значения максимального элемента массива.

Варианты заданий (2 задания)

Задание 1

1. Дана вещественная матрица A, размерностью n*m. Найти наибольший элемент матрицы и номер строки и столбца, в которых он находится.
2. Дана вещественная квадратная матрица A, размерностью n*n. Найти сумму элементов матрицы, расположенных под главной диагональю.
3. Дана вещественная квадратная матрица A, размерностью n*n. Найти сумму элементов матрицы, расположенных над главной диагональю.

4. Дана вещественная матрица A, размерностью n*m. Найти в каждой строке матрицы минимальный элемент.
5. Дана вещественная матрица A, размерностью n*m. Найти в каждой строке матрицы максимальный элемент.
6. Дана вещественная матрица A, размерностью n*m. Найти в каждом столбце матрицы произведение положительных элементов.
7. Дана вещественная матрица A, размерностью n*m. Найти сумму и произведение элементов каждой строки матрицы.
8. Дана вещественная матрица A, размерностью n*m. Найти произведение квадратов отрицательных элементов.
9. Дана вещественная матрица A, размерностью n*n. Найти сумму элементов матрицы, расположенных на главной диагонали.
10. Дана вещественная матрица A, размерностью n*m. Найти сумму положительных элементов матрицы.
11. Дана квадратная матрица **A(4,4)**. За один просмотр элементов матрицы **A(4,4)** сформировать вектор **C(4),** каждый **j**-й элемент которого равен произведению элементов **j**-го столбца исходной матрицы.
12. Дана квадратная матрица **A(4,4)**. За один просмотр элементов матрицы **A(4,4)** сформировать вектор **D(4),** каждый j-й элемент которого равен сумме соответствующей строки матрицы A.
13. Даны две прямоугольные матрицы **A(3,4)** и **B(3,4).** Найти матрицу **C(3,4),** элементы которой равны сумме соответствующих элементов матриц **A** и **B.**
14. В данной целочисленной квадратной матрице A(5,5) указать индексы всех элементов, имеющих наибольшее значение.
15. Дана вещественная матрица A, размерностью n*m. Найти произведение квадратов положительных элементов.
16. Дана прямоугольная матрица **A(3,4).** Произвести транспонирование исходной матрицы в матрицу **C(4,3)**.
17. Дана вещественная матрица A, размерностью n*n. Найти сумму квадратов элементов матрицы, расположенных на главной диагонали.

Задание 2

1. Дано число **k (0<k<11)** и матрица размера **4 x 10**. Найти сумму и произведение элементов **k**-го столбца данной матрицы.
2. Дана матрица размера **N x M.** Найти суммы элементов всех ее четных строк.
3. Дана матрица размера **N x M**. Найти минимальное значение в каждой строке.
4. Дана матрица размера **N x M**. В каждой строке рассчитать среднее арифметическое каждой строки.
5. Дана матрица размера **N x M**. Преобразовать матрицу, поменяв местами минимальный и максимальный элемент.
6. Дана квадратная матрица порядка **M**. Найти сумму элементов ее главной диагонали.

7. Дана действительная квадратная матрица **M(5,5).** Требуется переставить 2 строки матрицы.
8. Дана действительная квадратная матрица **C(7,7).** Найти максимальный элемент в главной диагонали и напечатать элементы строки, в которой он находится.
9. Дана матрица M(7,4). Определить четные элементы, имеющие нечетную сумму индексов.
10. Дана квадратная целочисленная матрица **F(m,m).** Найти суммы элементов тех строк, имеющих четные элементы на главной диагонали.
11. Задана квадратная матрица **Y(5,5).** Определить, где больше четных элементов: выше или ниже главной диагонали?
12. Дана действительная квадратная матрица **C(7,7).** Найти максимальный элемент в главной диагонали и напечатать элементы столбца, в которой он находится.
13. Заполнить квадратную матрицу X(7,7) следующим образом: элементы, расположенные на главной диагонали, принять равными 1; выше главной диагонали – сумме индексов; ниже – их разности.
14. Дана матрица размера N x M. В каждой строке найти количество элементов, больших среднего арифметического всех элементов матрицы.
15. Дана матрица B(k,p). Определить сумму элементов, кратных 3, и количество отрицательных элементов.
16. Дана действительная квадратная матрица M(5,5). Требуется переставить строки матрицы по возрастанию первых элементов строк.

Контрольные вопросы

1. Что такое двумерный массив? Назовите способы его описания.
2. Как описать переменные для хранения следующей информации:

Массив

5	11	6	0
91	3	-12	2
14	1	219	7

Лабораторная работа № 16

Тема. Решение задач с использованием переменной типа Record

Цель работы: приобретение навыков работы с переменной типа Record

Теоретическая часть

Запись представляет собой совокупность ограниченного числа логически связанных компонентов, принадлежащих к разным типам. Компоненты записи называются полями, каждое из которых определяется именем. Поле записи содержит имя поля, вслед за которым через двоеточие

указывается тип этого поля. Поля записи могут относиться к любому типу, допустимому в языке Паскаль, за исключением файлового типа.

Описание записи в языке ПАСКАЛЬ осуществляется с помощью служебного слова RECORD, вслед за которым описываются компоненты записи. Завершается описание записи служебным словом END.

Например, записная книжка содержит фамилии, инициалы и номера телефона, поэтому отдельную строку в записной книжке удобно представить в виде следующей записи:

```
type   Row=Record
          FIO: String[20];
          TEL: String[7]
         end;
var    str: Row;
```

Описание записей возможно и без использования имени типа, например:

```
var  str: Record
          FIO: String[20];
          TEL: String[7]
         end;
```

Обращение к записи в целом допускается только в операторах присваивания, где слева и справа от знака присваивания используются имена записей одинакового типа. Во всех остальных случаях оперируют отдельными полями записей. Чтобы обратиться к отдельной компоненте записи, необходимо задать имя записи и через точку указать имя нужного поля, например:

```
str.FIO,   str.TEL
```

Такое имя называется составным. Компонентой записи может быть также запись, в таком случае составное имя будет содержать не два, а большее количество имен.

Обращение к компонентам записей можно упростить, если воспользоваться оператором присоединения with.

Он позволяет заменить составные имена, характеризующие каждое поле, просто на имена полей, а имя записи определить в операторе присоединения:

```
with M do OP;
```

Здесь M – имя записи, OP – оператор, простой или составной. Оператор OP представляет собой область действия оператора присоединения, в пределах которой можно не использовать составные имена.

Пример 16.1. Создать базу данных о студентах student, содержащую следующие поля:

- Номер зачетки (**n_zach**);
- фамилия и инициалы (**fio**);

- средний балл (**sr_b**);
- количество сданных зачетов и экзаменов (**kz**);
- форма обучения (**fo**).

Написать программу, выполняющую следующие действия:

1) ввод с клавиатуры данных в массив, состоящий из n записей;
2) вывод на экран всей введенной информации;
3) вывод на экран информации о студентах, которые получают стипендию (форма обучения – **b**, количество зачетов и экзаменов – 10, средний балл >=7**.5**).

```
Program Project1;
uses crt;
type student=record
                    n_zach:string[5];
                    fio:string[35];
                    sr_b:real;
                    kz:byte;
                    fo:string[2];
                    end;

var s:array[1..30] of student;
n,k,i:integer;srbg:real;

begin
write('n='); readln(n);
for i:=1 to n do        // начало цикла for для ввода исходных данных
with s [i] do
begin
writeln('n зачетки'); readln(n_zach);
writeln('ФИО'); readln(fio);
writeln('средний балл?'); readln(sr_b);
writeln('количество зачетов?'); readln(kz);
writeln('форма обучения?'); readln(fo);
end;                        // конец цикла for для ввода исходных данных

k:=0;
for i:=1 to n do
      with s [i] do
  begin
       if(fo='b') and (kz=10)and (sr_b>=7.5) then
                        begin
                        writeln(n_zach,'  ',fio,'  ',sr_b:3:1,'  ',kz,' ',fo);
                        k:=k+1;
                              end;
```

```
writeln('количество студентов, которые получают стипендию ',k:2);
readln;
end.
```

Пример 16.2. Необходимо организовать массив записей Паскаля, а затем из общего списка вывести фамилии студентов 2-го курса.

```
program Project2;
type anketa=record
            fio: string[45];
            dat_r: string[8];
            adres: string[50];
            curs: 1..5;
            grupp: string[3]
      end;
var student: array [1..100] of anketa;  I: integer;
begin
  //последовательно вводим каждую запись
  for I:=1 to 100 do
  begin
    writeln ('введите сведения о', I , '-м студенте');
    writeln ('введите фамилию, имя и отчество');  readln (student[I].fio);
    writeln ('введите дату рождения');   readln (student[I].dat_r);
    writeln ('введите адрес');   readln(student[I].adres);
    writeln ('введите курс');   readln(student[I].curs);
    writeln ('введите группу');  readln (student[I].grupp);
  end;
  writeln ('ввод закончен');
  writeln ;
  //просматриваем массив записей и выбираем только студентов 2-го курса
  for I:=1 to 100 do
    if student[I].curs=2 then
      writeln(' фамилия студента : ', student[I].fio);
end.
```

Например, фрагмент из предыдущей программы с использованием оператора присоединения (with …. Do ….) будет выглядеть так:
Фрагмент примера

```
for I:=1 to 100 do
  with student[I] do
  begin
    writeln ('введите сведения о', I , '-м студенте');
    writeln ('введите фамилию, имя и отчество');  readln (fio);
    writeln ('введите дату рождения');   readln (dat_r);
```

```
    writeln ('введите адрес');  readln(adres);
    writeln ('введите курс');    readln(curs);
    writeln ('введите группу');  readln (grupp);
  end;
```

Варианты заданий

В программе использовать одномерный массив с элементами массива типа Record. Исходные данные вводятся с клавиатуры в 10 элементов массива. Программа должна выводить соответствующее сообщение, если искомых данных в массиве нет.

1. Описать структуру с именем **STUDENT**, содержащую следующие поля: фамилия и инициалы; номер группы; успеваемость (средний балл).

Запрограммировать вывод на экран фамилий и номеров групп для всех студентов, включенных в массив, если средний балл студента больше 4,5.

2. Описать структуру с именем **STUDENT**, содержащую следующие поля: фамилия и инициалы; номер группы; успеваемость (средний балл).

Запрограммировать вывод на экран фамилий и номер группы для студентов, имеющих средний балл 4 или 5.

3. Описать структуру с именем **STUDENT**, содержащую следующие поля: фамилия и инициалы; номер группы; успеваемость (средний балл).

Запрограммировать вывод на экран фамилий и номеров групп для всех студентов, имеющих средний балл от 6 до 8.

4. Описать структуру с именем **AEROFLOT**, содержащую следующие поля: название пункта назначения рейса; номер рейса; тип самолета.

Запрограммировать вывод на экран номеров рейсов и типов самолетов, вылетающих в пункт назначения, название которого совпало с названием, введенным с клавиатуры.

5. Описать структуру с именем **AEROFLOT**, содержащую следующие поля: название пункта назначения рейса; номер рейса; тип самолета.

Запрограммировать вывод на экран пунктов назначения и номеров рейсов, обслуживаемых самолетом, тип которого введен с клавиатуры.

6. Описать структуру с именем **WORKER**, содержащую следующие поля: фамилия и инициалы работника; название занимаемой должности; год поступления на работу.

Запрограммировать вывод на экран фамилий работников, чей стаж работы в организации превышает значение стажа, введенного с клавиатуры.

7. Описать структуру с именем **POEZD**, содержащую следующие поля: название пункта назначения; номер поезда; время отправления.

Запрограммировать вывод на экран информации о поездах, отправляющихся после введенного с клавиатуры времени.

8. Описать структуру с именем **POEZD**, содержащую следующие поля: название пункта назначения; номер поезда; время отправления.

Запрограммировать вывод на экран информации о поездах, направляющихся в пункт, название которого введено с клавиатуры.

9. Описать структуру с именем **POEZD**, содержащую следующие поля: название пункта назначения; номер поезда; время отправления.

Запрограммировать вывод на экран информации о поезде, номер которого введен с клавиатуры.

10. Описать структуру с именем **MARSH**, содержащую следующие поля: название начального пункта маршрута; название конечного пункта маршрута; номер маршрута.

Запрограммировать вывод на экран информации о маршруте, номер которого введен с клавиатуры.

11. Описать структуру с именем **MARSH**, содержащую следующие поля: название начального пункта маршрута; название конечного пункта маршрута; номер маршрута.

Запрограммировать вывод на экран информации о маршрутах, которые начинаются или оканчиваются в пункте, название которого введено с клавиатуры.

12. Описать структуру с именем **NOTE**, содержащую следующие поля: фамилия, имя; номер телефона; год рождения.

Запрограммировать вывод на экран информации о человеке, номер телефона которого введен с клавиатуры.

13. Описать структуру с именем **NOTE**, содержащую следующие поля: фамилия, имя; номер телефона; год рождения.

Запрограммировать вывод на экран информации о людях, чьи дни рождения приходятся на год, значение которого введено с клавиатуры.

14. Описать структуру с именем **NOTE**, содержащую следующие поля: фамилия, имя; номер телефона; год рождения.

Запрограммировать вывод на экран информации о человеке, чья фамилия и имя введены с клавиатуры.

15. Описать структуру с именем **ZNAK**, содержащую следующие поля: фамилия, имя, отчество; месяц рождения; год рождения.

Запрограммировать вывод на экран информации о людях, родившихся в месяце, название которого введено с клавиатуры.

16. Описать структуру с именем **ZNAK**, содержащую следующие поля: фамилия, имя, отчество; знак Зодиака; год рождения.

Запрограммировать вывод на экран информации о людях, родившихся под знаком Зодиака, название которого введено с клавиатуры.

Литература

1. Архангельский, А. Я. Программирование в Delphi : учебник по классическим версиям Delphi / А. Я. Архангельский. – Москва : Бином, 2008. – 1154 с.
2. Бобровский, С. И. Delphi 7: учебный курс / С. И. Бобровский. – Санкт-Петербург : Питер, 2008. – 736 с.
3. Графические средства Delphi: методические указания для самостоятельной работы студентов механических специальностей дневной и заочной форм обучения по дисциплинам : "Информатика", "Вычислительная техника, программирование и расчеты на ЭВМ" / УО "ВГТУ" ; сост. В. П. Терентьев, Т. П. Стасеня. – Витебск, 2007. – 61 с.
4. Климов, Ю. С. Программирование в среде Turbo Pascal 6.0 : справ. пособие / Ю. С. Климов, А. И. Касаткин, С. М. Мороз. – Минск : Вышэйшая школа, 1992. – 158 с.
5. Методические указания к курсовому проектированию для студентов механических специальностей по предметам цикла "Информатика" / УО "ВГТУ" ; сост. В. П. Терентьев, А. С. Дягилев, Т. П. Стасеня. – Витебск : УО "ВГТУ", 2004. – 81 с.
6. Программирование на персональных ЭВМ : практикум / под ред. Д. В. Офицерова. – Минск : Вышэйшая школа, 1993. – 256 с.
7. Сурков, Д. А. Программирование в среде Borland Pascal для Windows : справочное пособие / Д. А. Сурков, К. А. Сурков, А. Н. Вальвачев. – Минск : Вышэйшая школа, 1996. – 432 с.
8. Фаронов, В. В. Delphi : учебник / В. В. Фаронов. – Санкт-Петербург : Питер, 2010. – 640 с.
9. Фаронов, В. В. Delphi. Программирование на языке высокого уровня : учебник для студентов вузов, обучающихся по направлению подготовки дипломированных специалистов "Информатика и вычислительная техника" / В. В. Фаронов. – Санкт-Петербург : Питер, 2010, 2007. – 640 с.
10. Фаронов, В. В. Основы Турбо Паскаля : учеб. пособие / В. В. Фаронов. – Москва : Учебно-инженерный центр «МВТУ-ФЕСТО ДИДАКТИК», 1992. – 304 с.
11. Язык программирования Object Pascal (Delphi) : курс лекций по дисциплине "Вычислительная техника, программирование и расчеты на ЭВМ" для студентов механических спец. / УО "ВГТУ" ; сост. В. П. Терентьев. – Витебск : УО "ВГТУ", 2005. – 75 с.