

Условный оператор

Рассмотрим простой пример задачи из курса алгебры. Требуется построить алгоритм вычисления значения функции $y=|x|$. Она задается 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:
проверить больше или равен нулю x
если x больше или равен 0, то присвоить y
значение x , если меньше 0, то присвоить y
значение $-x$*

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

Коротко алгоритм задачи может быть записан:

*ЕСЛИ $x \geq 0$ ТО $y := x$
ИНАЧЕ $y := -x$*

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

ЕСЛИ $x \geq 0$ ТО

у:=х

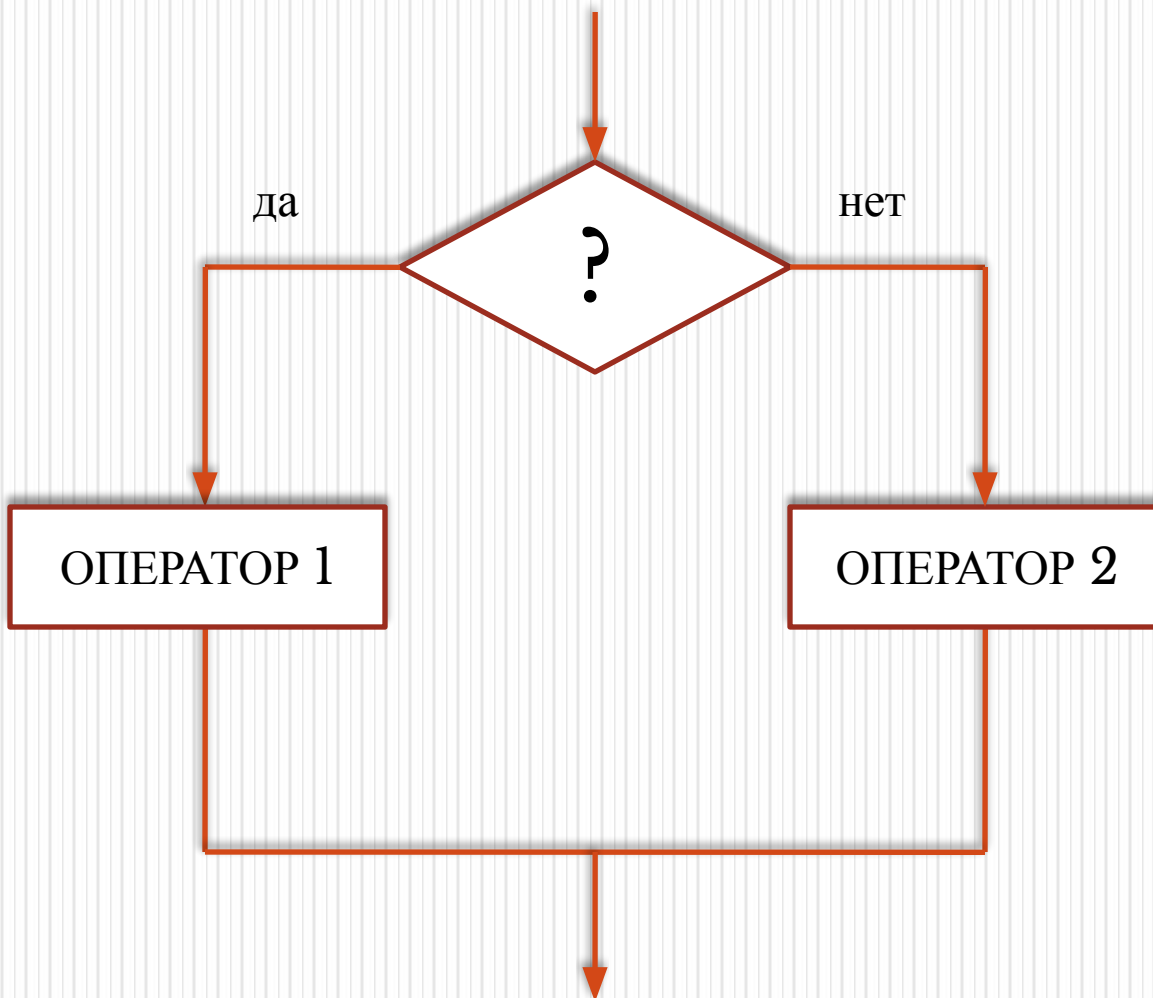
ИНАЧЕ

у:=-х

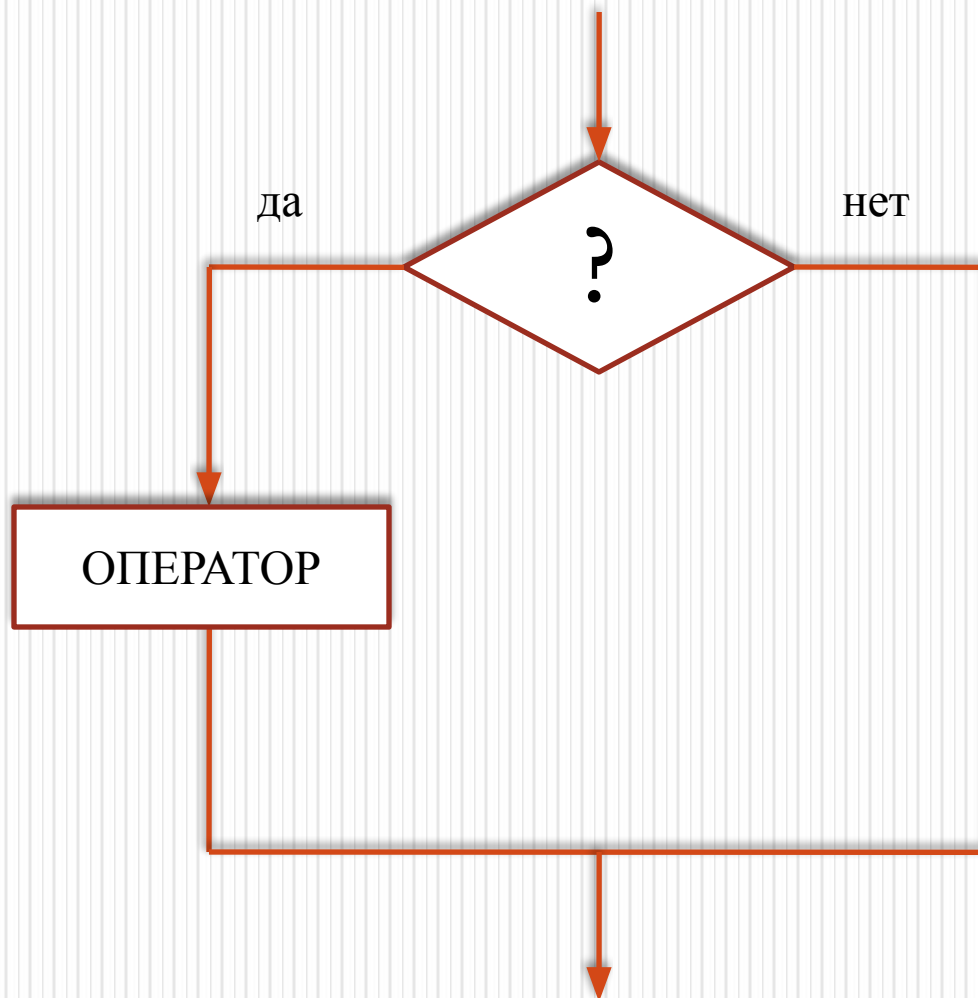
Команды, с помощью которых записываются алгоритмы подобного типа, называются командами ветвления. Командам ветвления в Паскале соответствует условный оператор.

*Условный оператор
может иметь две
формы , полную и
неполную.*

ПОЛНАЯ ФОРМА



НЕПОЛНАЯ ФОРМА



В Паскале:

неполный условный оператор имеет вид:

IF (условие) THEN (оператор 1);

полный оператор имеет вид:

***IF (условие) THEN (оператор 1)
ELSE (оператор 2);***

Пример программы:

```
Program uslov;
```

```
Var a, b: integer;
```

```
Begin
```

```
  Writeln('vvedite 2 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');
```

```
  Readln;
```

```
  Readln
```