

ЯЗЫК ПРОГРАММИРОВАНИЯ ПАСКАЛЬ.



Язык программирования *Pascal*

Язык программирования Паскаль был разработан профессором, директором Института информатики Швейцарской высшей политехнической школы Николаусом Виртом в *1968-1970* гг. как язык обучения студентов программированию.

Но думать, что Паскаль- язык исключительно для обучения, было бы неверно.



Николаус
Вирт

Вот что говорил об этом Н. Вирт (1984 г.):

«Утверждалось, что Паскаль был разработан в качестве языка для обучения. Хотя это утверждение справедливо, но его использование при обучении не являлось единственной целью. На самом деле я не верю в успешность применения во время обучения таких инструментов и методик, которые нельзя использовать при решении каких-то практических задач».*

Основные средства языка

Символы языка – это элементарные знаки, используемые при составлении текстов.

Алфавит языка – набор таких символов.

Алфавит языка Turbo Pascal 7.0 включает:

- все латинские прописные и строчные буквы
- арабские цифры (0 – 9)
- символы + - * / = < > , . ; : ‘ _ () { } и др.
- служебные (зарезервированные) слова

Основные средства языка

Для записи команд, имен функций, поясняющих терминов *Turbo Pascal 7.0* существует набор строго определенных слов, которые называются **служебными**.

Служебные слова делятся на три группы:

- операторы (*READ, WRITELN* и др.)
- имена функций (*SIN, COS* и др.)
- ключевые слова (*VAR, BEGIN, END* и др.)

Типы данных

Название	Обозначение	Допустимые значения	Область памяти
Целочисленный	Integer	-32 768 ... 32 767	2 байта со знаком
Вещественный	Real	$\pm(2,9 \cdot 10^{-39} \dots 1,7 \cdot 10^{+38})$	6 байтов
Символьный	Char	Произвольный символ алфавита	1 байт
Строковый	String	Последовательность символов длиной 255	1 байт на символ
Логический	boolean	True и False	1 байт

Структура программы на языке Паскаль

Общий вид программы:

Program	(имя программы)
const	(список постоянных значений)
type	(описания сложных типов данных)
var	(описания данных программы)
begin	(начало программного блока)
	<i>(алгоритм)</i>
end.	(конец программы)

Оператор вывода Write

Оператор вывода Write	Результат выполнения оператора
Write (-500)	-500
Write (3 + 2)	5
Write ('Хорошо')	Хорошо
Write (3 + 2, 8+1)	59
Write (3 + 2, ' ', 8+1)	5 9

Примечание:

Текст в отличие от чисел и выражений, нужно брать в одинарные кавычки

Формат вывода

Формат вывода – это указываемое после двоеточия целое число, определяющее, сколько позиций на экране должна занимать выводимая величина

Оператор вывода Write	Результат выполнения оператора
Write ('s=', s:2:0);	s = 15
Write ('s=', s:2:1);	s = 15.0
Write ('s=', s:5:1);	s = 15.0

Первая программа на Паскале

```
BEGIN  
  Write ('Начало');  
  Write (8+1);  
  Write (5-2 , 3*8);  
  Write (5-2, ' , 3*8);  
  Write (5-2, ', ' , 3*8);  
END.
```

BEGIN будем ставить в начале программы, чтобы компьютеру было видно, откуда она начинается.

END с точкой ставится в конце программы.

Точкой с запятой необходимо отделять операторы друг от друга.

Оператор WriteLn

При выполнении нового оператора *write* вывод продолжается в той же строке. Чтобы осуществить переход к новой строке, используется оператор *writeln*. Других различий между операторами *writ* и *writeln* нет.

Программа	Что видим на экране
BEGIN Write ('Ама'); Write ('зонка') END.	Амазонка
BEGIN Write ('Ама'); Writeln ('зонка') END.	Амазонка
BEGIN Writeln ('Ама'); Write ('зонка') END.	Ама зонка
BEGIN Writeln ('Ама'); Writeln ('зонка') END.	Ама зонка

Вторая программа на Паскале

Напишите программу вычисляющую длину окружности и площадь круга радиуса 5,4 см.

Площадь круга
 $S = \pi r^2$

Длина круга
 $C = 2\pi r$

```
Program p_2;  
const pi=3.14;  
var r, c, s: real;  
begin  
    r:=5.4;  
    c:=2*pi*r;  
    s:=pi*r*r;  
writeln ('c=', c:6:4);  
writeln ('s=', s:6:4)  
End.
```

Задания

1. В углу прямоугольного двора размером 50 на 30 м стоит прямоугольный дом размером 20 на 10 м. Подсчитать площадь дома, свободную площадь двора и длину забора.
2. Автомобиль три часа ехал со скоростью 80 км/ч и 2 часа со скоростью 90 км/ч. Вычислить среднюю скорость автомобиля (она равна суммарному пути, деленному на суммарное время).

Решение задачи №1

```
VAR    a, b, c, d, Sdoma, Sdvora,    DLzabora: integer;  
BEGIN  
    a:=50;  
    b:=30;  
    c:=20;  
    d:=10;  
    Sdoma:=c*d;  
    Sdvora:=a*b-c*d;  
    DLzabora:=2*(a+b)-(c+d);  
    WriteLn ('Площадь дома=',Sdoma, ' ', 'Площадь  
двора=',Sdvora, ' ', 'Длина забора=',DLzabora)  
END.
```

Решение задачи №2

```
VAR t1, t2, v1, v2, S1, S2, Vsred: real;  
BEGIN  
    t1:=3;  
    t2:=2;  
    v1:=80;  
    v2:=90;  
    S1:=t1*v1;  
    S2:=t2*v2;  
    Vsred:=(S1+S2)/(t1+t2);  
    WriteLn ('Средняя скорость автомобиля=', Vsred,  
            'км/ч')  
END.
```

Операторы ввода данных

Для ввода в оперативную память значений переменных

используется **read.**

READ (*<имя переменной>, <имя переменной>, <имя переменной>*)

СПИСОК ВВОДА

Для ввода данных с клавиатуры можно также использовать

оператор **readln**, который отличается от read только тем, что после его выполнения курсор переходит на новую строку.

Пример. Сложить два числа – 20 и 16.

Сравните две программы решения этой задачи:

```
VAR a, b: integer;  
BEGIN  
    a:=20;  
    b:=16;  
    WriteLn (a+b)  
END.
```

```
VAR a, b: integer;  
BEGIN  
    ReadLn (a,b);  
    WriteLn (a+b)  
END.
```

Условный оператор

Общий вид:

If *<условие>* **then** *<оператор_1>* **else** *<оператор_2>*

Для записи неполных ветвлений используется сокращенная форма записи:

If *<условие>* **then** *<оператор>*

 Перед **else** знак « ; » не ставится

Условный оператор

Пример

Определить принадлежность точки **X** отрезку **[a; b]**. Если точка **X** принадлежит данному отрезку, то выводится ответ **‘Да’**, в противном случае - **‘Нет’**

```
Program n_9;  
Var x, a, b:real;  
begin  
  writeln ('Введите a, b');  
  readln (a, b);  
  writeln ('Введите x');  
  readln (x);  
  if (x>=a) and (x<=b) then  
    writeln ('Да')  
  else writeln ('Нет')  
End.
```

Условный оператор

Пример

Присвоить
переменной Y
значение большей из
трех величин A, B, C

```
Program n_10;  
Var y, a, b, c:integer;  
begin  
  writeln ('Введите a, b, c');  
  readln (a, b, c);  
  y:=a;  
  if (b>y) then y:=b;  
  if (c>y) then y:=c;  
  writeln ('y=', y)  
End.
```

*Сокращенная
форма ветвления*



Составной оператор



В условном операторе и после **then**, и после **else** можно использовать только один оператор. Если требуется выполнить несколько операторов, то их объединяют в составной оператор.

Begin *<последовательность операторов>* **end**

Составной оператор

Пример:

Решить квадратное уравнение

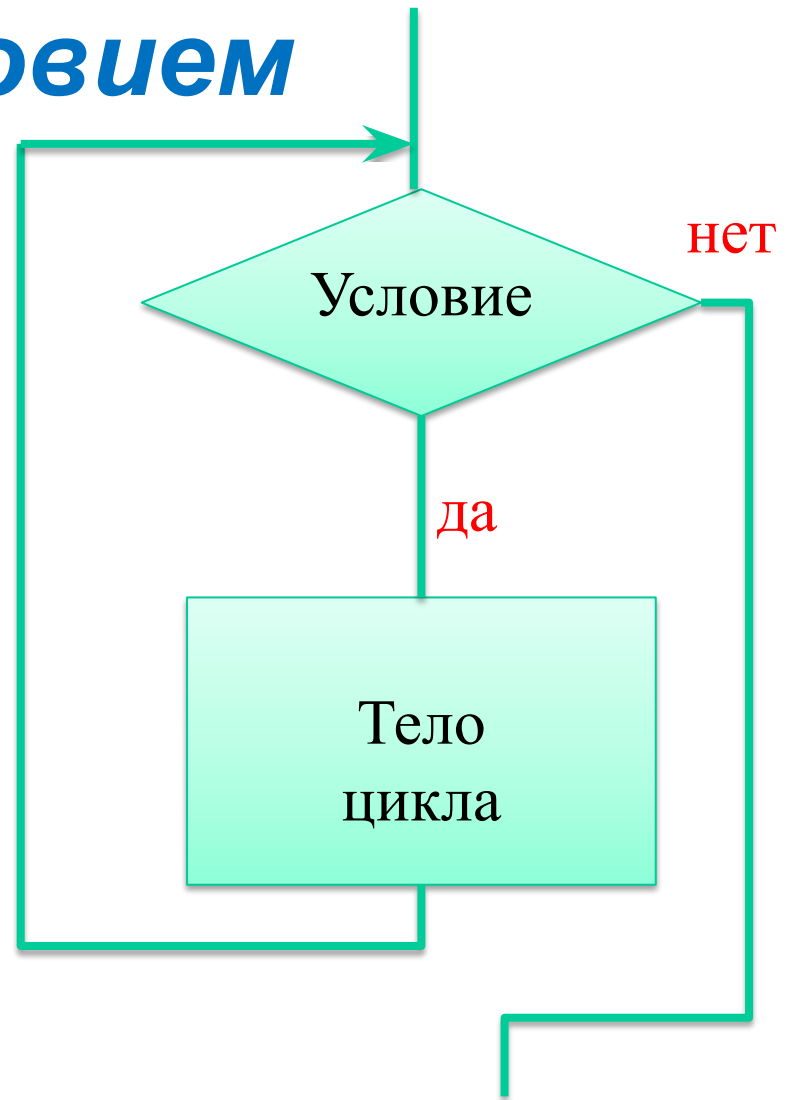
$x^2 + 18x - 19 = 0$
 $D = 400$
 $x_1 = -19$
 $x_2 = 1$

```
program n_11;  
var a, b, c, d: real;  
var x, x1, x2: real;  
begin  
write ('Введите коэффициенты a, b, c');  
readln (a, b, c);  
d:= b*b- 4*a*c;  
if d<0 then writeln ('Корней нет');  
if d=0 then  
begin  
x:=-b/(2*a);  
writeln ('Корень уравнения x=', x)  
end;  
if d > 0 then  
begin  
x1:= (-b+sqrt(d))/(2*a);  
x2:= (-b-sqrt(d))/(2*a);  
writeln ('Корни уравнения');  
writeln ('x1=', x1);  
writeln ('x2=', x2)  
end  
end.
```

Циклический оператор

Цикл с предусловием

Цикл повторяется до тех пор, пока проверка этого условия будет давать результат «истина» (true), т.е. пока условие выполняется. Если условие сразу оказывается ложным, цикл не будет выполнен ни разу.



Цикл с предусловием

Оператор цикла с предусловием в Паскале имеет следующий формат:

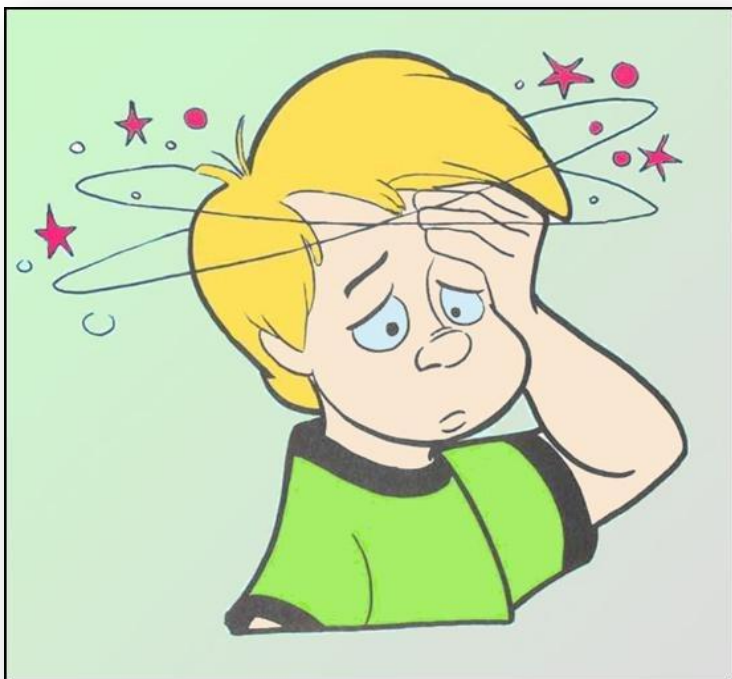
```
while <выражение> do <оператор>;
```

Служебное слово **while** означает «пока», **do** – «делать».

Оператор стоящий после слова **do**, называется *телом цикла*. Тело цикла может быть простым или составным оператором, т.е. последовательностью операторов между служебными словами **begin** и **end**.

Задание

Не пользуясь оператором деления, получить частное q и остаток r от деления целого числа x на целое число y .



Program n_14;

```
var x, y, q, r: integer;
```

Begin

```
writeln ('Введите делимое x');
```

```
readln (x);
```

```
write('Введите делитель y');
```

```
readln (y);
```

```
r:=x;
```

```
q:=0;
```

```
while r >= y do
```

```
begin
```

```
  r:=r-y;
```

```
  q:=q+1
```

```
end;
```

```
writeln ('Частное q=', q);
```

```
writeln ('Остаток r=', r)
```

End.

Задание

Вычислить факториал заданного числа N .

Примечание

Факториалом целого положительного числа N называется произведением всех целых чисел от 1 до N включительно.

Например:

$1! = 1;$
 $2! = 1 * 2 = 2;$
 $3! = 1 * 2 * 3 = 6;$
 $4! = 1 * 2 * 3 * 4 = 24;$
 $5! = 1 * 2 * 3 * 4 * 5 = 120$ и т.д.

```
Program factorial;  
var a, n: integer;  
var fact: real;  
Begin  
  writeln ('Введите число');  
  readln (a);  
  n:=1;  
  fact:=1;  
while n < a do  
  begin  
    n:=n+1;  
    fact:=fact*n;  
  end;  
  writeln ('Число', a, ', ', 'Факториал  
числа =', fact);  
end.
```

Задание

Напишите программу на Паскале вычисления суммы всех натуральных чисел, не превышающих заданного натурального числа N .

