

Лекция. Основные понятия Турбо Паскаль

Структура программы

```
Program <имя> (Input, Output);  
Uses  <имя1,имя2,...>;  
Label . . . ;  
Const . . . ;  
Type  . . . ;  
Var   . . . ;  
  
Procedure  <имя>;  
    <тело процедуры>  
Function <имя>  
    <тело функции>  
  
begin  
    <операторы>  
End.
```

2. РАЗДЕЛ Uses

Пример

```
Uses Crt, Dos, MyLib;
```

3. РАЗДЕЛ ОПИСАНИЯ МЕТОК

Пример

```
Label M1,M2;  
begin  
M1:   a:=3;  
      b:=5;  
      s:=a+b; goto M1;  
end.
```

4. Раздел писания констант.

Пример

Const

A = 5;

Start = 'Начало'; {строковая константа}

Kod = \$124; {константа шестнадцатеричное значение}

5. Раздел описания типов данных

Пример

Type

Dni = 1. .31;

Bukva = ('a' . . 'z');

6. Раздел описания переменных

Пример

Var

A,B,C : integer;

Summa : real;

Start : boolean;

7. Раздел описания процедур и функций

Для описания подпрограмм используются зарезервированные слова **Procedure** и **Function**.

8. Раздел операторов

```
begin  
    оператор;  
    оператор;  
    оператор;  
end.
```

9. Комментарии

Комментарий ограничивается скобками { } или (* *).

Простейшие операторы Turbo Pascal

1. Оператор присваивания

Пример 1.

```
Radius:=5.75;
```

```
Nomer:=2; Zadacha:=5;
```

```
Nazv:='Модель 1030';
```

```
Telefon:='Nokia';
```

Пример 2. Поменять значения двух переменных a и b.

Способ 1. Решить эту задачу можно, например, введением третьей переменной t для временного хранения значений одной из переменных.

```
a:=5;  
b:=7;  
. . .  
t:=a;  
a:=b;  
b:=t;
```

Способ 2. Эту же задачу можно решить не используя третью переменную, используя оригинальную последовательность операций присваивания.

```
a:=5;  
b:=7;  
. . .  
a:=a+b;  
b:=a-b;  
a:=a-b;
```

2. Оператор безусловного перехода goto

3. Пустой оператор

4. Составной оператор

Пример

```
begin  
  A:=A+B-(N-D);  
  Rez:=A*Pi;  
end;
```

5. Операторы ввода- вывода

Пример

```
var
A,B,Sum1 : integer;
C,D,Sum2 : real;
. . . .
read (A,B);
Sum1:=A+B;
read(C,D);
Sum2:=C+D;
```

Пример

```
write(234);
write (A+B-2);
write('Результат вычислений = ', Rezalt);
```


6. Условные операторы

1. `if <условие> then <оператор1> else <оператор2>`
2. `if <условие> then <оператор>`

Примеры фрагментов программ с оператором `if`

```
if A<20 then
if A>=15 then
writeln ('диапазон 15-20')
else
writeln ('за пределами');
```

```
if (A=B) and (C=D) then
begin
  writeln('Норма');
  F:=0;
end
else
begin
  writeln('Превышение нормы');
  F:=100
end;
```

Оператор выбора case

```
case <выражение-селектор> of
    <список1>:<оператор1;>
    . . . .
    <списокN>:<операторN;>
else <оператор>
end;
```

Примеры

```
case s of
1..10: writeln('число ', s:4, ' в диапазоне 1-10');
11..20: writeln('число ', s:4, ' в диапазоне 11-20');
21..30: writeln('число ', s:4, ' в диапазоне 21-30');
else writeln ('число вне пределов контроля');
```

```
case m of
1: z:=m+10;
2: z:=m+100;
3: z:=m+1000;
```

7. Операторы повтора

Оператор цикла for

1. for <параметр цикла> := <S1> to <S2> do <оператор>;
2. for <параметр цикла> := <S1> downto <S2> do <оператор>;

Примеры

```
for i:=10 to 14 do write (i:3);
```

```
for i:=14 downto 10 write (i:3);
```

```
for Ch:='a' to 'e' do write (Ch:2);
```

Оператор повтора repeat

```
repeat
    <оператор;>
    . . .
    <оператор:>
until <условие>;
```

Примеры

```
repeat
    s:=100/A; D:=A/5;
    if s<0.5 then goto Metka;
    A:=A+1
until (A<=100) and (D<>0);
Metka: . . .
```

```
repeat
    s:=100/A;
    if s<0.5 then A:=-999;
    A:=A+1
until (A<=100);
```

Оператор while

```
while <условие> do <тело цикла>;
```

Примеры

```
while True do  
write ('бесконечный цикл');
```

```
i:=30;  
while (i<0) do i:=i+i;  
write (i);
```

Примеры простейших программ.

Алгоритм накопления суммы.

Дана последовательность: $\sin 2x, \sin 4x, \sin 6x, \dots, \sin 16x$, где x - заданное вещественное число. Вычислить сумму членов последовательности, которые по модулю больше 0.3.

```
Program Z2;  
Var a, x, S: real;  
    n: integer;  
Begin  
    writeln('Введите число x');  
    read (x); S:=0;  
    for n := 1 to 8 do  
        begin  
            a := sin(2 * n * x);  
            if abs(a) > 0.3 then S := S + a;  
        end;  
    writeln('S=', S:6:2);  
End.
```

Алгоритм накопления произведения

Дана последовательность: $\cos 0.1, \cos 0.2, \cos 0.3, \dots, \cos 10$. Вычислить значение: $P = |PO|$, где PO - произведение отрицательных членов последовательности.

```
Program Z_3;
Var
  x, y, P, PO: real;
Begin
  PO := 1; x := 0.1;
  while x <= 10 do
    begin
      y := cos(x);
      if y < 0 then PO := PO * y;
      x := x + 0.1;
    end;
  P := abs(PO);
  writeln('P=', P:6:2);
End.
```

Составить программу проверки есть ли в тексте буква "s"

```
program z11;
uses crt;
var t   : string;
    i   : integer;
    ot  : boolean;
begin
  clrscr;
  writeln('введите текст:');readln(t);
  for i:=1 to length(t) do
    if t[i]='s' then ot:=true;
  if ot=true then write('да')
                else write('нет');
  readln;
end.
```


Определить лежит ли точка а на прямой $y=kx+l$

```
uses crt;
program z14;
var x,y,l,k : integer;
begin
  clrscr;
  write('x=');readln(x);
  write('y=');readln(y);
  write('k=');readln(k);
  write('l=');readln(l);
  if y=k*x+l then write('Да')
                 else write('Нет');
  readln;
end.
```

Даны координаты вершин треугольника ABC A(x1;y1), B(x2;y2), C(x3;y3) является ли треугольник равнобедренным

```
program z19;
uses crt;
var x1,x2,x3,y1,y2,y3,a,b,c : real;
begin
  clrscr;
  write('x1=');readln(x1); write('y1=');readln(y1);
  write('x2=');readln(x2); write('y2=');readln(y2);
  write('x3=');readln(x3); write('y3=');readln(y3);
  a:=sqrt(sqr(x1-x2)+sqr(y1-y2));
  b:=sqrt(sqr(x2-x3)+sqr(y2-y3));
  c:=sqrt(sqr(x1-x3)+sqr(y1-y3));
  if (a=b)or(a=c)or(b=c) then write('равнобедренный')
  else write('не равнобедренный');
  readln;
end.
```

Составить программу для определения лежит ли точка (x3;y3), на прямой проходящей через точки (x1;y1),C(x2;y2)

```
program z20;
uses crt;
var x1,x2,x3,y1,y2,y3 : real;
begin
  clrscr;
  write('x1=');readln(x1);
  write('y1=');readln(y1);
  write('x2=');readln(x2);
  write('y2=');readln(y2);
  write('x3=');readln(x3);
  write('y3=');readln(y3);
  if (x3-x1)*(y2-y1)-(y3-y1)*(x2-x1)=0
    then write('лежит')
    else write('не лежит');
  readln;
end.
```

Дано предложение, определить кол-во слов в нём.

```
program z38;  
uses crt;  
var tec : string;  
    l,i,n : longint;  
begin  
    clrscr; write('введите текст:');readln(tec);  
    l:=length(tec)+1;tec[l]:=' '  
    for i:=1 to l do if tec[i]=' ' then n:=n+1;  
    write('В тексте ',n,' слов');  
    readln;  
end.
```

Определить является ли данное слово перевертышем.

```
program z40;  
uses crt;  
var a,b,c : string;  
    i : longint;  
begin  
    clrscr;  
    write('Введите слово: ');readln(a);  
    b:='';  
    for i:=1 to length(a) do b:=a[i]+b;{ переворачиваем слово }  
    if a=b then write('перевертыш')  
        else write('не перевертыш');  
    readln;  
end.
```

Дано натуральное число n. Верно ли, что сумма цифр этого числа является нечётной.

```
program z1;
uses crt;
var a : string; t,er,n,i,s : integer;
begin
  clrscr;
  write('введите число ');readln(a);
  s:=0;
  for i:=1 to length(a) do
  begin
    val(a[i],t,er)    преобразует строковое в числовое
    s:=s+t;
  end;
  if s mod 2<>0  остаток от деления
  then write('сумма яв-ся нечётной')
  else write('сумма яв-ся чётной');
  readln;
end.
```