

# ЛИНЕЙНЫЕ ПРОГРАММЫ

Программирование простых программ

# Вид программы на Паскале

- Программа на языке Паскаль представляет собой текст. Она может быть написана в любом текстовом редакторе, с расширением pas.
- Программа состоит из 4<sup>х</sup> основных частей:
  1. Заголовок программы
  2. Раздел описаний (соглашения)
  3. Исполняемая часть (операторы)
  4. Вывод результатов.

## I. Заголовок

**Program** <имя программы>;

## II. Раздел объявлений (описаний)

**Const** имя=значение;

**Var** имя, имя: тип1;

имя, имя: тип2;

## III. Тело программы

**Begin**

операторы

**End.**

# Стандартные функции

Функция	Название	Примеры
<b>abs(x)</b>	Модуль X	$\text{abs}(-5) = 5$
<b>sqr(x)</b>	Квадрат X	$\text{sqr}(3) = 9$
<b>sqrt(x)</b>	Корень из X	$\text{sqrt}(16) = 4, x \geq 0$
<b>pi</b>	число $\pi$	$\text{Pi} = 3.14\dots$
<b>round(x)</b>	Округление до целых	$\text{Round}(2.8) = 3$ $\text{Round}(2.3) = 2$
<b>trunc(X)</b>	Целая часть от X	$\text{Trunc}(2.8) = 2$ $\text{Trunc}(2.3) = 2$
<b>sin(x)</b>	$\sin x$	Аргумент записывается в радианах: 1 рад - это примерно $57^\circ$ , $\pi$ рад = $180^\circ$
<b>cos(x)</b>	$\cos x$	

# Числовые типы данных

Функция	Назначение	Тип аргумента	Тип результата
abs (x)	Модуль X	integer, real	Такой же, как у аргумента
sqr (x)	Квадрат X	integer, real	Такой же, как у аргумента
sqrt (x)	Квадратный корень из X	integer, real	real
round (x)	Округление x до ближайшего целого	real	integer
int (x)	Целая часть X	real	integer
frac (x)	Дробная часть X	real	real
random	Случайное число от 0 до 1	-	real
random (x)	Случайное число от 0 до X	integer	integer

**ЗАДАЧА №1.** Ввести с клавиатуры длины катетов  
треугольника. Найти гипотенузу.

1. Аргументы: **a, b**
2. Результаты: **c**

Формулы:  **$c := \text{Sqrt}(\text{Sqr}(a) + \text{Sqr}(b))$** ;

**Program** Gip;

Var a, b, c: real;

**Begin**

Write ('Ввести А и В ');

Readln (a,b);

c:=Sqrt (Sqr(a)+Sqr(b));

Writeln ('c=',c:6:2);

**End.**

# Случайные числа

1) Включить генератор случайных чисел  
**Randomize.**

2) Значение переменной с помощью  
функции "**Random**"

а) для дробных чисел из интервала [ X0; X1 ]

**X:=Random \* (X1-X0) + X0;**

б) для целых чисел из интервала [ X0; X1 ]

**X:=Random (X1-X0) + X0;**

в) упрощённый вариант

**X:=Random (30);**

## Задача №2

➤ Значение массы тела  $m$  ввести с использованием функции **random**. Считая  $g=9.8$ , найти вес тела  $P$ .

Аргументы:  $g, m$

Результаты:  $P$

```
Program Ves;  
  uses crt;  
  const g=9.8;  
  var m: integer;  
      P: real;  
  
Begin  
  m:=Random (30);  
  P:=m*g;  
  Writeln ('Вес=', P:5:0)  
  
End.
```

## Задача №3

- ✓ Ввести с клавиатуры скорость катера  $v_k$  и скорость течения  $v_t$ .
- ✓ Какой путь  $S$  пройдет катер по течению за  $t=3$  часа?

Аргументы:  $V_k, V_t, t$

Результаты:  $S$

Промежуточные переменные:  $V_{kt}$

$$S = V_{kt} * t$$

$$V_{kt} = V_k + V_t$$

# **ПРОГРАММЫ ВЕТВЛЕНИЯ**

Программирование программ с ветвлением

# Задачи

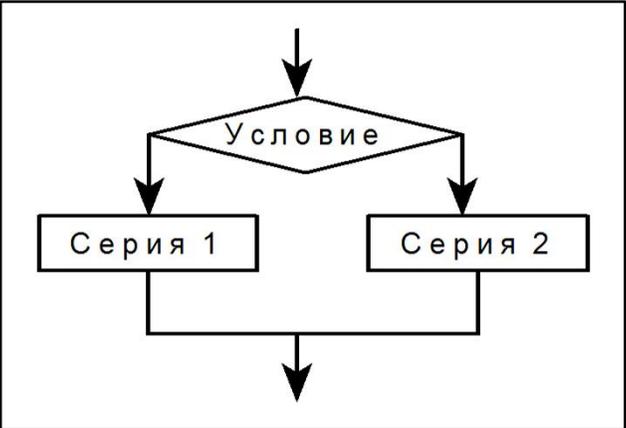
1. Даны три натуральных числа. Напишите программу, определяющую, существует ли треугольник с такими длинами сторон. Если такой треугольник существует, то определите его тип (равносторонний, равнобедренный, разносторонний).
2. Дано трёхзначное число. Напишите программу, которая определяет: есть ли среди цифр заданного целого трёхзначного числа одинаковые.

# Проверка программы

Пример входных данных	Пример выходных данных
a b c>> 1 2 1	Не существует
a b c>> 2 2 2	Равносторонний
a b c>> 20 20 30	Равнобедренный
a b c>> 3 4 5	Разносторонний

# Общий вид команды ветвления

■ Ветвление это оператор, выполняющий разные действия в зависимости от результата проверки условия.

Блок-схема:	Алгоритмический язык	Язык Бейсик
	<u>если</u> условие	<b>IF</b> условие <b>THEN</b> серия1 <b>ELSE</b> серия2
	<u>то</u> серия 1	Язык Паскаль
	<u>иначе</u> серия 2 <u>все</u>	<b>IF</b> условие <b>THEN</b> серия1 <b>ELSE</b> серия2

# Понятие команды ветвления

1. Полная форма условного оператора:

**If** условие **then** оператор1 **else** оператор2

2. Неполная форма условного оператора:

**if** условие **then** оператор

Перед **else** знак «;» не ставится.

# Пример решения задачи - БИД

## Алгоритмический язык

```
алг Пример
нач
цел А, В, Х
Ввод а, б
  если А >= В
    то Х := А
    иначе Х := В
  все
Вывод х
кон
```

## Язык Паскаль

```
Program pr1;
  Var a,b,X :integer;
Begin
  Write ('a='); Readln (a);
  Write ('b='); Readln (b);
  IF a>=b THEN X:=a ELSE X:=b;
  Writeln ('X=',X);
End.
```

Решение БИД

[Решение БИТ](#)

# Алгоритмический язык

алг Пример

нач

цел A, B, C, X

ввод A, B, C

X:=A

если  $B \geq X$  то  $X := B$

если  $C \geq X$  то  $X := C$

все

Вывод X

кон

# Паскаль

Program pr1;

Var a,b,X :integer;

Begin

Write ('a='); Readln (a);

Write ('b='); Readln (b);

X:=A;

IF  $b \geq x$  THEN  $x := b$ ;

IF  $c \geq x$  THEN  $x := c$ ;

Writeln ('X=',X);

End.

# Логические связки

Условия - логические выражения

Простые

Операции отношений

> < <> = >= <=

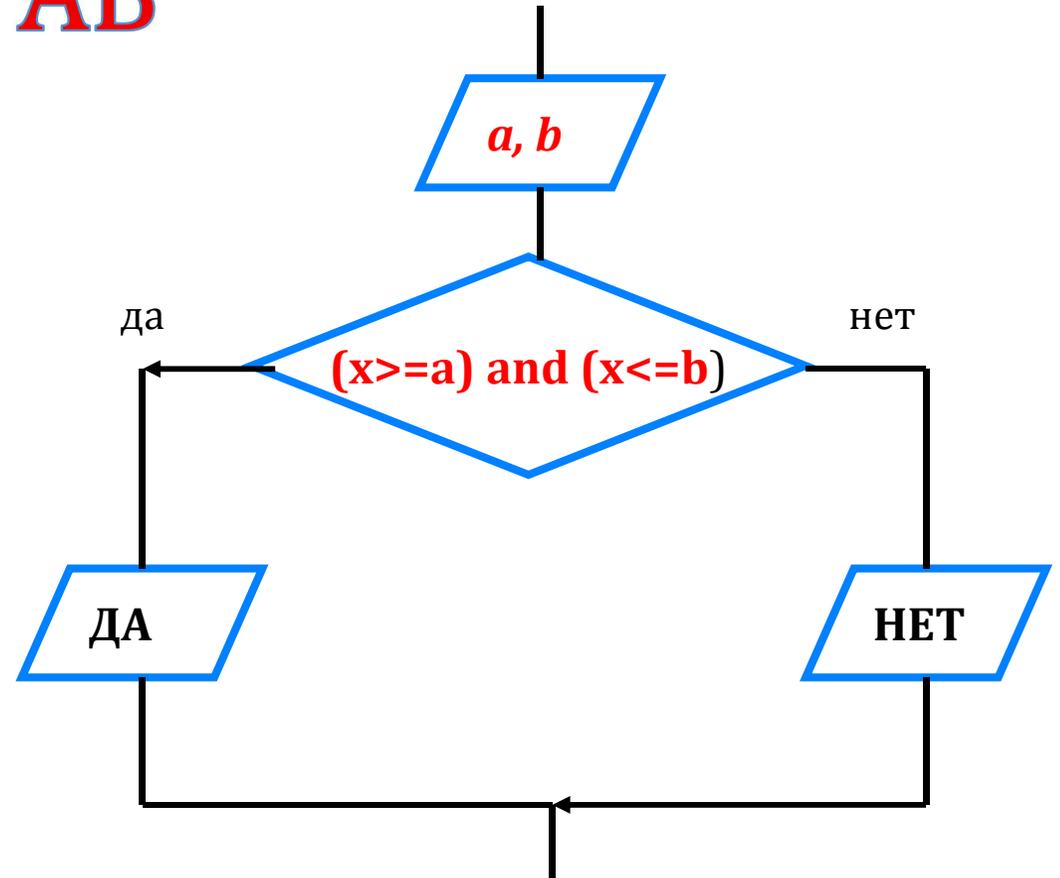
Сложные

Логические операции

и - and или - or

# Определение принадлежности точки "X" отрезку "AB"

```
program n_9;  
  var x, a, b: real;  
begin  
  readln (a, b);  
  write ('Введите значение точки');  
  readln (x);  
  if (x >= a) and (x <= b)  
    then writeln (' принадлежит отрезку')  
    else writeln (' не принадлежит отрезку')  
end.
```



**If** условие

**then begin**

оператор 1;

оператор 2

**end**

**else begin**

оператор 1;

оператор 2

**end;**

Составной оператор

```
program kvur;  
  var a, b, c, d, x, x1, x2: real;  
begin  
  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:9:3)  
  end;  
  if d>0 then begin  
    x1:=(-b+sqrt(d))/2/a;  
    x2:=(-b-sqrt(d))/2/a;  
    writeln ('Корни уравнения:');  
    writeln ('x1=', x1:9:3);  
    writeln ('x2=', x2:9:3)  
  end;  
end.
```

# Решение квадратного уравнения 2 способ



[Решение](#)

# Виды циклов

С  
параметром

```
For i:=N1 to N2 do  
Begin  
тело цикла  
End;
```

С пред-  
условием

```
While условие do  
begin  
оператор;  
...  
end;
```

С пост-  
условием

```
Repeat  
оператор;  
...  
оператор  
Until условие;
```

# ЦИКЛ С ПРЕД-УСЛОВИЕМ

Программирование циклов с заданным  
условием продолжения работы

Оператор один

**while** условие **do** оператор

Операторов больше одного

**While** условие **do**

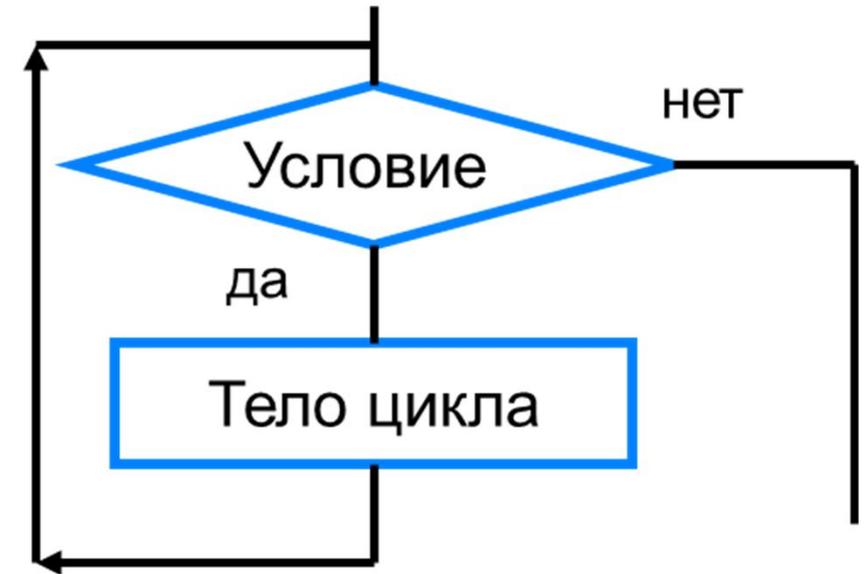
**begin**

оператор;

оператор;

...

**end;**



$\alpha$ <условие> - логическое выражение;

пока оно истинно, выполняется тело цикла;

$\alpha$ <оператор> - простой или составной оператор, с помощью которого записано тело цикла.

**Задача №1.** Суммировать целые числа от 1 до 10, используя цикл ПОКА Результат положить в ячейку S

**алг сумма10**

**нач**

**цел  $s, k$**

**$s:=0; k:=1$**

**нц пока  $k \leq 10$**

**$s:=s+k$**

**$k:=k+1$**

**кц**

**ВЫВОД S**

**КОН**

**Program** summa\_while;

Var s, k: integer;

**Begin**

k:=1;

**While** k <= 10

**begin**

s := s + k;

**k:=k+1**

**end;**

writeln ('s = ', s);

**End.**

решение

## ЗАДАЧА №2

Ввести с клавиатуры ряд чисел, признак  
окончания ввода – число 0.

Найти сумму введённых чисел

решение

**алг** сумма

**нач**

цел  $x, s$

$s := 0; x := 1$

**нц пока**  $x \neq 0$

ВВОД  $x$

$s := s + x$

**кц**

ВЫВОД  $s$

**кон**

**Program** summa\_while;

Var  $x, s$ : integer;

**Begin**

$x = 1;$

**While**  $x \neq 0$  **do**

**begin**

Write ('x='); Readln (x);

$s := s + x;$

**end;**

writeln ('s = ', s);

**End.**

решение

# ЦИКЛ С ПОСТ-УСЛОВИЕМ

Программирование циклов с заданным  
условием окончания работы

# Общий вид оператора

## **Repeat**

**<оператор>;**

**<оператор>;**

**...**

**<оператор>**

**Until <условие>;**

- Операторы в теле цикла записываются через «;»  
Перед «Until» точка с запятой не ставится.

# Формулы

**S:=0; P:=1; read (X);**

1. Сумма чисел: **S:=S+X**
2. Количество чисел: **S:=S+1**
3. Произведение чисел: **P:=P\*X**

**ЗАДАЧА Ввести с клавиатуры ряд чисел, признак окончания ввода – число 0. Найти сумму чисел**

**Program** Summa\_Repeat;

Var x, S: integer;

**Begin**

S:= 0;

**Repeat**

Readln (x);

S := S + x;

**Until** x = 0;

Writeln ('Сумма = ', S);

**End.**

# Самостоятельная работа

Ввести с клавиатуры ряд чисел, признак окончания ввода – число 0. Найти:

- а) сумму четных чисел;
- б) сумму чисел, кратных 5;
- в) кол-во чисел, оканчивающихся на 3

Смотреть  
[презентацию](#)  
виды условий

# ЦИКЛ С ПАРАМЕТРОМ

Программирование циклов с заданным числом повторений

# Сравните!

Алгоритмический язык	Паскаль
<p><u>Для</u> I <u>от</u> N1 <u>до</u> N2</p> <p><u>нц</u></p> <p>тело цикла</p> <p><u>кц</u></p>	<p><b>For</b> i:=N1 to N2 <b>do</b></p> <p><b>Begin</b></p> <p>тело цикла</p> <p><b>End;</b></p>
<p><u>Для</u></p>	<p><b>For</b></p>
<p><u>от</u></p>	<p><b>i:=</b></p>
<p><u>до</u></p>	<p><b>to</b></p>

# For $i:=N1$ to $N2$ do

- **Параметр цикла** (переменная цикла I) - это имя переменной дискретного типа, управляющей количеством повторений (считает, сколько раз будут выполнены операторы тела цикла). Параметр цикла пробегает все значения от начального до конечного, каждый раз автоматически увеличиваясь на один шаг.
- **Начальное и конечное значение** ( $N1, N2$ ) могут быть числами, переменными или выражениями. Они вычисляются только однажды при входе в цикл, эти значения запоминаются и не изменяются.

# Оператор цикла с шагом

Вид оператора	шаг
<b>For i:=1 to N do</b> тело цикла ;	1
<b>For i:=N downto 1 do</b> тело цикла;	-1

# Суммировать целые числа от 1 до 10

**алг** Сумма

**нач**

цел i, s

s:=0

нц для i от 1 до 10

s := s+i

кц

**кон**

Program Summa;

Var I, S: Integer;

Begin

S:=0;

For I:=1 to 10 do s:=s+I;

WriteLn ('s=',s);

End.

# Задача Вычисления факториала 10!

**Program** fakterial;

**Var** I, P: Integer;

**Begin**

P:=1;

For I:=1 to 10 do P:=P\*I;

WriteLn ('P=',P);

**End.**

# Задача Алгоритм вычисления факториала $N!$ $N! = 1*2*3*...*N$

**Program** fakterialn;

uses crt;

var I, P, n: Integer;

**Begin**

P:=1; readln (n);

For I:=1 to n do P:=P\*I;

WriteLn ('P=',P);

**End.**



решение

## РЕШЕНИЕ ЗАДАЧ

1. Задать случайным образом 10 чисел от 22 до 135. Напечатать те из них, которые больше 100.
2. Задать с клавиатуры A и N. Напечатать значение A в степени N.
3. Задать с клавиатуры N. Напечатать значение  $N! = 1*2*3*4*...*N$ .
4. Ввести с клавиатуры 5 значений для X. Вычислить и напечатать для каждого из них значение Y по правилу:

$3*X$ , если $X \leq -9$	$Y = X - 8$ , если $-9 < X < 9$	$3*X$ , если $X \geq 9$
$X/10$ , если $X \leq -5$	$Y = X * 10$ , если $-9 < X < 9$	$X/10$ , если $X \geq 9$

# Домашняя работа

1. Вводить с клавиатуры символы до тех пор, пока не будет введен пробел.
2. Определить, сколько было введено маленьких английских букв.
3. Изменить программу так, чтобы она находила количество введенных чисел.
4. Задать случайным образом 10 чисел. Найти их среднее арифметическое.