



ЯЗЫК  
ПРОГРАММИРОВА  
НИЯ **PASCAL**

- **Программа** — это упорядоченный список команд, необходимых для решения некоторой задачи.
- **Языком программирования** называют систему обозначений, служащую в целях точного описания алгоритмов для ЭВМ. Эти языки являются *искусственными языками* со строго определенным *синтаксисом*.



Интегрированная система **Турбо Паскаль** состоит из языка программирования и среды программирования. В эту систему входят:

- текстовый редактор;
- компилятор;
- компоновщик;
- загрузчик.



# АЛФАВИТ И НЕКОТОРЫЕ ОСНОВНЫЕ ОПЕРАТОРЫ ЯЗЫКА PASCAL

- Алфавит языка Паскаль включает *буквы, цифры, шестнадцатеричные цифры, специальные символы, пробелы и зарезервированные слова.*
- **Буквы** — это буквы латинского языка, а также знак подчеркивания. Паскаль не различает прописные и строчные буквы, если только они не используются в качестве данных программы.



- ❑ **Цифры** — арабские цифры от 0 до 9.
- ❑ Каждая шестнадцатеричная цифра имеет значение от 0 до 15. Первые 10 шестнадцатеричных цифр обозначаются арабскими цифрами от 0 до 9. Оставшиеся 6 латинскими буквами A .. F.
- ❑ Специальные символы:  
+ - \* / , ' . : ; <> [ ] { } ( ) ^ @ \$ # <> >= <= := (\* \*) (. .)
- ❑ **Пробелы** в Паскале являются разделителями.



*Идентификаторы — это имена констант, переменных, меток, типов, объектов, процедур, модулей, функций и других конструкций языка.*

- Длина идентификатора может быть произвольной, но значащими считаются первые 63 символа.
- Идентификатор состоит из любых букв латинского алфавита, цифр, знака подчеркивания.
- Идентификатор не может начинаться с цифры.
- Идентификатор не должен совпадать ни с одним из зарезервированных слов.



- ▣ **Переменные** можно сравнить с устройством, способным хранить элемент информации, например, число. Во время выполнения программы содержимое этого устройства может меняться.



## ▣ **Операция присваивания.**

В результате присваивания переменная получает конкретное значение.

*Имя Переменной := Выражение*

**:=** — оператор присваивания языка Паскаль.





*Выполнение инструкций присваивания заключается в следующем:*

- вычисляется результат выражения, находящегося справа от символа «:=»;
- вычисленное значение записывается в переменную, имя которой стоит слева от символа «:=».

**Пример.**

Counter:=0;

Pi:=3.1415926;

z:=(r1+r2)/(r2\*r1);

d:=b\*b-4\*c\*a;

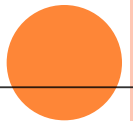


- ▣ **Выражение** состоит из **операндов и операторов**.  
Операторы находятся между операндами.
- ▣ **Операндами** выражений могут быть числа (константы) и переменные.
- ▣ **Операторы** обозначают действия, выполняемые над операндами.



# Арифметические операторы, которые применяются для записи алгебраических выражений на Паскале:

Оператор	Выполняемое действие	Пример использования	Примечание
+	Сложение	$A+3,4$	При $A=1$ выражение принимает значение 4,4
-	Вычитание	$A-3,4$	При $A=1$ выражение принимает значение -2,4
*	Умножение	$A*3,4$	При $A=1$ выражение принимает значение 3,4
/	Деление вещественных чисел	$A/3,4$	При $A=1$ выражение принимает значение приблизительно 0,294
Div	Целая часть от деления целого числа на целое число	a) $4 \text{ div } 2$ b) $5 \text{ div } 2$	Данная операция еще называется делением нацело, т.к. округляет результат деления путем отбрасывания дробной части. a) = 2 b) = 2
Mod	Остаток от деления	a) $4 \text{ div } 2$ b) $5 \text{ div } 2$	a) = 0 b) = 1



## Правила выполнения операторов:

- Операторы **\*,/, div, mod** имеют более высокий приоритет и выполняются раньше, чем операторы **+** и **-**.
- Если приоритет операторов одинаков, то оператор, стоящий левее, выполняется раньше.
- Для задания любого нужного порядка выполнения операций в выражениях можно использовать скобки.



**Пример.**

Математическая запись функции имеет вид:

$$y = \frac{2x}{x + 3}$$

Запись этой функции на Паскале:

**Y:=(2\*x)/(x+3)**



# СТРУКТУРА ПРОГРАММЫ НА ЯЗЫКЕ PASCAL

Любая программа (или программная единица — процедура или функция) на Паскале состоит из трех основных разделов:

- Раздел заголовка
- Раздел описаний
- Раздел исполняемых операторов (тело программы)



- ▣ *Заголовок* состоит из зарезервированного слова **program**, после которого следует *собственно имя программы*. В конце обязательно ставится знак **;**.
- ▣ *Описание переменных* начинается с зарезервированного слова **VAR**, а описание констант (если они используются) — **CONST**.
- ▣ *Тело программы* - раздел исполняемых операторов. Оформляется операторными скобками:  
**begin ... end**. Все операторы, реализующие алгоритм, должны помещаться между этими операторными скобками.



# СТРУКТУРА ПРОГРАММЫ PASCAL

ИМЕЕТ ВИД:

**Program** **ИмяПрограммы;**

**Var ...;**

**Begin**


**...**

**End.**





## ТИПЫ ДАННЫХ ЯЗЫКА ПАСКАЛЬ

- Для всех переменных в программе должен быть определен их тип.
- **Целый — Integer** — значением переменной этого типа может быть целое число из отрезка  $[-32768; 32767]$ .
- **Вещественный — Real** — значением переменной этого типа может быть вещественное число, модуль которого принадлежит отрезку  $[2,9 \cdot 10^{-39}; 1,7 \cdot 10^8]$ .
- **Логический — Boolean** — переменная этого типа может принимать только два значения: истина (**true**) или ложь (**false**).
- **Символьный — Char** — значением переменной этого типа может быть любой символ из набора ASCII-символов. 

<b>Стандартные математические функции Турбо Паскаля</b>			
<b>Обращение</b>	<b>Тип аргумента</b>	<b>Тип результата</b>	<b>Примечание</b>
ABS(X)	Real	Real	Модуль аргумента
arctan(x)	Real	Real	Арктангенс
cos(x)	Real	Real	Косинус
exp(x)	Real	Real	Экспонента ( $e^x$ )
frac(x)	Real	Real	Дробная часть
ln(x)	Real	Real	Натуральный логарифм
Random	Real	Real	Псевдослучайное число, равномерно распределенное в диапазоне [0,1]
Random(x)	Integer	Integer	Псевдослучайное число, равномерно распределенное в диапазоне [0,x-1]
Rendomize			Инициация генератора псевдослучайных чисел
Sin(x)	Real	Real	Синус (угол в радианах)
Sqr(x)	real, integer	real, integer	Квадрат аргумента ( $x^2$ )
Sqrt(x)	Real	Real	Корень квадратный $x$



▣ *Составной оператор* — это последовательность произвольных операторов программы, заключенная в операторные скобки — зарезервированные слова **Begin ... end**.

Begin

...

begin

...

...

end;

...

end

End.

Символ «**;**» называется пустым оператором и означает окончание предшествующего ему оператора.



# ОПЕРАТОРЫ

- Оператор вывода – **Write( );** или **Writeln( )**
- Оператор ввода – **Read( );** или **Readln( )**
- Оператор присваивания - **:=**
  
- Для вывода вещественного значения используют **форматы вывода.**

Например: Write(D:5:2 ), где первая цифра – количество знаков в числе, вторая – количество знаков после запятой.



# ЛИНЕЙНЫЙ АЛГОРИТМ

Задание. Составить программу.

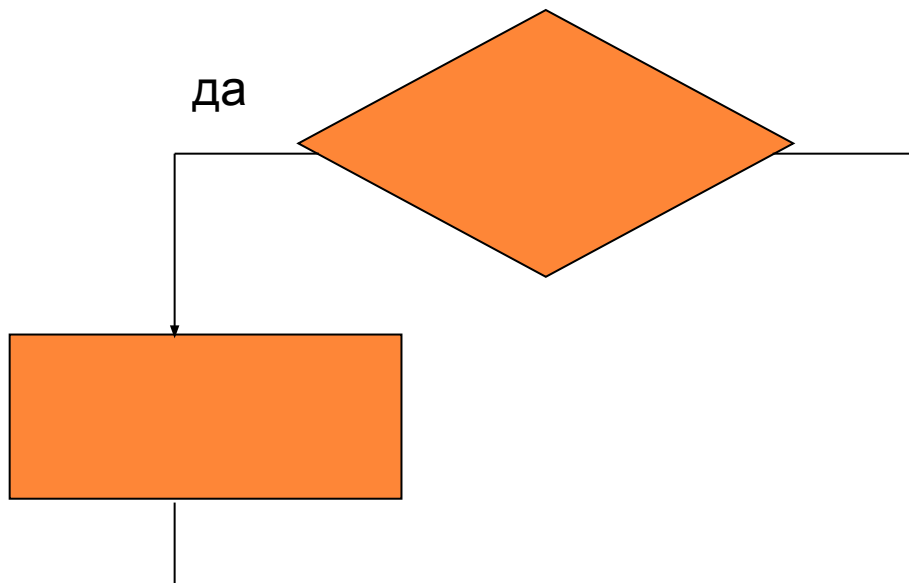
Даны переменные A и B. Требуется обменять их значения, т.е. переменная A должна получить значение B, а B — значение A.

```
Program Obmen;  
Var a,b,dop:integer;  
Begin  
Write('Введите значения двух переменных a,b ');  
{Значения можно вводить через пробел или запятую}  
Readln(a,b);  
dop:=a;  
a:=b;  
b:=dop;  
Writeln('a= ',b,'b= ',a);  
End.
```

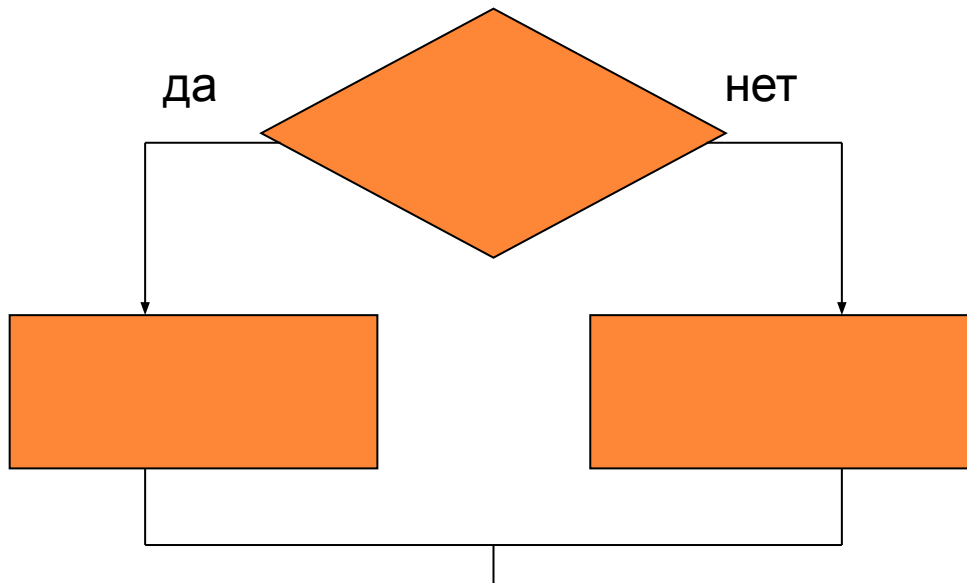


# ОПЕРАТОР УСЛОВНОГО ПЕРЕХОДА

□ **if** <условие> **then** <действие1>;



□ **if** <условие> **then** <действие1>  
**else** <действие2>;



# ВЕТВЯЩИЙСЯ АЛГОРИТМ

Задание. Составить программу.

Вычислить значение Y:

Y=

x-2, если  $10 \leq x$

x+2, если  $x < 10$

*Program Vetv1;*

*var x, y:real;*

*Begin*

*Write (x);*

*Readln(x);*

*if x<10 then y:=x+2*

*else y:=x-2;*

*Writeln(y);*

*End.*

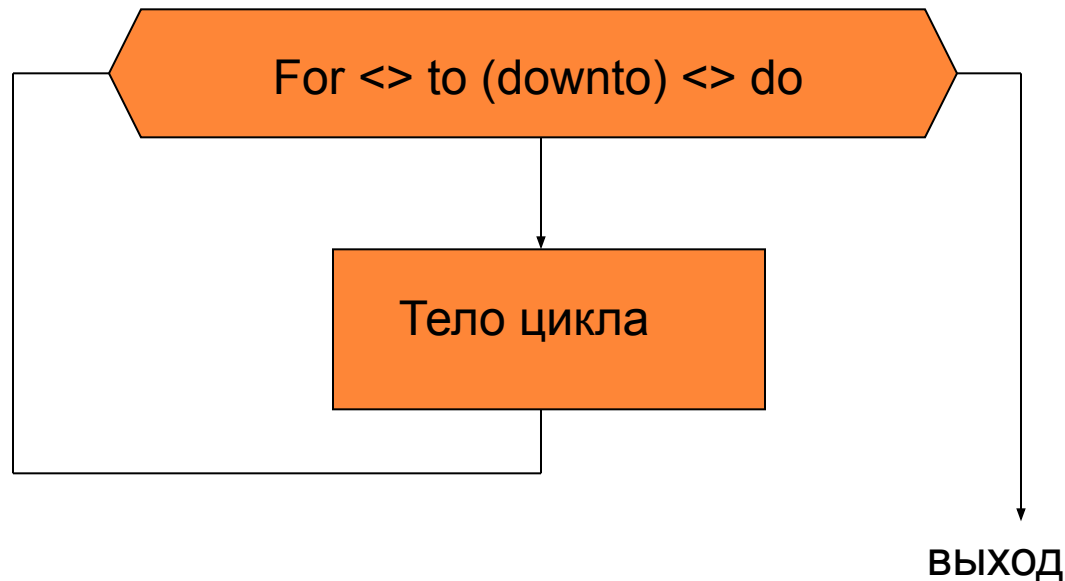




# ОПЕРАТОР ЦИКЛА С ПАРАМЕТРОМ

а) шаг равен +1

**For** <параметр> := <нач.знач.> **to** <кон.  
знач.> **do** <оператор>;



б) шаг равен -1

**For** <параметр>:=<нач.знач.>  
**downto** <кон.знач.> **do**  
<оператор>;



# ПАРАМЕТРОМ

## Пример.

Вычислить сумму натуральных чисел:  $S=1+2+3+\dots+N$ .

```
Program Sum1;  
  var n:integer;  
Begin  
  write(n);  
  readln(n);  
  S:=0;  
  for i:=1 do n  
    do S:=S+i;  
  writeln(S);  
End.
```



# ОПЕРАТОР ЦИКЛА С ПОСТУСЛОВИЕМ

**repeat**

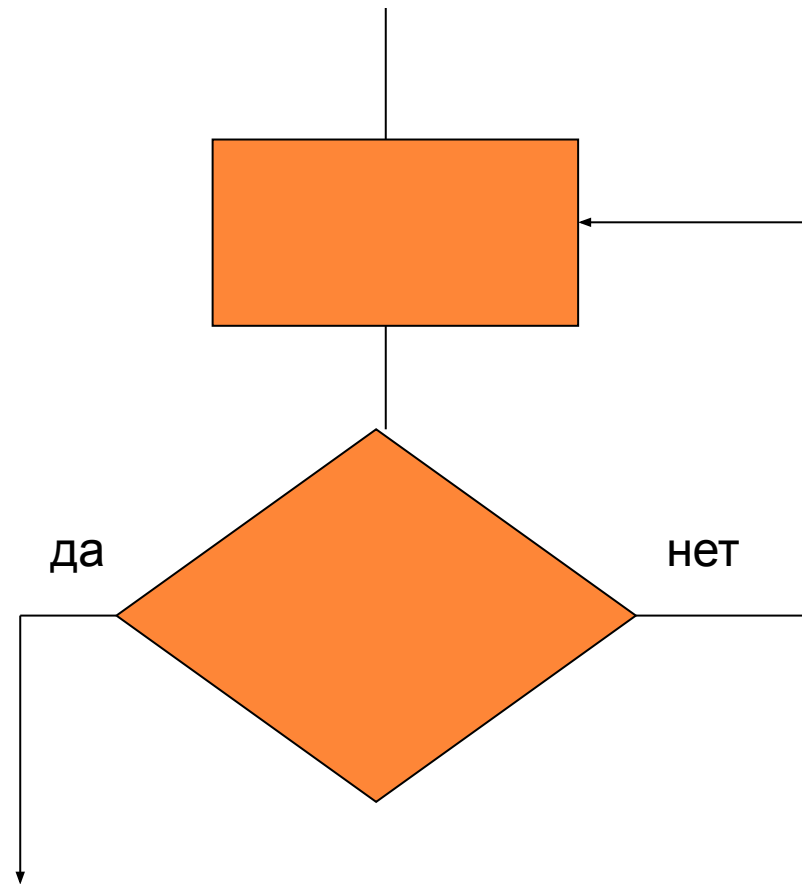
<оператор1>;

<оператор2>;

...

<операторN>

**until** <условие>;



# ОПЕРАТОР ЦИКЛА С ПОСТУСЛОВИЕМ

Пример. Составить программу, которая вводит символ и выводит на экран его код.

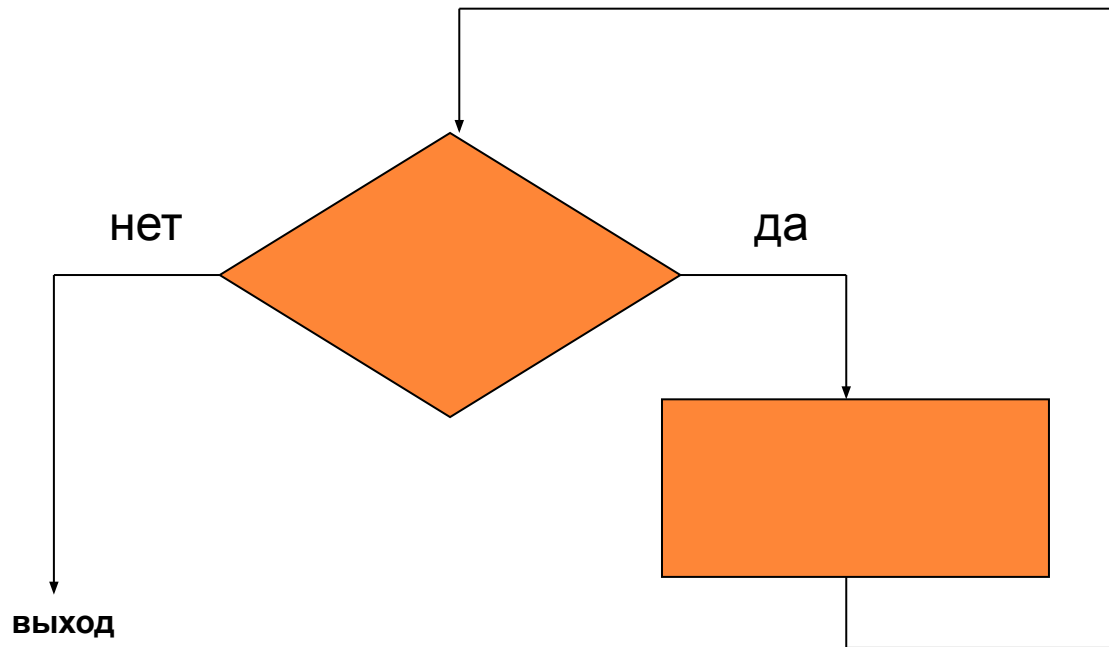
Функция **Ord(ch)** – возвращает код символа **ch** в ASCII-таблице

```
Program Symbol;  
    var ch:char;  
    const Cr=13;  
Begin  
    repeat  
    write(ch);  
    readln(ch);  
    writeln(Ord(ch));  
    until Ord(ch)=Cr;  
End.
```



# ОПЕРАТОР ЦИКЛА С ПРЕДУСЛОВИЕМ

**while** <условие> **do** <оператор>;



# ОПЕРАТОР ЦИКЛА С ПРЕДУСЛОВИЕМ

## Пример.

Дано натуральное число  $N$ . Просуммировать это число 5 раз.

```
Program Kol;  
    var k,N,S:integer;  
Begin  
    write (N);  
    readln(N);  
    S:=0;  
    k:=0;  
    while k<5 do  
        begin  
            k:=k+1; S:=S+N;  
        end;  
    writeln (S);  
End.
```

