

**Задания для студентов по дисциплине «Информатика»**  
**Специальность 35.02.08 Электрфикация и автоматизация сельского хозяйства**  
(группа 1Д)

*Задания выполнять в тетради. Выполненные задания сфотографировать и отправить на адрес электронной почты [admin@ptmecx.ru](mailto:admin@ptmecx.ru) или VK <https://vk.com/id190930822> в личные сообщения*

**Занятие №25**

**Дата:** 30 ноября 2020г.

**Инструкционно-технологическая карта  
практического занятия №8**

**ТЕМА:** Языки программирования.

**НАИМЕНОВАНИЕ РАБОТЫ:** Разработка линейного алгоритма (программы)

**ЦЕЛИ РАБОТЫ:** Знакомство со средой программирования Pascal; изучить структуру программы, стандартные функции, оператор присваивания и процедуры ввода-вывода; научиться создавать программы на языке Pascal с использованием стандартных функций.

**Теоретический материал.**

**1. Алфавит языка**

При записи программ разрешены символы:

- буквы латинского алфавита A-Z (в любом регистре), а также знак подчеркивания \_;
- буквы русского алфавита А-Я;
- цифры 0-9;
- специальные символы  $\> < = + - / * [ ] ( ) \{ \} . , : ; ^ @ ' \$ \#$
- пары символов (их нельзя разделять пробелами)  $\< > = > = := (* *) (..)$
- пробелы (рассматриваются как ограничители идентификаторов, констант, чисел, зарезервированных слов).

**2. Идентификаторы**

Неделимые последовательности символов алфавита образуют слова - *идентификаторы*, используемые для обозначения констант, переменных, процедур, функций и т.д.

Идентификатор должен начинаться с буквы или символа подчеркивания, не должен содержать пробелов и специальных символов.

**3. Константы**

В качестве констант могут использоваться числа, логические константы, символы и строки символов.

*Целые числа* записываются со знаком или без него по обычным правилам и могут иметь значение от – 2147483648 до +2147483647.

*Вещественные числа* записываются со знаком или без него с использованием десятичной точки и/или экспоненциальной части. Экспоненциальная часть начинается символом *e* или *E*, за которым могут следовать знаки «+» или «-» и десятичный порядок. Символ *e* (*E*) означает десятичный порядок и смысл «умножить на 10 в степени». Например, запись 3.14E5 означает  $3,14 \cdot 10^5$ , а запись -17e-2 – это  $-17 \cdot 10^{-2}$ .

**4. Выражения**

Выражение задает порядок выполнения действий над элементами данных и состоит из операндов (констант, переменных, функций, круглых скобок и знаков операций).

Действия в выражении выполняются слева направо с соблюдением старшинства (в порядке убывания):

- 1) \* (умножение), / (деление), div (целочисленное деление), mod (целочисленное деление с остатком по модулю), and (логическая операция “И”);
- 2) + (сложение), - (вычитание), or (логическая операция “ИЛИ”);

3) операции отношений: = (равно),  $\neq$  (не равно), < (меньше), > (больше),  $\leq$  (меньше или равно),  $\geq$  (больше или равно).

Для изменения порядка выполнения действий используются круглые скобки. Число открывающихся скобок равно числу закрывающихся. Любое выражение в скобках вычисляется раньше, чем выполняется операция, предшествующая скобкам.

В выражение могут входить функции. Наиболее часто употребляемые функции называют стандартными. Для работы с ними не надо ни заказывать библиотеку, ни описывать их предварительно в программе. Примеры стандартных математических функций:

ABS(x) – модуль  $x$  ( $|x|$ );

SQR(x) – квадрат числа  $x$  ( $x^2$ );

SQRT(x) – квадратный корень из  $x$  ( $\sqrt{x}$ );

LN(x) – натуральный логарифм от  $x$  ( $\ln x$ );

LG(X) - LN(x)/LN(10)

EXP(x) –  $e$  в степени  $x$  ( $e^x$ );

SIN(x) – синус  $x$  ( $\sin x$ );

COS(x) – косинус  $x$  ( $\cos x$ );

EXP(x\*ln(k)) –  $k$  в степени  $x$  ( $k^x$ );

Аргумент этих функций может быть как вещественным, так и целым. Результат – всегда вещественный.

## 5. Типы данных

Любые данные, т.е. константы, переменные, значения функций или выражения в Турбо Паскале характеризуются своими типами. Тип определяет множество допустимых значений, которые может иметь тот или иной объект, а также множество допустимых операций, которые применимы к нему. Все типы данных разделяются на две группы – простые и составные.

К простым (скалярным) типам относятся:

INTEGER - данные этого типа могут принимать только целые значения (положительные, отрицательные, 0) в диапазоне от -32768 до +32767;

REAL - величины этого типа могут принимать только вещественные значения (числа с дробной частью, целая часть от дробной отделяется точкой);

## 6. Структура программы

Структура программы должна быть такой:

<Заголовок программы>

{Блок описаний}

BEGIN

{Раздел исполняемых операторов}

END.

## 7. Заголовок программы

В заголовке указывается имя программы. Общий вид заголовка:

program n;

здесь  $n$  – имя программы.

Заголовок программы необязателен, его можно опускать без каких-либо последствий для программы.

## 8. Блок описаний

В блоке описаний объявляются идентификаторы типов, констант, переменных, а также метки, процедуры и функции. Блок описаний может состоять из пяти разделов, которые должны следовать в строго определенном порядке:

1) раздел меток (label);

2) раздел констант (const);

3) раздел типов (type);

4) раздел переменных (var);

5) раздел процедур и функций.

## 9. Раздел меток (label)

Любой выполняемый оператор может быть снабжен меткой – положительной константой, содержащей не более 4-х цифр. Метка отделяется от оператора двоеточием. Все метки, встречающиеся в программе, должны быть описаны в разделе label. Общий вид:

label l1, l2, l3...;

здесь l1, l2, l3... - метки.

## 10. Раздел констант (const)

Если в программе используются константы, имеющие достаточно громоздкую запись (например, число  $\pi$  с 8-ю знаками), либо сменные константы (например, для задания варианта программы), то такие константы обычно обозначаются какими-либо именами и описываются в разделе const. Это делает программу более наглядной и удобной при отладке и внесении изменений.

Общий вид:

const a1 = c1; a2 = c2; ...

Здесь a1, a2, ... – имя константы, c1, c2, ... – значение константы.

Пример.

const pi=3.14; c=2.7531;

## 11. Раздел переменных (var)

В разделе var вводится имя каждой переменной и указывается, к какому типу эта переменная принадлежит:

var v11, v12, ...: type1;  
v21, v22, ...: type2; ...

Здесь v11, v12, ... - имена переменных; type1 – тип переменных v11, v12, ...; type2 - тип переменных v21, v22, ...

Раздел процедур и функций

Те алгоритмы, которые оформляются как подпрограммы (процедуры и функции) помещаются в главной программе после раздела var и перед begin программы.

## 12. Раздел действий (операторов).

Эта часть программы начинается с ключевого слова begin и заканчивается словом end, после которого должна стоять точка (end.). Раздел действий - это выполняемая часть программы, состоящая из операторов.

## 13. Оператор присваивания

Под операторами в языке Паскаль подразумевают описание действий. Операторы отделяются друг от друга точкой с запятой. Если оператор стоит перед end, until или else, то в этом случае точка с запятой не ставится.

Общий вид оператора присваивания:

v:=a;

здесь v – переменная, a – выражение, := операция присваивания. Выражение a может содержать константы, переменные, названия функций, знаки операций и скобки. В операторе v:=a переменная v и выражение a должны иметь один и тот тип.

Примеры.

f:=3\*c+2\*sin(x);

x:=x+1;

Замечание. Разрешается присваивать переменной типа real выражение типа integer. Но нельзя присваивать переменной типа integer выражение типа real.

## 14. Процедура ввода информации

Общий вид:

Read (v1, v2, ...,vn);

или

Readln (v1, v2, ...,vn);

здесь v1, v2, ...,vn – идентификаторы переменных.

Значения переменных вводятся с клавиатуры и должны соответствовать типам переменных. В случае использования процедуры readln, после ввода происходит переход на следующую строку.

## 15. Процедура вывода информации на печать

Общий вид оператора:

write(p<sub>1</sub>, p<sub>2</sub>, ..., p<sub>n</sub>);

или

writeln(p<sub>1</sub>, p<sub>2</sub>, ..., p<sub>n</sub>);

Здесь p<sub>1</sub>, p<sub>2</sub>, ..., p<sub>n</sub> - список выражений, значения которых выводятся на печать.

Оператор write оставляет курсор в конце выведенной строки текста.

В случае использования процедуры writeln, после печати происходит переход на следующую строку.

Кроме значений выражений, на печать можно выводить и произвольный набор символов, заключенный в апострофы, например

writeln('p=', p);

Этот оператор выполняется так: сначала выводятся символы, заключенные в апострофы. Затем выводится значение переменной p. На экране в результате работы оператора появится:

p=13.5

**Пример 1.** Вычислить длину окружности радиуса 5,785.

**program** t10;

var r=5.785;

**var** l:real;

**begin**

l:=2\*3.1416\*r;

**writeln**(' l=',l);

**end.**

Имеется возможность задать ширину поля (число позиций) M для выводимой величины P:

Write (P1:M1, P2:M2, ...PN:MN);

Для вещественных чисел можно задавать поля M и N, где M – общее число позиций, отводимых под все число, N – число позиций под его дробную часть.

Например,

Write (P:10:2);

Здесь под P отводится 10 позиций, 2 из них под дробную часть.

## Пример №2

Вычислить значения  $p = \lg(a + x^2) + \sin^2\left(\frac{z}{a}\right)$ ,  $t = x^a$  и  $y = a \operatorname{tg}^3(a + x^2) + \sqrt{\frac{z^2}{a^2 + x^2}}$

при a=0,59; z=-4,8; x=2,1

**Program** pr\_1;

var a,z,x,y, p, c, t: real;

**begin**

a:=0.59; z:=-4.8; x:=2.1;

c:=sin(x\*x)/cos(x\*x);

y:=a\*c\*sqr(c)+sqr(z\*z/(a\*a+x\*x));

p:=(ln(a+x\*x))/ln(10)+sqr(sin(z/a));sin(z/a)\*sin(z/a)

t:=Exp(a\*ln(x));

writeln('При a=', a: 4:2, ' z=', z:4:1, ' x=', x: 3:1);

writeln ('p=', p:9:4, ' y=', y:9:4, ' t=', t: 9:4);

**readln**;

**end.**

Результаты вычислений:

При a=0.59 z = - 4.8 x =2.1

p =1.6217 y = 21.6350 t =1.5492

### **Контрольные вопросы.**

1. Как в программе на языке Pascal описываются переменные?
2. Какие бывают типы переменных?
3. Какой вид имеет оператор присваивания?
4. Каким символом отделяются друг от друга операторы в программе?
5. Какая процедура служит для вывода информации на печать?
6. Какая процедура служит для ввода значений с клавиатуры?
7. Какие стандартные тригонометрические функции существуют в языке Pascal?

### **Занятие №26 (лекция)**

**Дата:** 02 декабря 2020г.

**Тема:** *Этапы решения задач на компьютере.*

#### **Задание:**

1. Изучить теоретический материал (презентацию), для изучения перейти по [ссылке](#)
2. Ответить письменно на контрольные вопросы:

1. Что включает в себя этап постановка задачи?
2. Что включает в себя этап разработка алгоритма?
3. Приведите алгоритм этапов решения задач?

### **Занятие №27 (лекция)**

**Дата:** 04 декабря 2020г.

**Тема:** *Методы проектирования программ.*

#### **Задание:**

1. Изучить теоретический материал (презентацию), для изучения перейти по [ссылке](#)
2. Ответить письменно на контрольные вопросы:

1. Перечислите подсистемы информационных систем, дайте их определение.
2. Перечислите жизненные циклы информационных систем, дайте их определение