

Операторы Pascal

Операторы Pascal

Операторы записываются в разделе операторов и отделяются друг от друга **точкой с запятой**:

Пример:

```
оператор1;  
оператор2;  
...;  
оператор3;
```

Операторы Pascal делятся на:

- **простые;**
- **структурированные.**

Простые операторы:

Оператор присвоения: **:=**

Формат записи:

имя_переменной := значение ;

Примеры:

a := 3;

b := 2 + 3;

c := a + b;

где: a, b и c – переменные.

Примечания:

- Двоеточие и равно пишутся **СЛИТНО**, без пробела;
- Нельзя присвоить значение константе внутри программы;
- При присвоении значений переменным необходимо следить за типом переменных.

Простые операторы:

Оператор ввода данных:

Read, Readln

Формат записи:

Read(имя_переменной, имя_переменной1);

Readln(имя_переменной, имя_переменной1);

Примеры:

Read(a);

Read(a, d, c);

Readln(c, x);

Простые операторы:

Оператор вывода данных:

Write, Writeln

Формат записи:

Write(имя_переменной, имя_переменной1);

Writeln(имя_переменной, имя_переменной1);

Примеры:

Write(a);

Write(a, d, c);

Writeln(c, x);

Простые операторы:

- Пустой оператор: **;**
 - Не выполняет никаких действий и ставится на то место, где может стоять быть любой другой оператор.
- Оператор перехода: **GoTo имя_метки;**
 - осуществляет переход к указанной метке в программе.

Структурированные операторы:

Составной оператор (операторные скобки):

- Содержит произвольное количество любых операторов, отделенных друг от друга точкой с запятой и ограниченных операторными скобками

Begin End

Пример:

```
Begin  
    оператор1;  
    оператор2;  
    .....;  
    операторN;  
End
```

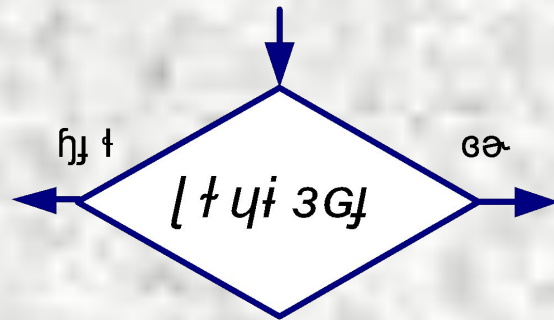
Примечание:

- после завершающего оператора, т.е. перед **END**, точку с запятой можно не ставить.
- может содержать внутри себя вложенные составные операторы.

Структурированные операторы:

Условный оператор:

IF логическое_условие **THEN** оператор_если_истина
[**ELSE** оператор_если_ложь] ;



Примечания:

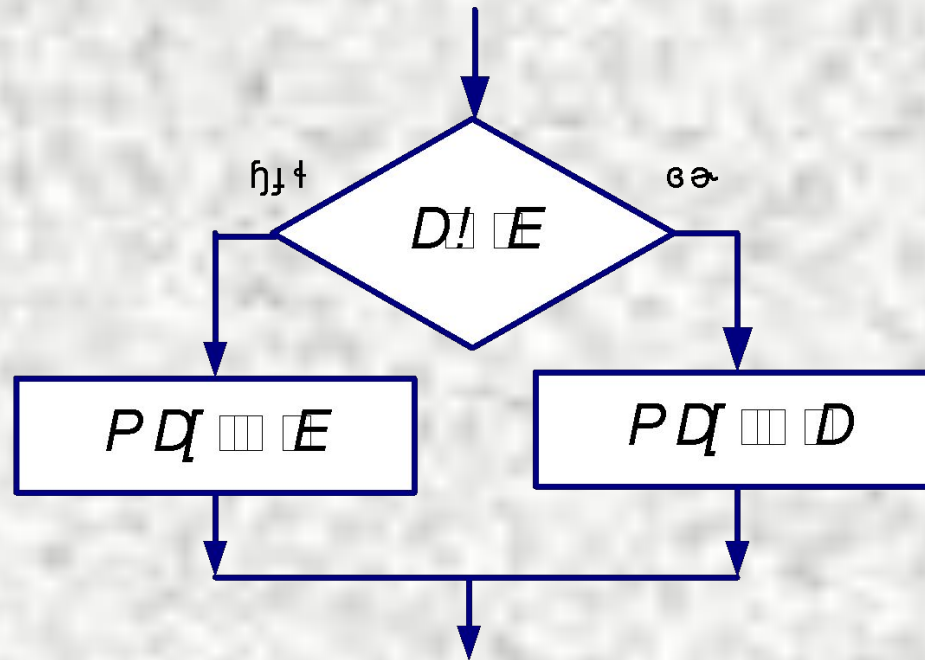
- Несколько логических условий **закljučаются в скобки** и отделяются друг от друга логическими функциями: **and** или **or**;
- После **then** или **else** может **выполняться только ОДИН** оператор;
- Если же необходимо выполнить несколько действий – используются операторные скобки (**begin...end**)
- Часть **else** условного оператора может и отсутствовать.

Структурированные операторы:

Условный оператор:

IF условие **THEN** оператор_истина **ELSE** оператор_ложь;

Пример: Даны два числа, найти максимальное из них



if a>b then max := a else max := b ;

Структурированные операторы:

Условный оператор:

IF условие **THEN** оператор_истины **ELSE** оператор_ложь;

Пример: Даны три числа. Проверить условие $a < b < c$

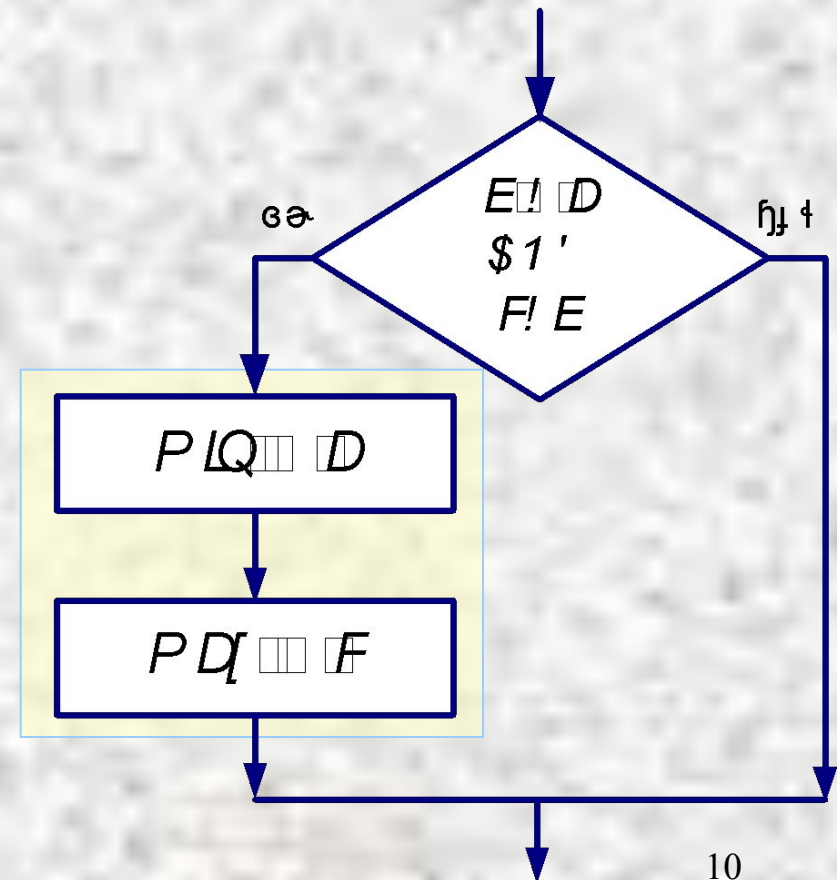
if (b>a) and (c>b)
then

begin

min := a;

max := c

end;



Цикл FOR:

FOR $i := A$ to B do оператор;

где: i - параметр цикла;

A - начальное значение параметра цикла;

B - конечное значение параметра цикла;

при этом $A < B$, **ШАГ** изменение параметра цикла РАВЕН +1

FOR $i := A$ downto B do оператор;

где: i - параметр цикла;

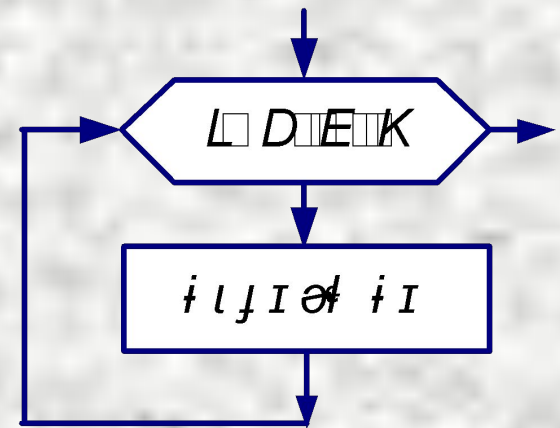
A - начальное значение параметра цикла;

B - конечное значение параметра цикла;

при этом $A > B$, **ШАГ** изменение параметра цикла РАВЕН -1

Примечания:

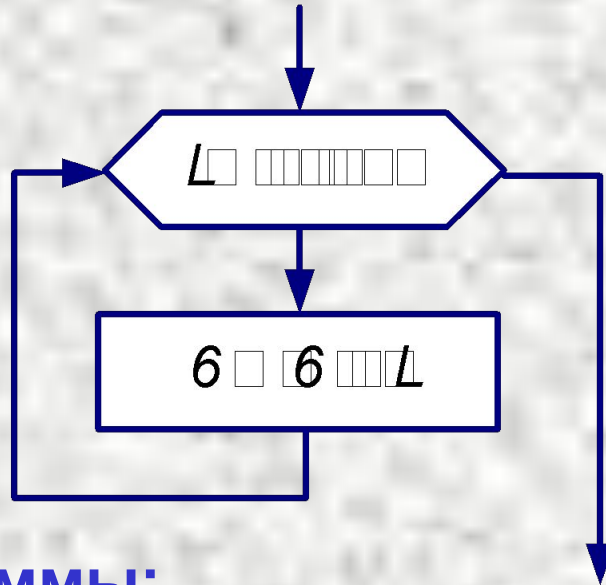
- Параметр цикла должен быть целым, порядкового типа;
- Изменять параметр цикла внутри цикла **НЕ ДОПУСКАЕТСЯ**
- В качестве оператора может **выполняться только ОДИН** оператор;
- В случае необходимости выполнения внутри цикла нескольких операторов - они заключаются в операторные скобки **begin...end**;¹¹



Оператор FOR ... to ... do

Пример: Найти сумму цифр от 1 до 100

Блок-схема:



Часть программы:

```
FOR i := 1 to 100 do S := S + i;
```


Цикл с ПРЕДУСЛОВИЕМ:

While условие **do** оператор ;

Примечания:

- Цикл будет выполняться

ПОКА ВЫПОЛНЯЕТСЯ УСЛОВИЕ;

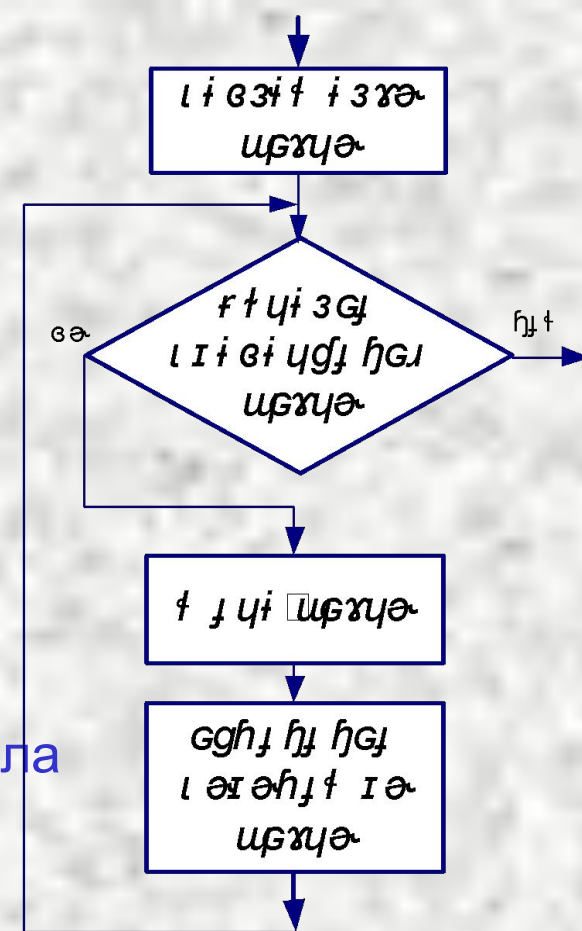
- Рано или поздно условие должно выполниться, иначе этот цикл станет бесконечным;

- В случае необходимости выполнения внутри цикла нескольких операторов - они заключаются в операторные скобки **begin...end;**

- Начальное значение параметра цикла необходимо задавать до цикла;

- Параметр цикла необходимо менять внутри цикла;

- Шаг цикла может быть любым числом.

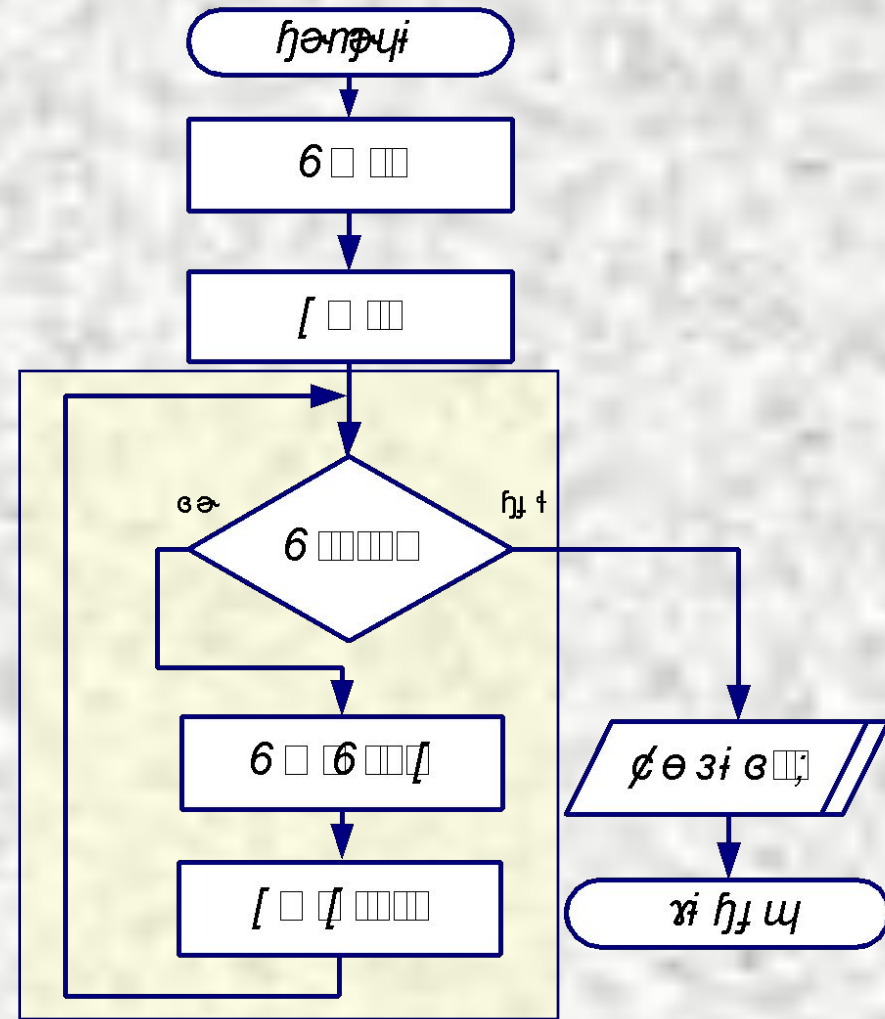


Оператор While ... do ...

Пример:

Найти число, на котором
сумма чисел от 1 до этого числа
превысит 55.

Блок-схема:



Часть программы:

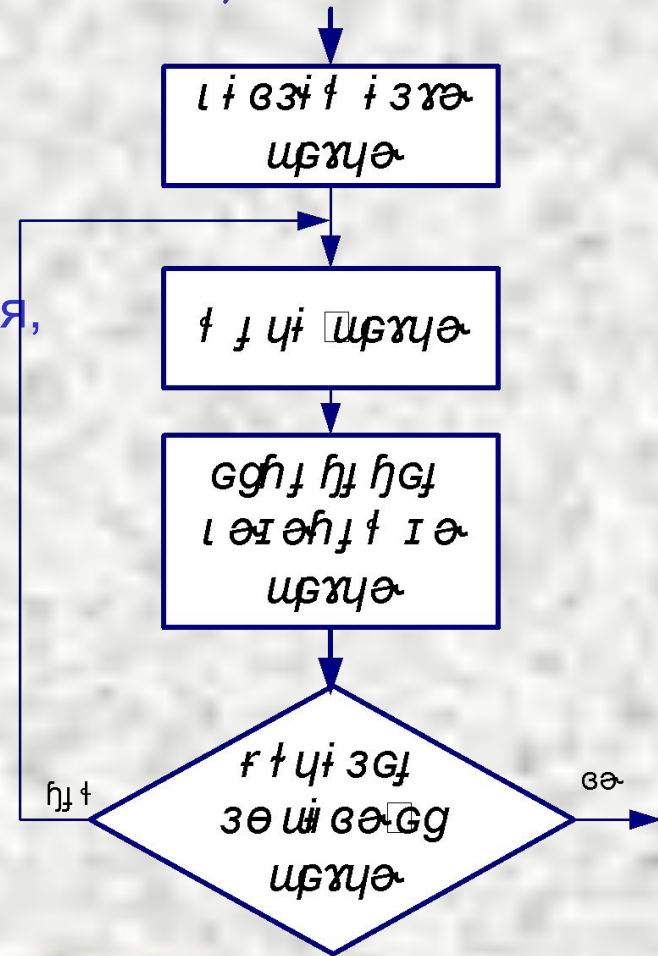
```
While S < 55 do begin S := S + x; x := x + 1 end;
```

Цикл с ПОСТУСЛОВИЕМ:

Repeat группа операторов **until** условие;

Примечания:

- Цикл будет выполняться **ДО ВЫПОЛНЕНИЯ УСЛОВИЯ;**
- Рано или поздно условие должно выполниться, иначе этот цикл станет бесконечным;
- В качестве тела цикла могут выступать несколько операторов, разделенных точкой с запятой;

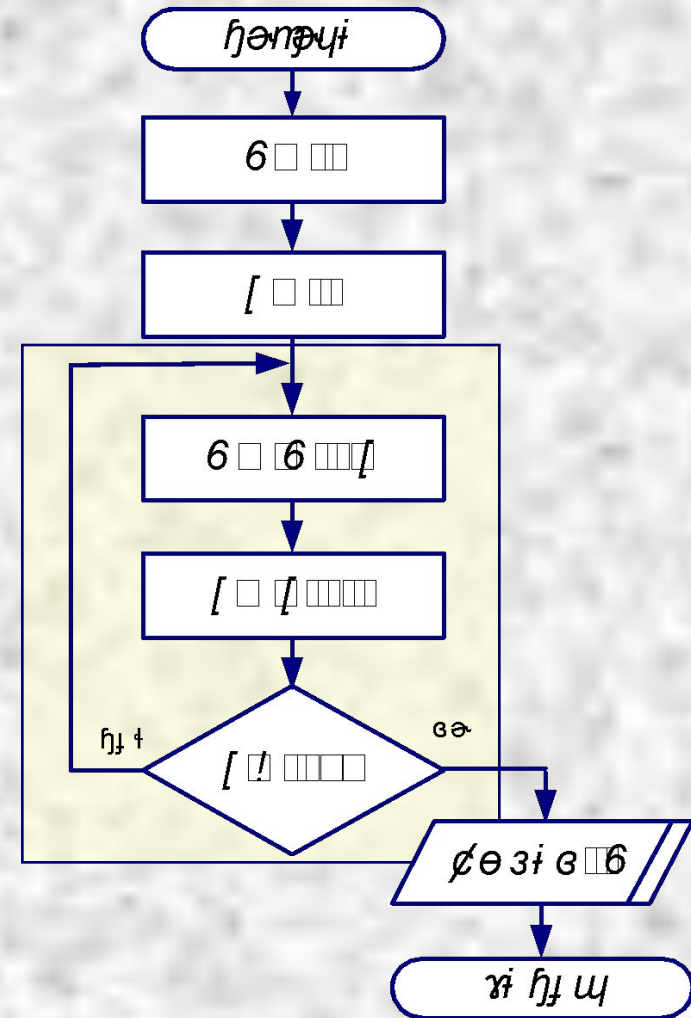


Оператор Repeat ... until ...

Пример:

Вычислить сумму нечетных чисел от 1 до 101.

Блок-схема:



Часть программы:

```
Repeat S:=S+x; x:=x+2 until x>101;
```