

ГПОУ ТО «Крапивенский лесхоз-техникум»

Язык программирования Pascal

Презентация для студентов 1 курсов

Разработал преподаватель
информатики: Блажевич Л.С.

Структура программы

{1. заголовок программы}

program Имя_Программы;

{2. раздел указания используемых модулей}

uses Список_Используемых_Модулей;

{3. Раздел описаний}

label Описания_меток;

const Описания_Констант;

type Описания_Типов;

var Описания_Переменных;

procedure Описания_Процедур_и_функций;

function

exports Описания_Экспортируемых_Имен;

{4. Раздел операторов}

begin

 Операторы

end.

Упрощенная структура программы

{1. заголовок программы}

program Имя_Программы;

{2. раздел указания используемых модулей}

uses Список_Используемых_Модулей;

{3. Раздел описаний}

const Описания_Констант;

var Описания_Переменных;

{4. Раздел операторов}

begin

 Операторы программы

end.

- Команды языка программирования называются операторами
- Разделителем операторов в Паскале является ; (точка с запятой)

Комментарий представляет собой фрагмент текста программы, ограниченный символами ***{ }***.
Комментарии в программе выполняют информационную функцию.

{Моя первая программа }

```
program first;  
begin  
    writeln('Hello, World!')  
end.
```

Пример: периметр прямоугольника

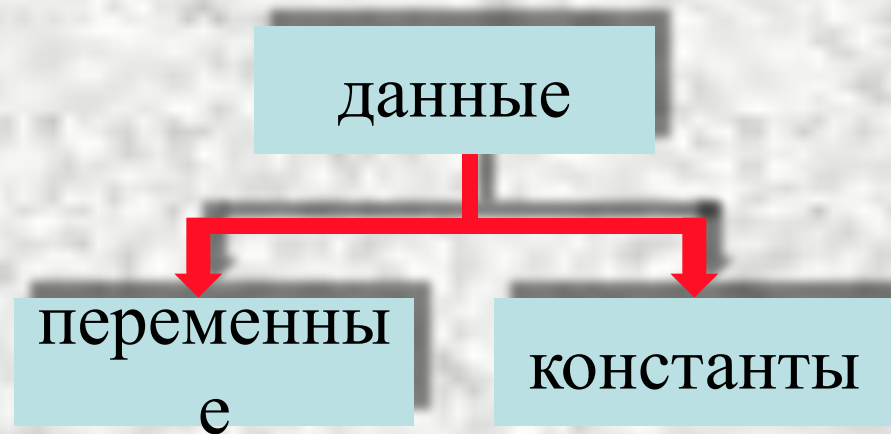
```
program perimetr;           {заголовок программы}
var a,b:integer;           {объявление переменных}
    P:integer;
Begin                     {начало программы}
    clrscr;                 {очистка экрана}
    a:=12; {присваиваем переменной a значение 12}
    b:=7;  {присваиваем переменной b значение 7}
    P:=2*(a+b); {значение выражения присваиваем P}
    write('P = ',P);        {выводим на экран значение P}
end.                      {конец программы}
```

Программа в своей работе имеет дело с данными.

Некоторые данные устанавливаются еще до того, как программа начнет выполняться, а после ее запуска сохраняют свои значения неизменными на всем протяжении работы программы. Это ***константы***.

Другие данные могут изменяться во время выполнения программы. Они называются ***переменными***.

- ✓ Различие между **переменной** и **константой** довольно очевидно: во время выполнения программы значение переменной может быть изменено, а значение константы нет.



Данные бывают различных
ТИПОВ...

Стандартные типы данных

1. группа целых типов (Shortint, Integer, Longint, Byte, Word);
2. группа действительных типов (Single, Real, Double, Extended, Comp);
3. логические (булевские типы) (Boolean, ByteBool, WordBool, LongBool);
4. символьный (Char);
5. строковый (String);
6. указательный (Pointer);
7. текстовый тип (Text).

Группа целых типов

Название типа	Тип	диапазон значений	требуемая память
Короткое целое со знаком	Shortint	-128 .. 127	1 байт
Целое со знаком	Integer	-32768 .. 32767	2 байта
Длинное целое со знаком	Longint	-2147483648 .. 2147483647	4 байта
Короткое целое без знака	Byte	0 .. 255	1 байт
Целое без знака	Word	0 .. 65535	2 байта

Группа вещественных типов

Название типа	Тип	Диапазон значений	Количество цифр мантиссы	Размер (байт)
Вещественное число одинарной точности	Real	$2.9e^{-39} \dots 1.7e^{+38}$	11	6
Вещественное число	Single	$1.5e^{-45} \dots 3.4e^{+38}$	7	4
Вещественное число двойной точности	Double	$5.0e^{-324} \dots 1.7e^{+308}$	15	8
Вещественное число повышенной точности	Extended	$3.4e^{-4932} \dots 1.1e^{+4932}$	19	10
Целое число в формате вещественного	Comp	$-9.2e^{+18} \dots 9.2e^{+18}$	19	8

Действительные числа записываются в форме с десятичной **точкой** или в форме с использованием десятичного порядка, который изображается буквой **E**:

28.6 0.65 -0.018 4.0

5E12 -1.72E9 73.1E-16

Булевскому типу данных соответствует идентификатор **Boolean**. Переменные булевского типа имеют размер **1 байт** и могут содержать значения **TRUE** или **FALSE**.

Значению FALSE соответствует 0, любое число отличное от нуля считается TRUE.

```
Var кнопка, flag : boolean;
```

```
Begin
```

```
    кнопка:=true;
```

Строковому типу соответствует стандартный идентификатор **String**.

Var

S : String;

{строка от 0 до 255 символов}

S2: String[5];

{стока из 5-ти символов}

Строка символов представляет собой последовательность символов из набора символов кода ASCII, заключенную в одиночные кавычки.

Строки в языке ПАСКАЛЬ - это последовательность символов, записанная между апострофами. Если в строке в качестве содержательного символа необходимо употребить сам апостроф, то следует записать два апострофа.

Примеры строк:

'СТРОКА' 'STRING' 'АД"ЮТАНТ'

Стандартный ввод и вывод

- ✓ Осуществляется встроенными процедурами
- ✓ ***Read***(список переменных); – чтение значений, введенных с клавиатуры
- ✓ ***Readln***(список переменных); - ln в конце означает переход на новую строку
- ✓ ***Write***(список переменных); - вывод на экран
- ✓ ***Writeln***(список переменных);

Пример

```
Program primer;  
Uses crt;  
Var a,b,summa:integer;  
Begin  
    clrscr;  
    write('Введите 2 целых числа: ');  
    readln(a,b);  
    Summa:=a+b;  
    write('Сумма = ',summa);  
End.
```

Классификация операций

1. арифметические операции:
 - унарные: +, -
 - бинарные: +, -, *, /, div, mod
2. операции отношения: =, <>, <, >, <=, >=
3. булевские (логические) операции:
not (логическое отрицание), **and** (лог. И), **or** (лог. ИЛИ), **xor** (исключающее ИЛИ)
4. строковая операция (конкатенация) +

div – частное от деления

$$7 \text{ div } 3 = 2$$

A:=10 div 3;

mod – остаток от деления

$$7 \text{ mod } 3 = 1$$

A:=25 mod 10;

Оператор ветвления на Паскале



Операции отношения

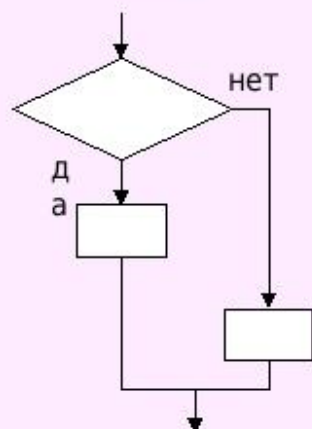
>	больше
<	меньше
>=	больше или равно
<=	меньше или равно
<>	не равно
=	равно

Сложное
условие

Логические операции

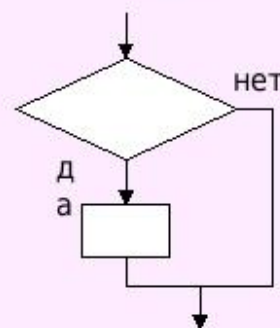
AND OR NOT

Полное ветвление



IF условие
THEN
оператор 1
ELSE
оператор 2

Неполное ветвление



IF условие
THEN
оператор

Операторы в структуре ветвления могут быть простыми или составными

BEGIN Последовательность простых и (или) составных операторов END

Сначала проверяется условие
ЕСЛИ условие истинно,
ТО выполняется оператор 1
ИНАЧЕ (если оно ложно)
выполняется оператор 2

x	y	x AND y	x OR y	NOT x
false	false	false	false	true
false	true	false	true	true
true	false	false	true	false
true	true	true	true	false

В сложных логических выражениях отношения, связываемые логическими операциями, заключаются в скобки !!! Например, требуется определить, есть ли среди чисел a, b, c хотя бы одно отрицательное. Эту задачу решает следующий условный оператор:

```
IF (a < 0) OR (b < 0) OR (c < 0) THEN WRITE (' да ') ELSE WRITE (' нет ');
```



Цикл с предусловием (цикл-пока)

While <условие> *do* <тело цикла>;

Пока условие – истинно, выполняется тело цикла. Тело цикла может быть простым или составным оператором.

Цикл с постусловием (цикл-до)

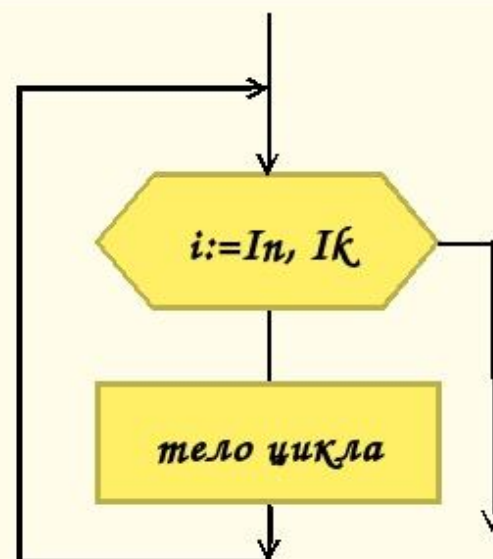
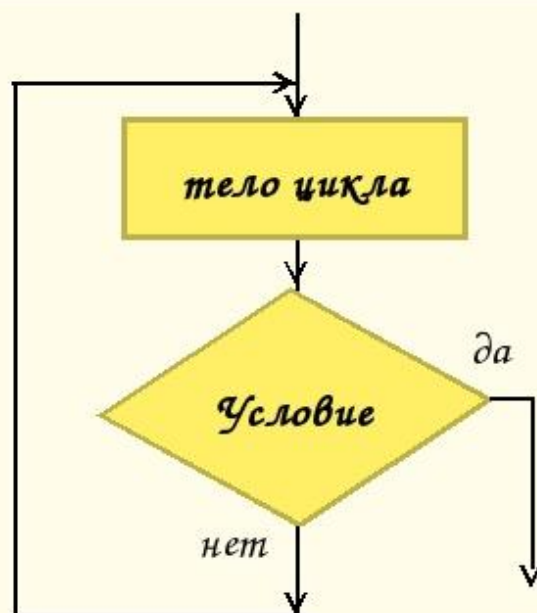
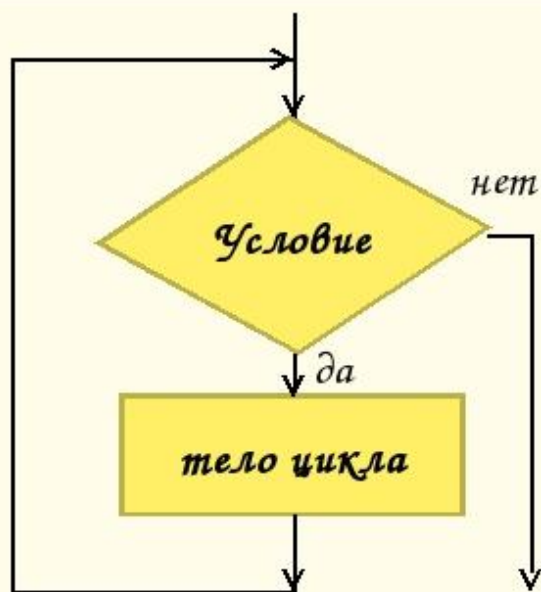
Repeat <тело цикла> *until* <условие>;

Повторяется выполнение тела цикла до истинности условия. Тело цикла с постусловием выполняется хотя бы один раз.

Цикл с параметром (цикл-до)

1. *for* $i := I_n$ *to* I_k *do* <тело цикла>;
2. *for* $i := I_n$ *downto* I_k *do* <тело цикла>;

Цикл повторяется, пока значение параметра лежит в интервале между I_n и I_k .



ЗАДАЧА

ВЫЧИСЛИТЬ СУММУ НАТУРАЛЬНОГО РЯДА ОТ 50 ДО 100

Цикл с предусловием

```
Program SNW;  
VAR K,S:integer;  
BEGIN  
  
K:=50;  
S:=0;  
WHILE K<=100 DO  
  BEGIN  
    S:=S+K;  
    K:=K+1;  
  END;  
WRITE(S);  
END.
```

Цикл с постусловием

```
Program SNR;  
VAR K,S:integer;  
BEGIN  
  
K:=50;  
S:=0;  
REPEAT  
  
  S:=S+K;  
  K:=K+1;  
UNTIL K>100;  
WRITE(S);  
END.
```

Цикл с параметром

```
Program SNF;  
VAR K,S:integer;  
BEGIN  
  
S:=0;  
FOR K=50 TO 100 DO  
  
  S:=S+K;  
  
WRITE(S);  
END.
```