



Министерство образования и науки Республики Казахстан

Павлодарский государственный университет им. С. Торайгырова

Кафедра «Вычислительная техника и программирование»

# **МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

к лабораторным работам

по дисциплине «Программирование на алгоритмических языках»

для студентов специальности 050704 «Вычислительная техника и программное обеспечение»

Лист утверждён  
методическим указанием  
СО ПГУ 7.18.1/05



Ф

**УТВЕРЖДАЮ**

Декан ФМиИТ

\_\_\_\_\_ С.К. Тлеуменов

«\_\_\_» \_\_\_\_\_ 200\_г

Составители: ст. преподаватель \_\_\_\_\_ О.В. Загребельная  
преподаватель \_\_\_\_\_ З.Р. Ахмерова

Кафедра «Вычислительная техника и программирование»

### **Методические указания**

к лабораторным работам

по дисциплине «Программирование на алгоритмических языках»

для студентов специальности 050704 «Вычислительная техника и программное обеспечение»

**Рекомендовано** на заседании кафедры от «\_\_\_» \_\_\_\_\_ 200\_ г.

Протокол №\_\_\_

Заведующий кафедрой \_\_\_\_\_ О.Г. Потапенко  
(подпись)

**Одобрено** методическим советом факультета ФМиИТ

«\_\_\_» \_\_\_\_\_ 200\_ г. Протокол №\_\_\_

Председатель МС \_\_\_\_\_ А. З. Даутова  
(подпись)

## Содержание

Введение.....	4
1 Разработка программы, управляющей текстовым экраном, звуком и клавиатурой (лабораторная работа №1).....	5
1.1 Теоретические сведения.....	5
1.2 Указания по выполнению лабораторной работы.....	7
2 Разработка программы последовательной структуры с вводом-выводом данных (лабораторная работа №2).....	8
2.1 Теоретические сведения.....	8
2.2 Указания по выполнению лабораторной работы.....	11
3 Разработка программы разветвляющейся структуры (лабораторная работа №3).....	12
3.1 Теоретические сведения.....	12
3.2 Указания по выполнению лабораторной работы.....	14
4 Разработка программы циклической структуры (лабораторная работа №4).....	17
4.1 Теоретические сведения.....	17
4.2 Указания по выполнению лабораторной работы.....	19
5 Разработка программы с использованием массивов (лабораторная работа №5).....	24
5.1 Теоретические сведения.....	24
5.2 Указания по выполнению лабораторной работы.....	25
6 Разработка программы с использованием типа данных “запись”(RECORD) (лабораторная работа №6).....	27
6.1 Теоретические сведения.....	27
6.2 Указания по выполнению лабораторной работы.....	28
7 Разработка программы, обрабатывающей файловые данные (лабораторная работа №7).....	31
7.1 Теоретические сведения.....	31
7.1.1 Файлы в Turbo Pascal.....	31
7.1.2 Текстовые файлы.....	32
7.1.3 Типизированные файлы.....	33
7.2 Указания по выполнению лабораторной работы.....	34
8 Разработка программы с применением процедур и функций (лабораторная работа №8).....	37
8.1 Теоретические сведения.....	37
8.2 Указания по выполнению лабораторной работы.....	38
9 Разработка программ, обрабатывающих строковые (STRING) и символьные (CHAR) данные (лабораторная работа №9).....	39
9.1 Теоретические сведения.....	39
9.2 Указания по выполнению лабораторной работы.....	40
Литература.....	41

## **Введение**

Предлагаемый лабораторный практикум из десяти работ включает в себя методический материал, относящийся к изучению языка программирования Pascal, также рассматриваются стандартные типы данных, основные конструкции языка, массивы, строки и структуры, функции, работа с файлами.

## 1 Разработка программы, управляющей текстовым экраном, звуком и клавиатурой (лабораторная работа №1)

**Цель работы:** научить студентов разрабатывать программы, имеющие многооконный интерфейс и сопровождаемые звуковым эффектом.

### 1.1 Теоретические сведения

Для разработки профессионального интерфейса программных продуктов в Паскале предусмотрено подключение стандартного модуля CRT с помощью следующей строки

Uses crt;

Эта строчка обычно располагается сразу за строкой

Program имя:

Модуль CRT обеспечивает практически полный спектр возможностей для доступа к экрану дисплея в текстовом режиме. Кроме того, в данный модуль включены средства чтения информации с клавиатуры (включая расширенные коды клавиш) и простейшего управления звуком. Модуль CRT содержит константы, переменные и процедуры, предназначенные для работы с консолью компьютера. В отличие от стандартного ввода-вывода, который осуществляется через операционную систему, подпрограммы модуля CRT работают с BIOS и непосредственно с видеопамью.

При работе с экраном через модуль CRT весь экран разбивается на отдельные строки, а каждая строка - на отдельные позиции, в каждую из которых можно поместить один символ. Таким образом, экран разбивается на отдельные неделимые элементы. Для каждого элемента можно задать цвет фона (задний план) и цвет символа (передний план), кроме того, можно сделать символ мерцающим.

Программист может сам определять размеры активного окна для ввода и вывода информации. Для этого существует процедура  $Window(x1,y1,x2,y2)$ , где  $x1,y1$  - координаты левого верхнего угла экрана,  $x2,y2$  - координаты правого нижнего угла экрана.

Цвет фона в активном окне задается с помощью процедуры  $Textbackground(m)$   
Где  $m$  - номер цвета фона ( $m$  может принимать значения от 0 до 7).  
Например,  
 $Textbackground(7)$ .

Цвет символа можно задать с помощью процедуры  $Textcolor(m)$

Где  $m$  - цвет символа ( $m$  может принимать значения от 0 до 15).

Коды цветов

0- черный

1- синий

2- зеленый

3- голубой

4- красный

5- фиолетовый

6- коричневый

7- светло-серый

8- темно-серый	12- розовый
9- светло-синий	13- светло-фиолетовый
10- светло-зеленый	14- желтый
11- светло-голубой	15- белый.

Для получения эффекта мерцания символов, необходимо установить "номер цвета +16" в качестве параметра в процедуре Textcolor.

Например, красный мерцающий символ может быть получен после вызова процедуры  
Textcolor(20).

С помощью процедуры Clrscr происходит очистка активного окна, причем, окно закрашивается тем цветом, который был установлен в процедуре Textbackground(t). Например:

```
Window(2,5,40,10);
Textbackground(7); clrscr;
```

Окно, с параметрами левого верхнего угла: x1=2,y1=5 и правого нижнего угла: x2=40, y2=10, будет очищено светлосерым цветом.

Удалить символы до конца строки в активном окне можно с помощью процедуры Clreol.

Удаление строки, в которой находится курсор с экрана может быть выполнено с помощью процедуры Deline. Добавление пустой строки на экран, перед строкой, в которой находится курсор выполняется процедурой Incline.

Перемещение курсора в заданную позицию экрана может быть выполнено процедурой Gotoxy(x1,y1). Здесь x1,y1 - координаты экрана.

Координаты положения курсора можно узнать с помощью функций Wherex и Wherey . Например, после выполнения следующего оператора в переменную целого типа posx будет занесена координата курсора x:  
Posx:= Wherex;

Простейший звуковой эффект может быть запрограммирован с помощью процедуры  
Sound(hz);

Здесь hz - частота звука в герцах. Так нота "до"-это 262 гц, "ре"- 294гц,"ми"- 330гц, "фа"- 349 гц, " соль"- 392гц, "ля"- 440гц, 2си2- 494гц, "до" следующей октавы- 524гц.

Прекращение звучания **ОБЯЗАТЕЛЬНО** должно быть выполнено процедурой Nosound. В противном случае звук не прекращается даже после выхода из программы!

Процедура Delay(ms) задерживает выполнение программы на ms миллисекунд. Таким образом, с помощью данной процедуры можно достичь различного по длительности звучания

Приведем еще две очень полезные функции, с помощью которых можно организовать профессиональную работу с клавиатурой и экраном. Функция Readkey считывает символ с клавиатуры, освобождает буфер

клавиатуры от этого символа и помещает его в переменную, стоящую слева в выражении:

a:= Readkey;

Функция Keypressed анализирует нажатие клавиш клавиатуры, то есть наличие в буфере клавиатуры символов (за исключением вспомогательных клавиш - Alt, Shift Ctrl и т.п.). Результат выполнения функции "истина", если была нажата хотя бы одна клавиша и ее символ остался в буфере клавиатуры (заметим, что после нажатия клавиши Enter информация из буфера помещается в переменную, которая ждет ее.).

## 1.2 Указания по выполнению лабораторной работы

Написать и отладить программу, выводящую на экран два окна с эффектом тени (тень окрасить в черный цвет). Цвет окон и фона произвольный. Цвет текста не должен сливаться с цветом окна. Размеры окон и их местоположение на экране должно быть достаточны для вывода следующей информации (ниже приведен пример работы программы):

- в окне, расположенном выше, вывести текст: в первой строке – "Лабораторная работа №1", во второй - "ПГУ им. С.Торайгырова";
- в окне, расположенном ниже, вывести в первой строке - "Авторы программы: имена", во второй строке - реквизиты авторов программы, возможный рекламный текст;
- программу оснастить небольшим по длительности звуковым сопровождением (не более 15 секунд);
- произвести задержку экрана с помощью оператора delay.

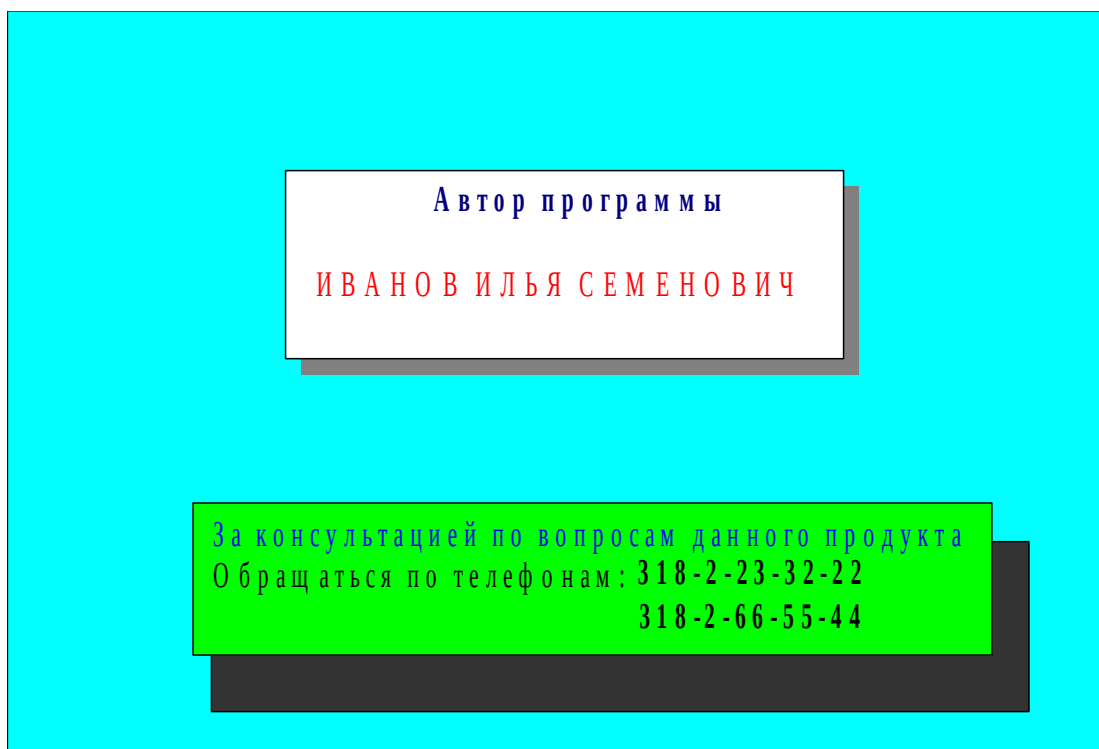


Рисунок 1 – Интерфейс программы, управляющей текстовым экраном, звуком и клавиатурой

## 2 Разработка программы последовательной структуры с вводом-выводом данных (лабораторная работа №2)

**Цель работы:** освоить переменные, типы переменных, константы, типизированные константы, арифметические операции, оператор присваивания, ввод и вывод информации, форматирование вывода.

### 2.1 Теоретические сведения

В программе на языке Паскаль любая используемая переменная за исключением системных, должна быть описана, причем определение переменной должно текстуально предшествовать ее первому использованию. Область "видимости" переменной ограничивается блоком. Переменные определяются посредством специальной конструкции. Описание переменных начинается со служебного слова `var`, вслед за которым располагается последовательность самих определений переменных.

Определение переменной должно содержать два элемента: имя переменной и ее тип. Имя переменной задается с помощью идентификатора, который "придумывает" сам программист. Имя в Паскале может содержать только буквы латинского алфавита, цифры и некоторые символы, причем, первым в имени должна быть обязательно буква.

Идентификатор переменной и ее тип разделяются двоеточием. Например,

```
Var fgh, g1, t234 : integer;
    Ds : char;
    S22, f, h_rav: real;
    Sos : string;
```

В Турбо Паскале можно выделить следующие группы типов:

- простые типы;
- структурированные типы;
- указатели;
- процедурные типы;
- объекты.

Среди типов, используемых в языке, есть стандартные, не требующие предварительного определения (предопределенные), и определяемые программистом.

К стандартным типам относятся:

- целые типы;



Таблица 1 – Целые типы данных

тип	диапазон	формат	кол. байт
shortint	-128..127	знаковый	1
integer	-32768..32767	знаковый	2
longint	-2147483648..2147483647	знаковый	4
byte	0 ..255	беззнаковый	1
word	0 .. 65535	беззнаковый	2

- логический тип (boolean, значения переменной данного типа могут быть "true" и "false", размер - 1 байт);

- символьный тип (char, значения переменной данного типа - коды символов, размер - 1 байт);

- вещественные типы;

Таблица 2 – Вещественные типы данных

тип	диапазон	число значащих цифр	размер
real	$2.9 \cdot 10^{-39} \dots 1.7 \cdot 10^{+38}$	11-12	6 байт
single	$1.5 \cdot 10^{-45} \dots 3.4 \cdot 10^{+38}$	7-8	4 байта
double	$5.0 \cdot 10^{-324} \dots 1.7 \cdot 10^{+308}$	15-16	8 байт
extended	$3.4 \cdot 10^{-4932} \dots 1.1 \cdot 10^{+4931}$	19-20	10 байт

Паскаль может работать с объектами, внешне похожими на переменные, однако не допускающие переопределения своего первоначального значения. Такие объекты называют константами. Их определение начинается со служебного слова const.

Ниже приведен пример описания констант.

```
Const gz=500;
      S=3.245;
      P=3.1442;
      Ident="кот ел рыбу";
```

В Паскале существуют также типизированные константы, которые инициализируются только один раз - в начале выполнения программы, а затем, в течение работы программы их значения могут быть переопределены. Этот механизм можно использовать при задании элементов массивов. Особенно он удобен при отладке программы.

Ниже приведен пример описания типизированных констант.

```
Const ggg: integer =1990;
      Min: real = 0;
```

Max: real =99999;

Iden: array[1..4] of byte =(122,0,100,9);

D12: array[1..2,1..3] of real = ((0,0,0),(1,1,1),(99,100,300));

Арифметические операции применимы только к величинам целых и вещественных типов. К арифметическим операциям относятся следующие:

Таблица 3 – Арифметические операции

знак	операция	типы операндов	тип результата
+	сложение	целые	целый
		хотя бы один вещ-й	вещественный
-	вычитание	целые	целый
		хотя бы один вещ-й	вещественный
*	умножение	целые	целый
		хотя бы один вещ-й	вещественный
/	деление	целые или веществ	вещественный
div	деление на цело	целые	целый
mod	остаток от деления	целые	целый

Оператор присваивания в Паскале служит для присвоения значения переменной. Для этого используется знак присваивания ":=", слева от которого записывается имя переменной, а справа - выражение, значение которого вычисляется перед присваиванием.

Тип выражения и тип переменной (или функции) должны быть совместимы для присваивания (см. типы операндов в таблице, приведенной выше). Примеры:

```
D22:= y*sin(a10);
```

```
Z:= 0.00235;
```

```
Res:= sqrt(Z)+D22;
```

Ввод данных с клавиатуры в переменные, описанные в программе может быть выполнен с помощью операторов read(список ввода) и readln(список ввода). После выполнения оператора read курсор остается в текущей позиции курсора, а после readln - курсор перемещается на следующую строку экрана. Ниже приведены примеры применения этих операторов.

```
Read(d11,x01,a0);
```

```
Readln(x1,cc,sos);
```

При выводе на экран монитора, который выполняется с помощью операторов Write(список вывода) или Writeln(список вывода) может быть задано форматирование выводимых результатов следующим образом:

a:m:n, где a - имя выводимой переменной, m - количество позиций на экране для вывода значения этой переменной, n - в том числе под дробную часть.

Примеры применения операторов вывода:

```
Write('вывод результатов:' , y,s12);
```

```
Writeln('y=', y:10:3,'z11=', z11:7:2).
```

## 2.2 Указания по выполнению лабораторной работы

Написать и отладить программу, выводящую на экран по два окна с эффектом тени (тень окрасить в черный цвет) для трех различных формул (арифметические выражения). На экране должны быть сформированы записи, поясняющие работу программы. Для ввода очередного набора данных курсор должен перемещаться в соответствующее окно (среднее в каждой строке), а результат вычисления помещаться в крайнее окно. Цвет окон и фона произвольный. Цвет текста не должен сливаться с цветом окна. Размеры окон и их местоположение на экране должно быть достаточны для вывода следующей информации:

- в окне, расположенном левее, вывести формулу, по которой будет производиться вычисление;
- в окне, расположенном в середине необходимо организовать ввод необходимых для вычисления данных;
- в крайнем окне вывести результат вычисления формулы;
- программа должна выводить на экран все девять окон с комментирующей информацией и только после появления сообщения "ВВЕДИ ЗНАЧЕНИЯ ПЕРЕМЕННЫХ" курсор должен перемещаться в нужное окно и нужную позицию окна для ввода значения соответствующей переменной;
- произвести задержку экрана с помощью оператора delay.

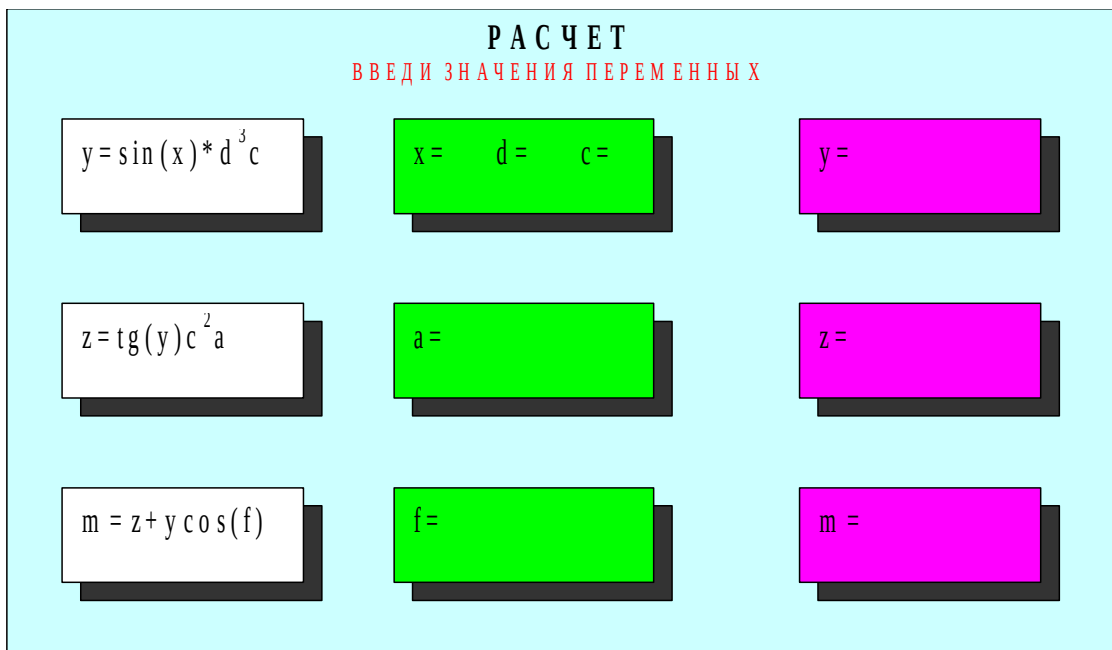


Рисунок 2 – Интерфейс программы последовательной структуры с вводом-выводом данных

Примечание. Окно становится активным после выполнения оператора `window(x1,y1,x2,y2)`. Этот оператор может встречаться с равными параметрами неограниченное количество раз в тексте одного программного

модуля. Далее внутри активизированного окна место положения курсора может быть изменено с помощью оператора gotoxy(x3,y3).

### **3 Разработка программы разветвляющейся структуры (лабораторная работа №3)**

**Цель работы:** освоить применение простого и составного операторов, базовых операций, логических операций NOT, AND, OR, оператора IF-THEN-ELSE, операторов GOTO и CASE OF.

#### **3.1 Теоретические сведения**

Все операторы языка Паскаль можно разбить на две группы: простые и составные (структурированные). Простыми являются те операторы, которые не содержат в себе других операторов. К ним относятся:

- оператор присваивания;
- обращение к процедуре;
- оператор безусловного перехода.

Оператор присваивания и обращения к процедуре (на примере процедур модуля CRT) были рассмотрены в предыдущих лабораторных работах. Рассмотрим оператор безусловного перехода.

Оператор GOTO m позволяет изменить стандартный последовательный порядок выполнения операторов и перейти к оператору, помеченному меткой m. Метки в Турбо Паскале могут быть целым числом в пределах от 0 до 9999 или обычным идентификатором. Все используемые метки должны быть объявлены в разделе описаний после стандартного слова label. Например:

```
Label m0, M1, m22, 10, 20;
```

К структурированным операторам относятся:

- составной оператор;
- условный оператор IF;
- оператор выбора варианта CASE;
- оператор цикла REPEAT;
- оператор цикла WHILE;
- оператор цикла FOR;
- оператор над записями WITH.

Составной оператор представляет собой совокупность последовательно выполняемых операторов, заключенных в операторные скобки begin ... end. Например:

```
begin
    read(f,a);
```

```
y1:= sin(f)* cos(a)/2.43;
```

```
end;
```

Оператор IF реализует алгоритмическую конструкцию РАЗВИЛКА и изменяет порядок выполнения операторов в зависимости от истинности или ложности некоторого условия. Существует два варианта оператора:

```
If логическое_выражение then оператор 1
```

```
Else оператор2;
```

и

```
If логическое_выражение then оператор1;
```

Если логическое\_выражение истинно, то выполняется оператор1, если ложно - то оператор2;

Условный оператор CASE позволяет запрограммировать выбор из любого количества вариантов. Структура этого оператора:

```
Case s of
```

```
C1: оператор1;
```

```
C2: оператор2;
```

```
.....
```

```
Cn : оператор n;
```

```
Else оператор
```

```
End;
```

В этом операторе s - выражение порядкового типа, значение которого вычисляется и сравнивается с одним из возможных значений C1, C2... Cn. Если значение совпало, то выполняется соответствующий ему оператор, в противном случае - оператор, стоящий после Else. Например:

```
Read (ch); read(a,b);
```

```
Case ch of
```

```
 '+': y:= a+b;
```

```
 '-': y:= a-b;
```

```
 '*': y:= a*b;
```

```
Else y:= a/b
```

```
End;
```

Логические выражения могут быть простыми и сложными, то есть соединенными знаками логических операций NOT, AND, OR .

Ниже приводится таблица истинности этих операций.

Таблица 4 - Таблица истинности логических операций

X	Y	X and Y	X or Y	not X
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Примеры сложных логических выражений:

$x > 0$  AND  $x \leq 5$

NOT  $s = 10$  AND  $x = 15$

### 3.2 Указания по выполнению лабораторной работы

Выбрать задание. Написать и отладить программу с применением операторов IF-THEN-ELSE, GOTO, CASE OF. При составлении алгоритма воспользоваться логическими операциями NOT, AND, OR. Программу составить с выходом из бесконечного цикла по условию ввода определенного значения.

#### Варианты заданий

1. Заданы площади круга R и квадрата S. Определить, поместится ли круг в квадрате. Выдать сообщения: 1- диагональ квадрата больше диаметра круга, квадрат не поместится в круге 2- диагональ квадрата равна диаметру круга, квадрат поместится в круге, 3- диагональ квадрата меньше диаметра круга, квадрат поместится в круге.
2. Написать программу, которая спрашивала полное имя, а на печать выводила сокращенное (Мария-Маша) для пяти ваших друзей. Ввод незнакомого имени должен провоцировать заявление типа: "Я с вами не знакома,...".
3. Составить программу, реализующую эпизод сказки: спрашивает, куда предпочитает пойти герой (налево, направо или прямо) и печатает, что его ждет в каждом случае.
4. Составить программу, контролирующую знание закона Ома, законов Ньютона. Обучаемый вводит формулу закона в символьную переменную большими буквами.
5. Заданы площади круга R и квадрата S. Определить, поместится ли квадрат в круге. Выдать сообщения: 1- диагональ квадрата больше диаметра круга, квадрат не поместится в круге 2- диагональ квадрата равна диаметру круга, квадрат поместится в круге, 3- диагональ квадрата меньше диаметра круга, квадрат поместится в круге.
6. Составить программу, которая запрашивает имя и здоровается с его обладателем, если это имя "известно программе" и выдает ответ: "Простите, я с вами не знакома". Количество "знакомых" имен должно быть не менее трех.

7. Заданы координаты двух точек. Определить, лежат ли они на одной окружности с центром в начале координат. Выдать сообщения: 1- одна лежит на окружности и ее координаты, 2-ни одна точка не лежит на окружности, 3- обе точки лежат на окружности.
8. На плоскости расположена окружность с центром не в начале координат. Определить, лежит ли заданные две точки внутри окружности. Выдать сообщения: 1- одна лежит внутри окружности и ее координаты, 2-ни одна точка не лежит внутри окружности, 3- обе точки лежат внутри окружности.
9. Заданы координаты двух точек. Определить, принадлежат ли точка фигуре, ограниченной осью абсцисс и прямыми  $y=x+1$  и  $y=-x+1$ . Выдать сообщения: 1- одна точка принадлежит данной фигуре, и ее координаты:..., 2-ни одна точка не принадлежит фигуре, 3- обе точки принадлежат фигуре.
10. Заданы координаты двух точек. Определить, принадлежит ли точка фигуре, ограниченной осью абсцисс и кривой  $y=\sin x$  на отрезке от 0 до  $2\pi$ . Выдать сообщения: 1- одна точка принадлежит данной фигуре, и ее координаты:..., 2-ни одна точка не принадлежит фигуре, 3- обе точки принадлежат фигуре.
11. Заданы площади круга  $R$  и равностороннего треугольника  $S$ . Определить, поместится ли круг в треугольнике. Выдать сообщения: 1- треугольник не поместится в круге 2- круг вписан в треугольник, 3- треугольник поместится в круге.
12. Написать программу, которая спрашивала полное имя, а печатала сокращенное (Михаил-Миша) для десяти ваших друзей. Ввод незнакомого имени должен провоцировать заявление типа: "Я с вами не знакома,...".
13. Составить программу, реализующую эпизод сказки: спрашивает, куда предпочитает пойти герой (налево, направо, прямо или остаться на месте) и печатает, что его ждет в каждом случае.
14. Составить программу, контролирующую знание закона Ома, законов Ньютона. Обучаемый вводит формулу закона в символьную переменную большими буквами.
15. Заданы площади круга  $R$  и равностороннего треугольника  $S$ . Определить, поместится ли треугольник в круге. Выдать сообщения: 1- треугольник не поместится в круге 2- треугольник вписан в круг, 3- треугольник поместится внутри круга.
16. Составить программу, которая запрашивает имя и здоровается с его обладателем, если это имя "известно программе" и выдает ответ: "Простите, я с вами не знакома". Количество "знакомых" имен должно быть не менее пяти. Работа программы прекращается при вводе вместо имени символов: "\*\*\*".
17. Заданы координаты трех точек. Определить, лежат ли они на одной окружности с центром в начале координат. Выдать сообщения: 1- одна лежит на окружности и ее координаты, 2-ни одна точка не лежит на окружности, 3- две точки лежат на окружности, 4- три точки лежат на окружности.

18. На плоскости расположена окружность с центром не в начале координат. Определить, лежат ли заданные три точки внутри окружности. Выдать сообщения: 1- ни одна не лежит внутри окружности и ее координаты, 2- одна точка лежит внутри окружности, 3- две точки лежат внутри окружности, 4- три точки лежат внутри окружности.

19. Заданы координаты трех точек. Определить, принадлежат ли точка фигуре, ограниченной осью абсцисс и прямыми  $y=x+1$  и  $y=-x+1$ .

Выдать сообщения: 1- ни одна точка не принадлежит фигуре, 2- одна точка принадлежит данной фигуре, и ее координаты:..., 3- две точки принадлежат фигуре, и их координаты:..., 4- три точки принадлежат данной фигуре, и их координаты:....

20. Заданы координаты трех точек. Определить, принадлежит ли точка фигуре, ограниченной осью абсцисс и кривой  $y=\cos x$  на отрезке от 0 до  $2\pi$ .

Выдать сообщения: 1- ни одна точка не принадлежит фигуре, 2- одна точка принадлежит данной фигуре, и ее координаты:..., 3- две точки принадлежат фигуре, 4- три точки принадлежат фигуре, и их координаты ... .



## 4 Разработка программы циклической структуры (лабораторная работа №4)

**Цель работы:** освоить применение оператора цикла FOR...DO с возможностью задания отрицательного шага изменения переменной, оператора цикла DOWNTO, оператора цикла с предусловием WHILE и оператора цикла с постусловием REPEAT...UNTIL.

### 4.1 Теоретические сведения

Оператор цикла FOR организует выполнение одного оператора заранее известное число раз. Существует два варианта оператора.

FOR par:=start TO finish DO оператор ;

FOR par:=start DOWNTO finish DO оператор ;

В этих операторах :

par -параметр цикла, являющийся переменной порядкового типа;

start - выражение, определяющее начальное значение параметра цикла;

finish- выражение, определяющее конечное значение параметра цикла.

Цикл действует таким образом: сначала вычисляются и запоминаются начальное и конечное значение параметра цикла. Далее параметру цикла присваивается начальное значение start, затем сравнивается с конечным значением finish . Далее, пока параметр цикла меньше или равен конечному значению (в первом варианте оператора) или больше или равен конечному значению (во втором варианте оператора), выполняется очередная итерация цикла; в противном случае происходит выход из цикла.

Выполнение очередной итерации включает в себя сначала вычисление оператора, а затем присваивание параметру цикла следующего большего на единицу (в первом случае) или меньшего на единицу (во втором случае) значения.

Естественно, что, если в первом варианте значение start больше finish , оператор не выполняется ни разу. Соответственно, если во втором варианте, значение start меньше finish , оператор не выполняется ни разу.

После выхода из цикла параметр цикла становится неопределенным за исключением случая, когда выход из цикла был осуществлен с помощью оператора GOTO.

Пример применения первого варианта оператора For (вывести на экран таблицу умножения на 9):

```
Var i, y: integer;
For i:=1 to 10 do
Begin y:= 9*i;
Writeln(' 9*',i:2,' = ',y:3);
End;
```

Пример применения второго варианта оператора For (вывести на экран числа от 10 до 0):

```
Var i: integer;
For i:=10 downto 0 do
```

```
Writeln(i:2);
```

Оператор цикла REPEAT организует выполнение цикла, состоящего из любого числа операторов с неизвестным заранее числом повторений. Тело цикла выполняется хотя бы один раз. Выход из цикла осуществляется при истинности логического выражения S. Структура этого оператора:

```
REPEAT
Оператор1;
Оператор2;
.....
ОператорN
UNTIL S;
```

Пример. Вычислить сумму двух чисел, работу программы прекратить при вводе символа "Y".

```
Var a,b,c: real;
    ch:char;
REPEAT
Write('Вводите числа a и b');
Readln(a,b);
c:=a+b;
writeln('Сумма чисел равна', c);
writeln('Завершить программу?');
readln(ch);
UNTIL (ch='Y') or (ch='y');
```

Оператор цикла WHILE организует выполнение одного оператора неизвестное заранее число раз. Выход из цикла осуществляется, если некоторое логическое условие S окажется ложным. Так как истинность логического выражения проверяется в начале каждой итерации, тело цикла может не выполниться ни разу. Структура оператора цикла имеет вид:

```
WHILE S DO
Оператор;
```

Пример. Найти сумму ряда  $y = \frac{1}{x}$ . Вычисление прекратить, когда значение слагаемого станет меньше 0.0001.

```
Var x,y:real;
y:=0;
WHILE 1/x > 0.0001 DO
Y:=y+1/x;
Writeln('y=',y:8:4);
```

## 4.2 Указания к выполнению лабораторной работы

Написать и отладить четыре программы с применением операторов FOR ... DO, FOR ... DOWNTO, WHILE, REPEAT...UNTIL.

### Варианты заданий

1. Составить программу помощника кассира в универсальном магазине. ЭВМ должна запрашивать цену товара и его количество, подсчитывать суммарную стоимость купленных товаров, запрашивать сумму денег, внесенных покупателем и определять причитающуюся ему сдачу. Прекращение работы программы должно произойти при вводе суммы за покупку равной или меньшей 0.

2. Составить программу, подсчитывающую число посещений в поликлинике врачей специалистов (отоларинголога, окулиста и хирурга). В конце дня выдавать итоговое сообщение. В начале работы программы должны вводиться дата, которая будет фигурировать в итоговом сообщении. Использовать множественный выбор.

3. Пассажирский самолет может поднять груз общим весом 30т. Составить программу для определения веса почтового груза, который можно поместить в самолет после посадки пассажиров и загрузки их багажа. Во время регистрации пассажиров ЭВМ должна подсчитывать количество пассажиров (условный вес одного человека - 100 кг) и суммировать вес багажа.

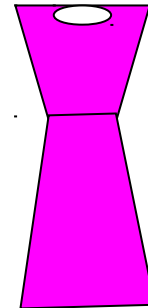
4. Составить программу для определения подходящего возраста кандидатуры для вступления в брак, используя следующее соображение: возраст девушки равен половине возраста мужчины плюс 7, возраст мужчины определяется соответственно как удвоенный возраст девушки минус 14. Данная программа должна выдать необходимые сведения для всех желающих, количество которых заранее не известно.

5. В книжном магазине ведется обслуживание покупателей с помощью ЭВМ. ЭВМ должна запрашивать стоимость книг, выбранных покупателем, печатать суммарную стоимость, запрашивать сумму денег, внесенных покупателем, определять причитающуюся покупателю сдачу, если денег внесено больше или выдавать сообщение о недостаточности внесенной суммы, а после расчета с покупателем выдавать "Спасибо". В конце рабочего дня программа должна определить общую сумму, полученную от продажи книг.

6. В ЭВМ по очереди вводятся фамилии спортсменов и их результаты в соревнованиях по прыжкам в длину. Число участников произвольно. Выдавать на печать лучший результат после выступления очередного спортсмена. После окончания соревнования выдать итоговую таблицу, куда войдут лучшие десять спортсменов (первым в таблице должен быть спортсмен с лучшими показателями, а ниже в порядке убывания результатов).

7. Стоимость платья зависит от материала, а также от фасона. Предполагается, что в ателье имеется 2 вида материала: шерсть и шелк (цена вводится). На пошив платья требуется 3 м материала. Стоимость пошива базового фасона (с минимальной отделкой) вводится. (Базовый фасон прилагается). За дополнительные детали отделки взимается дополнительная плата. Так

- 1 пуговица - 10 т
- воротник -50т
- 1 складка -15т
- 1 карман -55т
- 1 отстрочка -80т
- 1 манжета - 50т



Составить программу, определяющую стоимость заказываемого платья, а в конце рабочего дня общую сумму заказа.

8. Написать программу, обрабатывающую ход сдачи экзаменационной сессии студентами 1-го курса N-го факультета. На факультете имеется заданное число специальностей.. В группе максимум 25 человек. Каждый студент на каждом курсе должен сдать по 3 экзамена. Программа должна выдавать средний балл по каждой группе и по факультету в целом

9. Написать программу, обрабатывающую ход сдачи экзаменационной сессии студентами 1-го курса N-го факультета. На факультете имеется заданное число специальностей.. В группе максимум 25 человек. Каждый студент на каждом курсе должен сдать по 3 экзамена. Программа должна выдавать средний балл по каждому экзамену и количество студентов, не сдавших 1, 2 и 3 экзамена по факультету в целом

дата расчета _____	ср. балл	ср. балл	ср. балл
. . . .	1 экз.	2 экз.....3 экз.	
	_____	_____	_____
Кол-во студентов не сдавших			
. . . .	1 экз.	2 экз.....3 экз.	
	_____	_____	_____

10. Написать программу, обрабатывающую ход сдачи экзаменационной сессии студентами 1-го курса N-го факультета. На факультете имеется заданное число специальностей.. В группе максимум 25 человек. Каждый студент на каждом курсе должен сдать по 3 экзамена. Программа должна выдавать

- количество студентов, сдавших экзамены на отлично;
- количество студентов, не сдавших хотя бы один экзамен;

11. Составить программу помощника кассира в универсальном магазине. ЭВМ должна запрашивать цену товара и его количество, подсчитывать суммарную стоимость купленных товаров, запрашивать сумму денег, внесенных покупателем и определять причитающуюся ему сдачу. Если сумма, внесенная покупателем мала, то выдавать соответствующее сообщение. Прекращение работы программы должно произойти при вводе суммы за покупку равной или меньшей 0.

12. Составить программу, подсчитывающую число посещений в поликлинике врачей специалистов (невропатолога, окулиста и хирурга). В конце дня выдавать итоговое сообщение. В начале работы программы должны вводиться дата, которая будет фигурировать в итоговом сообщении. Использовать множественный выбор. Признаком прекращения работы программы принять ввод шифра специалиста равного "\*\*\*\*".

13. Пассажирский самолет может поднять груз общим весом 30т. Составить программу для определения веса почтового груза, который можно поместить в самолет после посадки пассажиров и загрузки их багажа. Во время регистрации пассажиров ЭВМ должна подсчитывать количество пассажиров (условный вес одного человека -100 кг) и суммировать вес багажа. Признаком прекращения работы программы принять ввод багажа нулевого или отрицательного веса.

14. Составить программу для определения подходящего возраста кандидатуры для вступления в брак, используя следующее соображение: возраст девушки равен половине возраста мужчины плюс 7, возраст мужчины определяется соответственно как удвоенный возраст девушки минус 14. Данная программа должна выдать необходимые сведения для всех желающих, количество которых заранее не известно. Признаком прекращения работы программы принять ввод фамилии равной "\*\*\*\*".

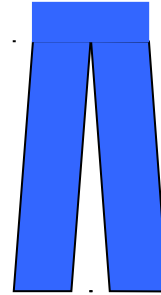
15. В книжном магазине ведется обслуживание покупателей с помощью ЭВМ. ЭВМ должна запрашивать стоимость книг, выбранных покупателем, печатать суммарную стоимость, запрашивать сумму денег, внесенных покупателем, определять причитающуюся покупателю сдачу, если денег внесено больше или выдавать сообщение о недостаточности внесенной суммы, а после расчета с покупателем выдавать "Спасибо". В конце рабочего дня программа должна определить общую сумму, полученную от продажи книг. Признаком прекращения работы программы принять ввод Признаком прекращения работы программы принять ввод отрицательной или нулевой цены.

16. В ЭВМ по очереди вводятся фамилии спортсменов и их результаты в соревнованиях по прыжкам в длину. Число участников произвольно. Выдавать на печать лучший результат после выступления очередного спортсмена. После окончания соревнования выдать итоговую таблицу, куда войдут лучшие десять спортсменов (первым в таблице должен быть спортсмен с лучшими показателями, а ниже в порядке убывания

результатов). Признаком прекращения работы программы принять ввод фамилии равной "\*\*\*\*".

17. Стоимость брюк зависит от материала, а также от фасона. Предполагается, что в ателье имеется 2 вида материала: шерсть и шелк (цена вводится). На пошив брюк требуется 1,3 м материала. Стоимость пошива базового фасона (с минимальной отделкой) вводится. (Базовый фасон прилагается). За дополнительные детали отделки взимается дополнительная плата. Так

1 пуговица - 10 т  
 пояс -50т  
 1 складка -15т  
 1 карман -55т  
 1 отстрочка -80т  
 1 манжета - 50т



Составить программу, определяющую стоимость заказываемых брюк, а в конце рабочего дня общую сумму заказа по брюкам.

18. Написать программу, обрабатывающую ход сдачи экзаменационной сессии студентами 1-го курса N-го факультета. На факультете имеется заданное число специальностей.. В группе максимум 25 человек. Каждый студент на каждом курсе должен сдать по 3 экзамена. Программа должна выдавать средний балл по каждой группе и по факультету в целом.

19. Написать программу, обрабатывающую ход сдачи экзаменационной сессии студентами 1-го курса N-го факультета. На факультете имеется заданное число специальностей.. В группе максимум 25 человек. Каждый студент на каждом курсе должен сдать по 3 экзамена. Программа должна выдавать средний балл по каждому экзамену и количество студентов, не сдавших 1, 2 и 3 экзамена по факультету в целом

дата расчета _____	ср. балл	ср. балл	ср. балл
. . . .	1 экз.	2 экз.....	3 экз.
	_____	_____	_____
Кол-во студентов не сдавших			
. . . .	1 экз.	2 экз.....	3 экз.
	_____	_____	_____

20 Написать программу, обрабатывающую ход сдачи экзаменационной сессии студентами 1-го курса N-го факультета. На факультете имеется заданное число специальностей.. В группе максимум 25 человек. Каждый

студент на каждом курсе должен сдать по 3 экзамена. Программа должна выдавать:

- количество студентов, сдавших экзамены на отлично;
- количество студентов, не сдавших хотя бы один экзамен.

## 5 Разработка программы с использованием массивов (лабораторная работа №5)

**Цель работы:** освоить применение массивов (**ARRAY**) и работу с ними, конструирование собственных типов при описании массивов.

### 5.1 Теоретические сведения

Тип массив представляет собой фиксированное количество однородных компонент, снабженных индексами. Он может быть одномерным и многомерным. Чтобы задать тип переменной "массив", используется зарезервированное слово **ARRAY**, после которого следует указать в квадратных скобках количество и пределы изменения индексов, и далее после слова **OF**- тип компонент.

```
Var c22: array[1..3,1..4] of integer;
```

Выше приведен пример описания двумерного массива **c22**, имеющего три строки и четыре столбца. Элементы этого массива - целые числа.

Паскаль позволяет определить тип массивам, а затем пользоваться этим типом внутри данного программного блока. Определение типа. Определение типа выполняется с помощью описания, которое начинается со служебного слова **type** и содержит описание типа. Каждое определение типа состоит из пары : имя типа и собственно тип . Имя типа обозначается идентификатором, сам тип описывается по правилам, приведенным выше, разделителем между ними является знак **"="**. Например:

Type

```
Me_arr = array[ 1..8] of real;
```

{тип- одномерный массив из 8-ми вещественных чисел}

```
Matrix = array[1..3,1..2] of integer;
```

{тип - двумерный массив из 3-х строк и 2-х столбцов целых чисел }.

Введя тип-массив, можно затем задать переменные или типизированные константы этого типа.

При задании значений константе-массиву компоненты указываются в круглых скобках и разделяются запятыми, причем, если массив многомерный, внешние круглые скобки соответствуют левому индексу, вложенные скобки- следующему индексу и т.д.

Так для введенных выше типов можно задать следующие переменные и типизированные константы:

Var

```
M1, m2: my_arr;
```

```
mas: Matrix
```



const

c22: Matrix = ((3,5),(0,65),(123,0));

Доступ к компонентам массива осуществляется указанием имени массива, за которым в квадратных скобках помещается значение индексов. В общем случае каждый индекс может быть задан выражением целого типа, например:

M1 [ 2 ], c22[ k, l ], c22[ 1, m ], mas [ j, j ].

## 5.2 Указания к выполнению лабораторной работы

Написать и отладить программу с применением массивов (ARRAY) данных. Предусмотреть описание типа массива и применить типизированные константы для присвоения начальных значений.

### Варианты заданий

1. Определить средний рост юношей и средний рост девушек одной группы. В группе задаваемое количество человек. Напечатать фамилии юношей и девушек выше среднего роста.

2. Определить средний балл каждого студента в группе и группы в целом. Напечатать фамилии студентов со средним баллом более 4,75. Напечатать фамилии студентов, имеющих средний балл выше среднего балла по группе.

3. В ЭВМ по очереди поступают результаты соревнований по плаванию, в которых участвует N спортсменов. Выдавать на печать лучший результат и фамилию рекорсмена после ввода результата очередного спортсмена.

4. Имеются координаты n точек, хранящиеся в таблице. Определить, сколько из них попадет в круг радиусом R с центром в точке (0,0). Отсортировать таблицу таким образом, что бы в ее начале находились точки, находящиеся ближе к точке (0,0), а к концу таблицы были помещены точки, наиболее отдаленные от начала координат.

5. Ученикам 1 класса назначается дополнительно стакан молока (200 мл), если их вес меньше 30 кг. Определить, сколько молока потребуется для одного класса, состоящего из n учеников. Вес каждого ученика и его фамилия заносится в ЭВМ. Напечатать список учеников, нуждающихся в дополнительном питании.

6. В ЭВМ вводятся фамилии и рост учеников 8 класса. Для определения кандидатов в баскетбольную команду вывести фамилии тех учеников, рост которых больше 170 см. В классе n учеников. Вычислить средний рост кандидатов в сборную.

7. В ЭВМ вводятся данные о температуре воздуха и количестве осадков в течение месяца. Определить среднюю температуру и среднее количество осадков. Вывести те числа месяца, когда температура воздуха была на 10% выше средней. Вывести те числа месяца, когда осадков была на 20% меньше среднего количества.

8. В ЭВМ вводятся данные о ценах на зерно в 30, торгующих зерном странах и доставки одной тонны зерна. Вывести наименование страны, в которой выгоднее всего закупить 1000 т зерна.

9. В ЭВМ вводятся данные о ценах на сахар каждой из 20 стран, торгующих сахаром, определить среднюю стоимость сахара на рынке и вывести список стран, продающих сахар ниже средней стоимости.

10. В ЭВМ вводятся данные о средней температуре, среднем количестве осадков и среднем числе солнечных дней в году 20-ти областей страны. Определить область с наиболее благоприятным климатом для строительства детского санатория.

11. Определить максимальное значение в таблице чисел. Количество строк и столбцов вводится. Затем заполняется сама таблица. Вывести номер строки, столбца и само максимальное значение.

12. Определить три максимальных значения в таблице чисел. Количество строк 4, столбцов - 6. Вывести полученные значения, а так же места их нахождения в таблице.

13. Определить минимальное значение в таблице чисел. Количество строк и столбцов вводится. Затем заполняется сама таблица. Вывести номер строки, столбца и само максимальное значение.

14. Определить три минимальных значения в таблице чисел. Количество строк 5, столбцов - 10. Вывести полученные значения, а так же места их нахождения в таблице.

15. Определить максимальные значения в каждой строке таблицы чисел. Количество строк и столбцов: 5 и 6. Вывести номер столбца для каждой строки, где находится максимальное значение и само значение.

16. Определить минимальные значения в каждом столбце таблицы чисел. Количество строк и столбцов: 10 и 8. Вывести номер строки для каждого столбца, где находится минимальное значение и само значение.

17. Подсчитать количество чисел в таблице меньше вводимого числа. Таблица имеет 5 строк и 4 столбца.

18. Определить минимальное значение в таблице чисел. Количество строк 5, столбцов - 10. Вывести полученное значение, а так же место его нахождения в таблице

19. Имеются данные о ценах на нефть и стоимость доставки нефти каждой из 20 стран, торгующих нефтью, определить среднюю стоимость нефти на рынке и вывести список стран, продающих нефть ниже средней стоимости. Вывести первые 5 стран, в которых выгоднее закупать нефть( учесть доставку).

20. Имеются следующие данные о сотрудниках фирмы: ФИО, год рождения, образование (закодировать с помощью одного символа), семейное положение, оклад. Вывести список сотрудников заданного образования, семейного положения, с окладом не менее заданной суммы.

## 6 Разработка программы с использованием типа данных “запись”(RECORD) (лабораторная работа №6)

**Цель работы:** освоить конструирование различных собственных типов записей и их применение, научить разрабатывать алгоритмы на основе обработки записей.

### 6.1 Теоретические сведения

Тип запись включает ряд компонент, называемых полями, которые могут быть разных типов. При задании типа - запись после зарезервированного слова RECORD следует перечислить все поля типа - записи с указанием через двоеточие их типов и завершить задание типа словом END. Поля отделяются друг от друга точкой с запятой. Количество полей записи может быть любым. Например:

```
Type
Complex = record
Re :real;
Im: real
End;
Person = record
Fam: string;
Age: integer;
Married : boolean
End;
```

После введения типа запись можно задать переменные этого типа и типизированные константы.

При задании значений константе ее поля записываются в круглых скобках через точку с запятой. Для задания каждого поля сначала указывается имя поля, а затем через двоеточие - его значение. Так для введенных выше типов, можно задать следующие переменные и константы:

```
Var
X,y,z: complex;
Const
Ivanov : person = (Fam: 'Иванов Ф. П.'; Age: 40; Married: True);
```

Доступ к полям записи осуществляется указанием имени переменной (константы) и имени поля, записываемого через точку, например:

```
Ivanov. Age      X.Re      Y.Im
```

Для того, чтобы не выписывать каждый раз имя записи при обращении к ее полям, можно использовать оператор над записями WITH. Его структура выглядит следующим образом:

```
With <имя записи> do <оператор >
```

В этом случае внутри оператора можно указывать только поле записи. Например:

```
With x do
Begin
Re:= 1.345;
Im:= - 0.65
End;
```

## 6.2 Указания к выполнению лабораторной работы

Написать и отладить программу, в которой разнотипные данные, обрабатываемые параллельно, должны быть объединены в одну структуру, то есть необходимо описать собственный тип "запись" и затем описать массив переменных с этим типом.

### Варианты заданий

1. Ученикам 1 класса назначается дополнительно стакан молока (200 мл), если их вес меньше 30 кг. Определить, сколько молока потребуется для одного класса, состоящего из  $n$  учеников. Вес каждого ученика и его фамилия заносится в ЭВМ. Напечатать список учеников, нуждающихся в дополнительном питании.

2. Имеются фамилии, пол и рост учеников 11-х классов. Для определения кандидатов в баскетбольную команду вывести фамилии тех учениц, рост которых не ниже 170 см, и учеников, рост которых не ниже 190 см. В классе  $n$  учеников. Вычислить средний рост кандидатов в сборную.

3. Имеются данные о температуре воздуха и количестве осадков в течение месяца. Определить среднюю температуру и среднее количество осадков. Вывести те числа месяца, когда температура воздуха была на 10% выше средней. Вывести те числа месяца, когда осадков была на 20% меньше среднего количества.

4. Имеются данные о средней температуре, среднем количестве осадков и среднем числе солнечных дней в году 20-ти областей страны. Определить область с наиболее благоприятным климатом для строительства детского санатория.

5. Определить средний рост юношей и средний рост девушек одной группы. В группе  $n$  человек. Напечатать фамилии юношей и фамилии девушек выше среднего роста.

6. Определить средний балл каждого студента в группе и группы в целом. Напечатать фамилии студентов со средним баллом более 4,75.

Напечатать фамилии студентов, имеющих средний балл выше среднего балла по группе.

7. Имеются результаты соревнований по плаванию, в которых участвует N спортсменов. Выдать на печать лучший результат и фамилию рекордсмена после ввода результата последнего спортсмена. Выдать на печать фамилии 10-ти спортсменов, показавшие лучшие результаты.

8. Имеются данные о ценах на зерно в 30, торгующих зерном странах и доставки одной тонны зерна. Вывести наименование страны, в которой выгоднее всего закупить зерно (учесть стоимость доставки).

9. Имеются следующие данные о сотрудниках фирмы: ФИО, год рождения, образование (закодировать с помощью одного символа), оклад. Вывести список сотрудников заданного возраста, с окладом не менее заданной суммы.

10. Имеются данные о ценах на сахар и стоимость доставки 1 тонны сахара каждой из 20 стран, торгующих сахаром, определить среднюю стоимость сахара на рынке и вывести список стран, продающих сахар ниже средней стоимости. Вывести первые 5 стран, в которых выгоднее закупать сахар( учесть доставку).

11. Имеются данные о температуре воздуха и количестве осадков в течение месяца. Определить среднюю температуру и среднее количество осадков. Вывести те числа месяца, когда температура воздуха была на 10% выше средней. Вывести те числа месяца, когда осадков была на 20% меньше среднего количества.

12. Имеются данные о средней температуре, среднем количестве осадков и среднем числе солнечных дней в году 20-ти областей страны. Определить область с наиболее благоприятным климатом для строительства детского санатория.

13. Определить средний рост юношей и средний рост девушек одной группы. В группе ..N человек. Напечатать фамилии юношей и фамилии девушек выше среднего роста.

14. Определить средний балл каждого студента в группе и группы в целом. Напечатать фамилии студентов со средним баллом более 4,75. Напечатать фамилии студентов, имеющих средний балл выше среднего балла по группе.

15. Имеются результаты соревнований по легкой атлетике, в которых участвует N спортсменов. Выдать на печать лучший результат по каждому виду (бег, прыжки и т. д.) и фамилию рекордсмена после ввода результата последнего спортсмена. Выдать на печать фамилии 10-ти спортсменов, показавшие лучшие результаты.

16. Имеются данные о ценах 30 торгующих хлопком странах и доставки одной тонны хлопка. Вывести наименование страны, в которой выгоднее всего закупить хлопок (учесть стоимость доставки).

17. Имеются следующие данные о сотрудниках фирмы: ФИО, год рождения, образование (закодировать с помощью одного символа), семейное

положение, оклад. Вывести список сотрудников заданного возраста, имеющих семью, с окладом не менее заданной суммы.

18. Имеются данные о ценах на сахар и стоимость доставки 1 тонны сахара каждой из 30 стран, торгующих сахаром, определить среднюю стоимость сахара на рынке и вывести список стран, продающих сахар ниже средней на рынке стоимости.

19. Имеются данные о ценах на нефть и стоимость доставки нефти каждой из 20 стран, торгующих нефтью, определить среднюю стоимость нефти на рынке и вывести список стран, продающих нефть ниже средней стоимости. Вывести первые 5 стран, в которых выгоднее закупать нефть( учесть доставку).

20. Имеются следующие данные о сотрудниках фирмы: ФИО, год рождения, образование (закодировать с помощью одного символа), семейное положение, оклад. Вывести список сотрудников заданного образования, семейного положения, с окладом не менее заданной суммы.

## 7 Разработка программы, обрабатывающей файловые данные (лабораторная работа №7)

**Цель работы:** освоить работу с файлами в среде Turbo Pascal: создание, редактирование файлов, вывод информации, хранящейся в файлах, обработку данных из файлов.

### 7.1 Теоретические сведения

#### 7.1.1 Файлы в Turbo Pascal

В языке Паскаль имеется возможность работы с файловыми данными.

Перед тем, как осуществлять ввод-вывод, файловая переменная должна быть связана с конкретным внешним файлом с помощью процедуры:

Assign(F, Name), где F - файловая переменная, Name- текстовая переменная или константа, представляющая собой полное имя файла.

Этим файлом может быть файл на магнитном диске или ленте, а так же устройство вычислительного комплекса (клавиатура, экран дисплея, принтер и т.д.). Затем файл должен быть открыт для чтения и/или записи. После этого можно осуществлять организацию ввода-вывода.

В Турбо Паскале имеются 3 типа файлов: текстовые (тип Text), типизированные (file of <тип>) и файлы без типа (file). Работа с этими категориями файлов отличается друг от друга. Описание файлов этих типов соответственно:

```
Var f1:text;
    F2,f3:file of real;
    F4: file;
```

Обычно все файлы считаются файлами последовательного доступа. Однако для типизированного файла и файла без типа можно с помощью процедуры Seek(F,Nom) ( F - файловая переменная, а Nom -номер записи в данном файле) установить режим произвольного доступа.

После работы с файлом он, как правило, должен быть закрыт процедурой Close(F). Это требование обязательно должно быть выполнено для файла, в который производилась запись информации.

При работе с файлами автоматически проверяются ошибки ввода - вывода. Если проверка ошибки включена (по умолчанию или с помощью ключа компиляции{\$I+}),то при возникновении ошибки выполнение программы автоматически завершается, если же проверка отключена ( с помощью ключа компиляции{\$I-}), программа продолжает выполняться, и можно будет обработать ошибку ввода-вывода с помощью функции IOResult. При отсутствии ошибок ввода-вывода системная переменная IOResult равна нулю, в противном случае ее значение равно коду ошибки ввода-вывода.

С помощью процедуры Rewrite(F) открывается новый файл для записи. Если файл с таким же именем уже существовал, то он будет уничтожен и откроется новый.

С помощью процедуры Reset(F) открывается файл для чтения или записи информации. При этом указатель настраивается на первую запись.

Таблица 5 – Процедуры для работы с файлами всех типов

Процедуры	Действие	Примечание
<b>Assign(F,Name)</b>	Связь файлов. перем с полным именем файла на диске	
<b>ChDir(Path)</b>	Устанавливает каталог Path	
<b>Close(F)</b>	Закрывает файл	
<b>Erase(F)</b>	Уничтожает файл F	
<b>GetDir(N,S)</b>	В S пом. имя текущ. каталога	N=0-тек.диск,1-A, 2- B и т.д.
<b>MkDir(Path)</b>	Создает каталог Path	
<b>Rename(F,New)</b>	Переименовывает файл F	New-новое имя
<b>Rmdir(Path)</b>	Удаляет пустой каталог Path	

Таблица 6 – Функции для работы с файлами всех типов

Функции	Действие	Примечание
<b>Eof(F)</b>	Проверка на конец файла	Результат True или False
<b>IOResult</b>	Возв. номер ошибки ввода-вывода	0- ошибки нет

Обработку ошибок ввода-вывода внутри вашей программы можно осуществить путем вставки ключа компиляции {\$I-}. Например, производится попытка чтения несуществующего файла:

```

Var f : text;
Begin assign(f,'c:\User\dan.txt');
{$I-}
reset(f); if IOResult<> 0 then rewrite(f);
{$I+}.

```

### 7.1.2 Текстовые файлы

Текстовый файл представляет собой совокупность символов, разделенных на строки, причем, в конце каждой строки стоит признак конца строки.

Особенностью работы с текстовыми файлами является то, что параметры, значения которых вводятся или выводятся в этот файл могут быть любых простых типов, а именно: целые, вещественные, символьные, строковые.



Файл типа Text может быть открыт либо для чтения процедурой Reset(F), либо для записи как новый, процедурой Rewrite(F), либо для дозаписи в конец файла - процедурой Append(F).

Таблица 7 – Процедуры для работы с текстовыми файлами

Процедуры	Действие	Примечание
<b>Append(F)</b>	Открывает для добавления в конец	
<b>Read(F,&lt;спис.вв&gt;)</b>	Чтение данных из файла в список ввода	
<b>Readln(F,&lt;спис.вв&gt;)</b>	Чтение строки из файла в список ввода	непрочит. данн. опуск.
<b>Write(F,&lt;сис.выв&gt;)</b>	запись значений из списка вывода	
<b>Writeln(F,&lt;сис.выв&gt;)</b>	то же, что Write, + признак конца строки	

Таблица 8 – Функции для работы с текстовыми файлами

Функции	Действие	Примечание
<b>Eoln(F)</b>	Проверка на конец строки файла	Результат True или False
<b>SeekEoln(F)</b>	Переход на конец строки	
<b>SeekEof(F)</b>	Переход на конец файла	пробелы и табуляц. пропускает

Ввод-вывод информации при работе с текстовым файлом полностью совпадает с вводом с клавиатуры и выводом на экран дисплея (при этом нужно иметь ввиду, что символы перевода строки и возврата "каретки" на экране не отображаются), следовательно можно применять при работе с файлом этого типа форматирование вывода, а так же вывод в файл текстовых комментариев.

### 7.1.3 Типизированные файлы

При работе с типизированными файлами используются следующие дополнительные процедуры и функции.

Таблица 9 – Процедуры для работы с типизированными файлами

Процедуры	Действие	Примечание
<b>Read(F,&lt;спис.вв&gt;)</b>	Чтение данных из файла в список ввода	
<b>Seek(F,Num)</b>	Переход на запись с номером Num	нумерация с нуля
<b>Write(F,&lt;сис.выв&gt;)</b>	запись значений из списка вывода	
<b>Truncate(F)</b>	Удаление части файла, с текущей позиц.	

Таблица 10 – Функции для работы с типизированными файлами

Функции	Действие	Примечание
<b>FilePos(F)</b>	Возв. номер тек. записи файла	
<b>FileSize(F)</b>	Возв. количество записей файла	

Пример: Записать в типизированный файл информацию о сотрудниках фирмы(ФИО, год рождения, пол), вводимую с клавиатуры.

```

Program nnn;
Type sotr=record
Fio: string;
Gr:intrger;
Pol:char
End;
Var
f1:file of sotr;
rab: sotr;
label m_beg,m_end;
begin assign(f1,'c:\user\worker.dan');
{1}  Writeln('Ввод информации о сотрудниках');
{2}  m_beg:
{3}  with rab do begin
{4}  Write('Фамилия, инициалы:');readln(fio);
{5}  if fio='***' then goto m_end;
{6}  Write('Год рождения:');read(Gr); Write('пол:');Readln(Pol);
{7}  End;
{8}  Write(f1, rab); goto m_beg;
{9}  m_end: close(f1);
end.

```

Чтобы добавить записи в конец типизированного файла, необходимо сначала прочитать все записи до конца файла и только потом осуществлять вывод новой информации. Так для задачи, описанной выше дозапись информации может быть выполнена одним из двух перечисленных ниже способов:

```

1)
Reset(f1);
While not eof(f1) do read(f1);
2)
Reset(f1);
kol:=filesize(f1); seek(f1, kol+1);

```

Далее должен идти фрагмент программы, помеченный в примере выше цифрами в фигурных скобках от {1} до {9}.

## 7.2 Указания к выполнению лабораторной работы

Выбрать задание. Написать и отладить программу, в которой создается и обрабатывается:

- типизированный файл, разнотипные данные, обрабатываемые параллельно, должны быть объединены в одну структуру, то есть необходимо описать собственный тип "запись" и затем описать файл с этим типом. Организовать добавление информации в существующий файл. Выводить информацию из файла на экран постранично ("поэкранно"- для продолжения вывода необходимо нажимать клавишу "Enter"). Обработку информации из

файла выполнить, решив задачу из лабораторной работы № 6 для соответствующего варианта;

- текстовый файл (задание то же, что и для типизированного файла)

#### Варианты заданий

1. Создать файл, содержащий следующую информацию в каждой записи: ФИО ученика и его вес.
2. Создать файл, содержащий следующую информацию в каждой записи: фамилии, пол и рост учеников 11-х классов.
3. Создать файл, содержащий следующую информацию в каждой записи: данные о температуре воздуха и количестве осадков в течение месяца.
4. Создать файл, содержащий следующую информацию в каждой записи: данные о средней температуре, среднем количестве осадков и среднем числе солнечных дней в году 20-ти областей страны.
5. Создать файл, содержащий следующую информацию в каждой записи: ФИО, пол, рост, название группы.
6. Создать файл, содержащий следующую информацию в каждой записи: ФИО, оценка1, оценка2, оценка3, оценка4, средний балл, название группы.
7. Создать файл, содержащий следующую информацию в каждой записи: ФИО, пол, результаты соревнований по плаванию.
8. Создать файл, содержащий следующую информацию в каждой записи: название страны, торгующей зерном в этой стране, цена на зерно в этой стране, цена доставки зерна из этой страны.
9. Создать файл, в котором имеются следующие данные о сотрудниках фирмы: ФИО, год рождения, образование (закодировать с помощью одного символа), оклад.
10. Создать файл, в котором имеются названия стран, торгующих сахаром, данные о ценах на сахар и стоимость доставки 1 тонны сахара.
11. Создать файл, в котором имеются следующие данные : температура воздуха и количество осадков, дата.
12. Создать файл, в котором имеются следующие данные: средняя температура, среднее количество осадков, среднее число солнечных дней в году областей страны.
13. Создать файл, в котором имеются следующие данные: ФИО, пол, рост.
14. Создать файл, в котором имеются следующие данные: фамилии студентов, оценки по четырем экзаменам.
15. Создать файл, в котором имеются следующие данные: ФИО, вид соревнования по легкой атлетике, результаты.
16. Создать файл, в котором имеются следующие данные: цены на хлопок, страна, цена доставки одной тонны хлопка.

17. Создать файл, в котором имеются следующие данные: ФИО, год рождения, образование (закодировать с помощью одного символа), семейное положение, оклад..

18. Создать файл, в котором имеются следующие данные: наименование страны, цена на сахар и стоимость доставки 1 тонны.

19. Создать файл, в котором имеются следующие данные: наименование страны, цена на нефть и стоимость доставки нефти.

20. Создать файл, в котором имеются следующие данные о сотрудниках фирмы: ФИО, год рождения, образование (закодировать с помощью одного символа), семейное положение, оклад.

## 8 Разработка программы с применением процедур и функций (лабораторная работа №8)

**Цель работы:** научить студентов разрабатывать программы модульной структуры, где в качестве основной единицы выступают процедуры и функции, программа также должна иметь меню.

### 8.1 Теоретические сведения

Профессиональный подход к разработке программ предполагает оформление в виде подпрограмм любого достаточно самостоятельного программного фрагмента. Понятие подпрограммы является основным во всех структурных языках программирования. В языке Паскаль имеется две разновидности подпрограмм - процедуры и функции.

Структура любой подпрограммы аналогична структуре всей программы. Подпрограмма должна быть описана до того, как она будет использована в программе или другой подпрограмме.

Все параметры, которые использует подпрограмма, можно разбить на две категории: локальные и глобальные. Локальные параметры описываются внутри подпрограммы и доступны только ей самой. Глобальные параметры объявляются в основной программе и доступны как программе, так и всем подпрограммам, которые описаны после этих параметров.

Подпрограмма может использовать глобальные параметры двумя способами: 1- обращаясь к параметру по его имени или, 2- используя механизм формальных параметров.

Необходимо помнить, что при использовании одинаковых идентификаторов при описании глобальных и локальных параметров, локальные параметры "заслоняют" глобальные.

Подпрограмма-процедура начинается с заголовка:

Procedure <имя >[(список формальных параметров)];

Квадратные скобки означают, что список формальных параметров может отсутствовать. За заголовком могут идти те же разделы, что и в основной программе. В отличие от основной программы процедура завершается не точкой, а запятой.

```
Program mmm;
```

```
Var m: array[1..3,1..3] of char;
```

```
{тело процедуры }
```

```
Procedure dan;
```

```
Var i,j:byte;
```

```
Begin
```

```
  writeln('Введите последовательно цифры от 1 до 9');
```

```
  For i:=1 to 3 do
```

```
    For j:=1 to 3 do read (m[i,j]);
```

```
  End;
```

```
{тело основной программы }
```

```
Begin. . . . . ; dan; . . .
```

End.

Для вызова процедуры следует написать оператор, состоящий из имени процедуры и списка фактических параметров, если они имеются. В примере, приведенном выше, процедура вызывается просто по своему имени: `dan`. Процедура выполняет ввод символов в массив `m`.

Подпрограмма-функция предназначена для вычисления какого-либо значения. У этой подпрограммы два основных отличия от процедуры.

1 -подпрограмма-функция начинается с заголовка:

```
function <имя >[(список форм. параметров)]:<тип результата>;
```

2 - в теле функции перед выходом должен стоять оператор, присваивающий значение имени функции: `<имя>:= выражение;`

Пример: функция вычисления факториала.

```
function fact (n : integer): longint;
```

```
var f: longint;
```

```
i : integer;
```

```
begin
```

```
f:= n;
```

```
for i:= n-1 downto 2 do f:=f*i;
```

```
fact:=f;
```

```
end;
```

Для вызова функции из основной программы или другой подпрограммы следует указать в выражении имя функции со списком фактических параметров(если они имеются), например:

```
G:=fact(25);
```

## 8.2 Указания к выполнению лабораторной работы

Разработать программу на основе результатов предыдущей лабораторной работы №7. Программа должна работать с помощью меню. Меню должно иметь следующие пункты: "1-Ввод информации", "2-Вывод информации", "3-Расчет", "4-Выход". Действия, провоцируемые выбором всех пунктов меню кроме пункта "Выход", должны быть оформлены в виде процедур. Реализация выбора пунктов меню должна быть выполнена с помощью оператора выбора CASE ... OF.

## 9 Разработка программ, обрабатывающих строковые (STRING) и символьные (CHAR) данные (лабораторная работа №9)

**Цель работы:** научить студентов разрабатывать программы, обрабатывающие строковые данные, использовать функции по работе со строковыми данными, применять циклические алгоритмы при обработке строк.

### 9.1 Теоретические сведения

Строковые типы являются одним из наиболее полезных и часто используемых типов. Строковый тип обобщает понятие символьных массивов, позволяя динамически изменять длину строки.

Для определения строкового типа используется служебное слово `string`, вслед за которым в квадратных скобках указывается максимальная длина строки, например:

```
Var nw : string [ 50 ] ;
```

В приведенном примере переменная `nw` в качестве своего значения может иметь любую последовательность символов произвольной длины в пределах от 0 до 50 символов. Указание длины строковой переменной при ее описании может быть опущено. В этом случае Паскаль подразумевает длину такой переменной в пределах от 0 до 255 символов. Таким образом, следующие описания эквивалентны:

```
Var z : string;
    G1:sstring[255];
    .....D,f:char;
```

Переменные типа `char` могут содержать только один символ, то есть длина такой переменной один байт.

Строку можно рассматривать как массив символов. То есть, обращение к символу, включенному в строку, может выполняться по имени строковой переменной с указанием в квадратных скобках места данного символа, который может выражаться константой или переменной целого типа, например:

```
z[ 51 ] , G1[ i ]
```

Для строковых переменных определена операция слияния (конкатенации), обозначаемая знаком "+". Строковые переменные можно сравнивать, причем более короткая строка всегда меньше более длинной, если же длины равны, то происходит поэлементное лексикографическое сравнение.

Для перевода строковых значений в численный формат существует процедура:

```
Val(s,v,Code);
```

где `v` - выражение численного типа, вещественного или целого, `s` - переменная типа `string`, куда помещается преобразованная величина, `Code` - номер ошибочного символа, если произошла ошибка при перекодировании, и 0- если ошибки не возникло.

Для перевода численных значений в строковый формат существует процедура:

```
Str(x[:m[:n]],s);
```

где x - выражение численного типа, вещественного или целого с возможным указанием формата, аналогично процедуре Write, s- переменная типа string , куда помещается преобразованная величина.

Для работы со строковыми и символьными переменными в Паскале имеется ряд процедур и функций. Так для определения длины строковой переменной можно использовать функцию Length (имя переменной), а процедура Insert(строка\_источник,строка-результат, поз) помещает значение строки-первого аргумента в "строку-результат", начиная с позиции "поз".

Противоположная операция - удаление может быть выполнена с помощью процедуры Delete(строка, поз, колич), которая удаляет из строки"строка", начиная с позиции "поз", "колич" символов.

Функция Copy(S, Index, Count) возвращает строку, созданную путем копирования из исходной строки S, начиная с позиции Index, Count символов.

Функция Pos(Substr, S) отыскивает в строке S последовательность символов Substr и возвращает номер символа, с которого начинается искомая последовательность, а если поиск оказался безуспешным, то возвращается значение 0.

Для слияния строковых величин может использоваться операция "+" или функция

```
Concat(S1, S2, [ , S3, ... Sn] ),
```

где S1, S2,...- строки. Результирующая строка усекается до 255 символов, если она оказалась длинее.

## 9.2 Указания к выполнению лабораторной работы

Разработать программу, производящую ввод информации в строковом виде с переводом в числовой формат необходимых строковых данных.

Использовать сравнение для поиска информации по указанному критерию.

Задание взять из лабораторной работы №7.



1. Поляков Д.Б., Круглов И.Ю. Программирование в среде Турбо Паскаль.- М.:Унитех, 1991, 358с.: ил.
2. Зуев Е.А. Язык программирования Turbo Pascal 6.0.- М.: Унитех, 1992. - 298с.: ил.
3. Епанешников А., Епанешников В., Программирование в среде Turbo Pascal 7.0. - 3-е изд. Стер. - М.: "ДИАЛОГ-МИФИ", 1995. - 288с.
4. Марченко А.И., Марченко Л.А. Программирование в среде Turbo Pascal 7.0. – К.: Век +, 2000, 464с.: ил.