

Условный оператор



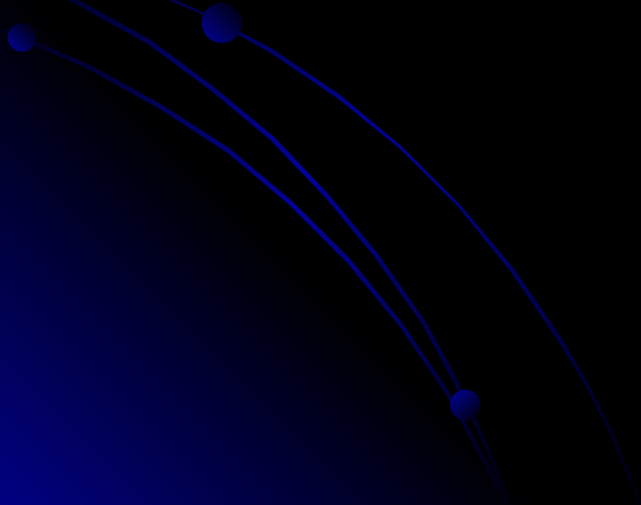
Определение линейного алгоритма.

- Линейный алгоритм – это алгоритм, этапы которого выполняются однократно и строго последовательно.

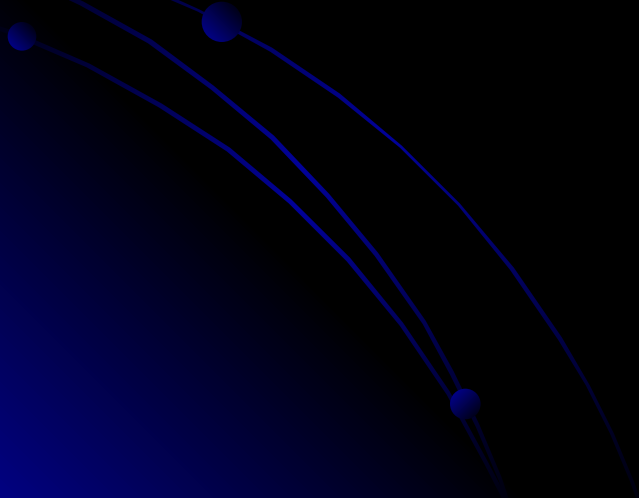


Что такое ветвление?

- **Разветвляющийся алгоритм – это алгоритм, в котором в зависимости от условия выполняется либо одна, либо другая последовательность действий.**



Примеры алгоритмов ветвления



Пример алгоритма ветвления

Требуется построить алгоритм вычисления значения функции $y=|x|$.

Она создается соотношением:

$$y = \begin{cases} x & \text{при } x \geq 0; \\ -x & \text{при } x < 0 \end{cases}$$

При решении этой задачи требуется выполнить следующие условия:

1. проверить больше или равен нулю x ;
2. если x больше или равен нулю, то присвоить y значение x ($y=x$),
если x меньше нуля, то присвоить y значение $-x$ ($y=-x$).

Короткая запись

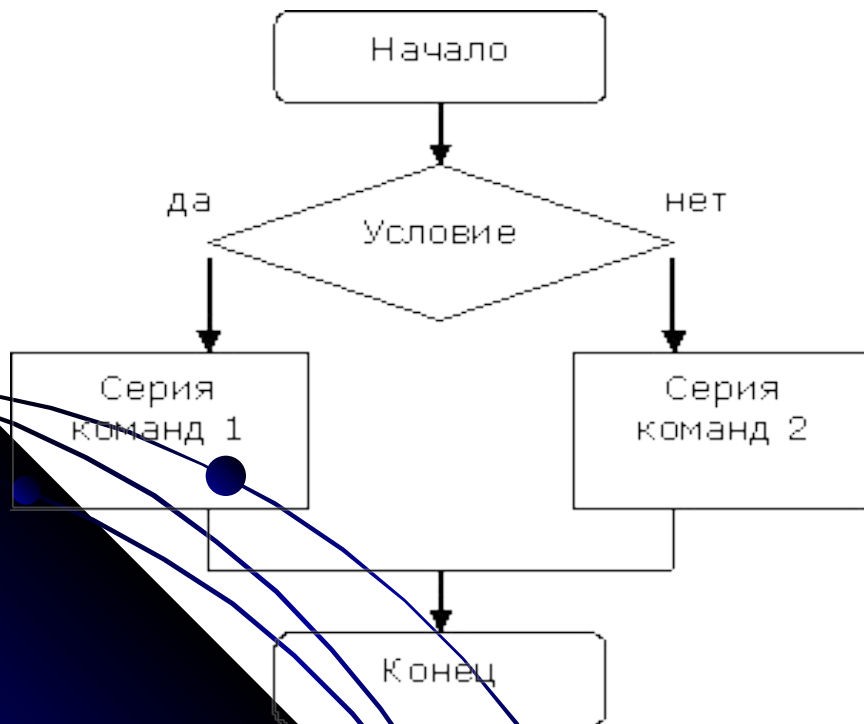
если $x \geq 0$,
то $y := x$,
иначе $y := -x$;

Команды, с помощью которых
записываются алгоритмы подобного
типа (разветвляющиеся алгоритмы)
называются командами ветвления.

Команде ветвления в Паскале соответствует *условный оператор*.

Условный оператор имеет две формы записи:

1. Полная форма условного оператора



- *В случае выполнения условия (выход «да» из ромба) выполняется одно действие, в случае невыполнения (выход «нет») – другое действие.*

2. Неполная форма ветвления.

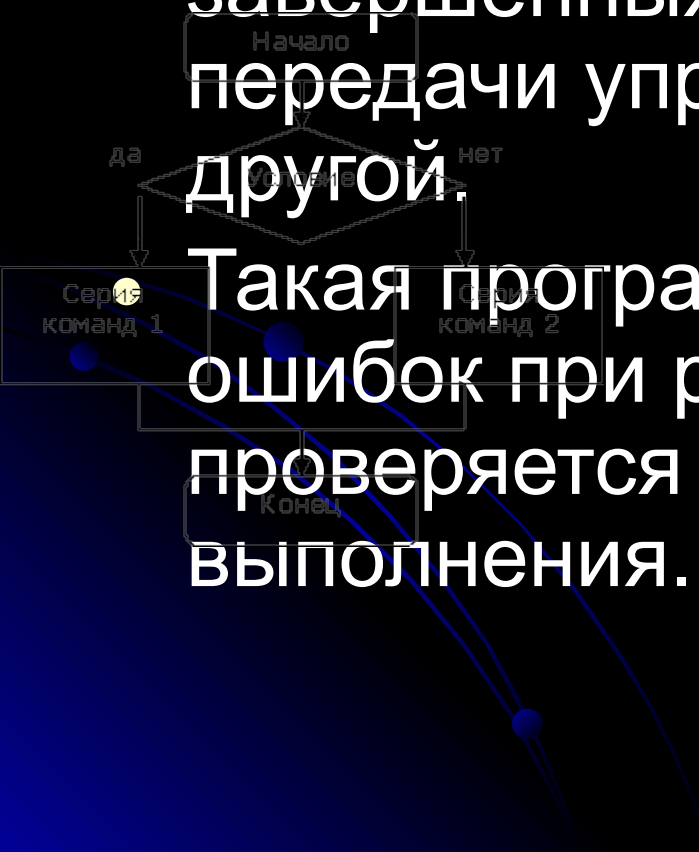
- Выполняется только тогда, когда выполняется записанное в ромбе условие.
- В случае невыполнения условия происходит переход к следующему оператору (выход из структуры)



Следует помнить!

- Программу рекомендуется строить из последовательных, логически завершенных блоков, не допуская передачи управления из одного блока в другой.

Такая программа содержит меньше ошибок при разработке, легче проверяется на правильность выполнения.



- **Неполный условный оператор имеет вид:**

IF <условие> THEN <ОПЕРАТОР>

- **Полный условный оператор имеет вид:**

**IF <условие> THEN <ОПЕРАТОР_1> ELSE
<ОПЕРАТОР_2>**

- **На русском языке (перевод):
ЕСЛИ условие, ТО оператор_1, иначе
оператор_2.**

- В различных случаях после слов **THEN** и **ELSE** надо выполнить не один оператор, а несколько. Тогда эти операторы заключаются в так называемые операторные скобки, открывающаяся скобка которых – **BEGIN**, а закрывающая **END**:

begin

(операторы)

end;

Перед словом **ELSE** точка с запятой не ставится.

В операторных скобках рекомендуется на каждую пару BEGIN – END записывать в одном столбце: так легче проверить соответствие каждой открывающейся скобке закрывающую.

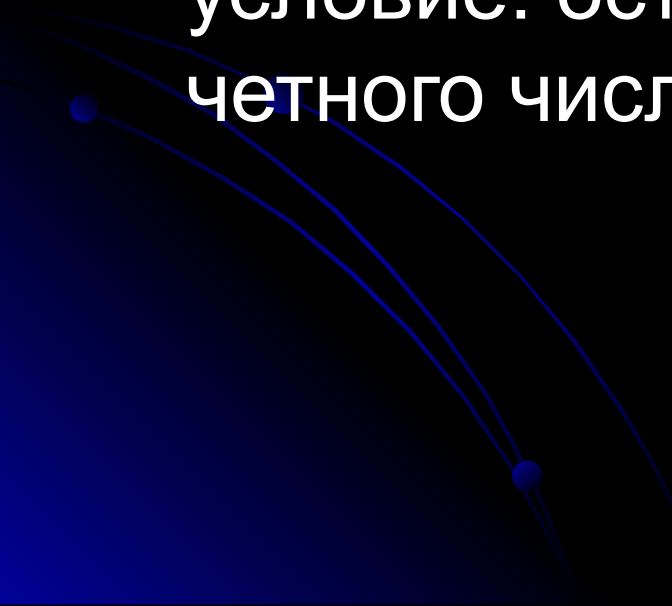
- `if a<b then y:=x;`
- `if x<0 then x:=-x;` {изменение знака переменной x}
- `if a+b<c then begin`
 - `z:=x;` {обмен значениями переменными x и y}
 - `x:=y;`
 - `y:=z;`
 - `end;`

- В качестве выполняемого в условном операторе действия может быть другой условный оператор. Например:

```
if sqr (x) + sqr (y) >1 then  
    if x>y then z:=0  
    else z:=1;
```

При такой записи, использующей сдвиг вправо для каждого внутреннего действия, легко понять, к какому из двух слов IF относится слово ELSE. Если этот оператор записать в одну строку, то ответ будет неоднозначным.

Пример программы с использованием условного оператора.

- Пусть для двух целых чисел надо определить, являются ли они четными или нет.
 - Для проверки четности используем условие: остаток от деления на 2 четного числа равен 0.
- 

Пример программы с использованием условного оператора.

Program E3;

var **a,b**: integer;

begin

writeln ('vvedite dva zelih chisla');

readln (**a,b**);

if **a** mod 2 = 0 then writeln ('**a** – chetnoe')

else writeln ('**a** – nechetnoe');

if **b** mod 2 = 0 then writeln ('**b** – chetnoe')

else writeln ('**b** – nechetnoe');

end.

P.s.: mod – остаток: $7:2=3$ в остатке 1 или $7\text{mod}2=1$

div – целое число: $7\text{div}2=3$.

Домашнее задание.

1. Записи в тетради.
 2. Попробовать составить алгоритм с условным оператором: присвойте величине **a** значение наибольшего из трех заданных чисел.
- 