



ОСНОВЫ ПРОГРАММИРОВАНИЯ

Учитель информатики и ИКТ
ГООУ г.Москвы СОШ №310
«У Чистых прудов»
Цыбикова Т.Р.



Тема 4.

УСЛОВНЫЙ ОПЕРАТОР





Условный оператор

- В рассмотренных до сих пор алгоритмах и программах все команды (операторы) выполнялись последовательно одна за другой в том порядке, в каком они были записаны. Однако таким образом может быть построен алгоритм для решения далеко не всякой задачи.
- В практике хорошо известны задачи, дальнейший ход решения которых зависит от выполнения какого-либо **условия**.



Рассмотрим простой пример из курса алгебры.

- Требуется построить алгоритм вычисления значения функции $y=|x|$. Она задаётся соотношением:

$$y = |x| \Leftrightarrow y = \begin{cases} x \text{ при} & x \geq 0, \\ -x \text{ при} & x \leq 0. \end{cases}$$



$$y = \begin{cases} x \text{ при} & x \geq 0, \\ -x \text{ при} & x \leq 0. \end{cases}$$

- При решении этой задачи требуется выполнить следующие условия:
 - 1) проверить больше или равен нулю x ;
 - 2) если x больше или равен 0, то присвоить y значение x ($y:=x$), если x меньше 0, то присвоить y значение $-x$ ($y:=-x$)
- Коротко алгоритм решения этой задачи может быть записан так:

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



Команды ветвления

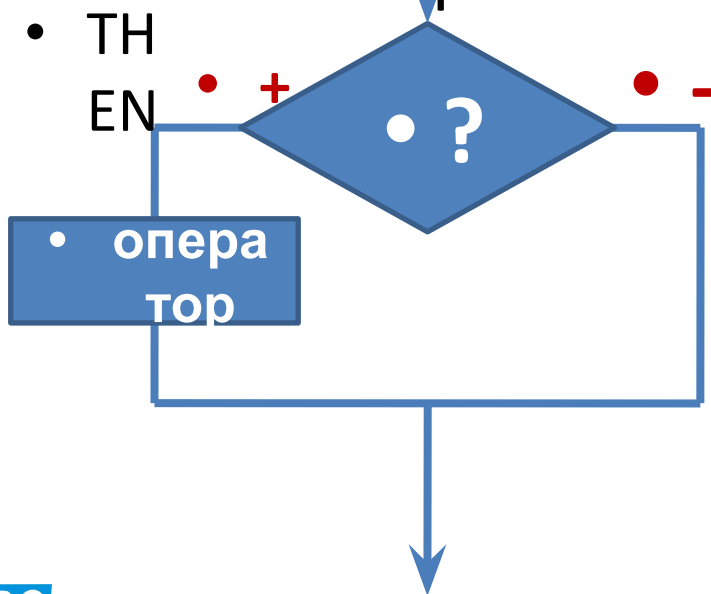
- Команды, с помощью которых записываются алгоритмы подобного типа (разветвляющиеся алгоритмы), называются **командами ветвления**.
- Команде ветвления в Паскале соответствует **условный оператор**.
- Условный оператор может иметь две формы (структуры), представленные на **рисунках 1 и 2**.



Рисунок 1

- Неполная форма

- условного оператора



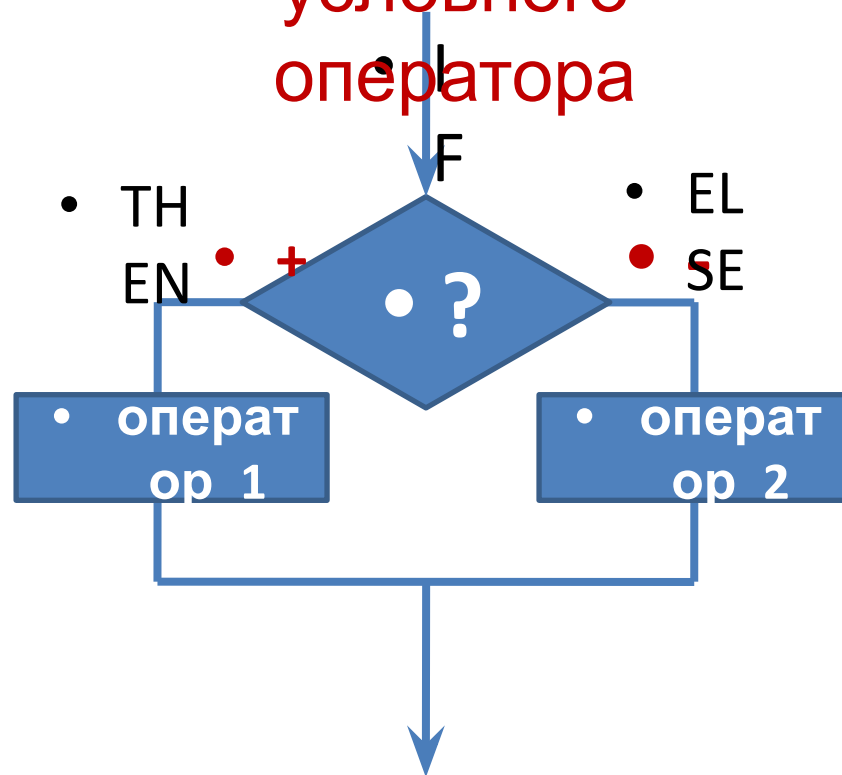
- На рис.1 показана неполная форма условного оператора: действие выполняется тогда и только тогда, когда выполняется записанное в ромбе условие. В случае невыполнения условия происходит переход к следующему оператору (выход из структуры).



Рисунок 2

- На рисунке 2 изображена полная форма условного оператора: в случае выполнения условия (выход «+» из ромба) выполняется одно действие, в случае невыполнения (выход «-») – другое действие.

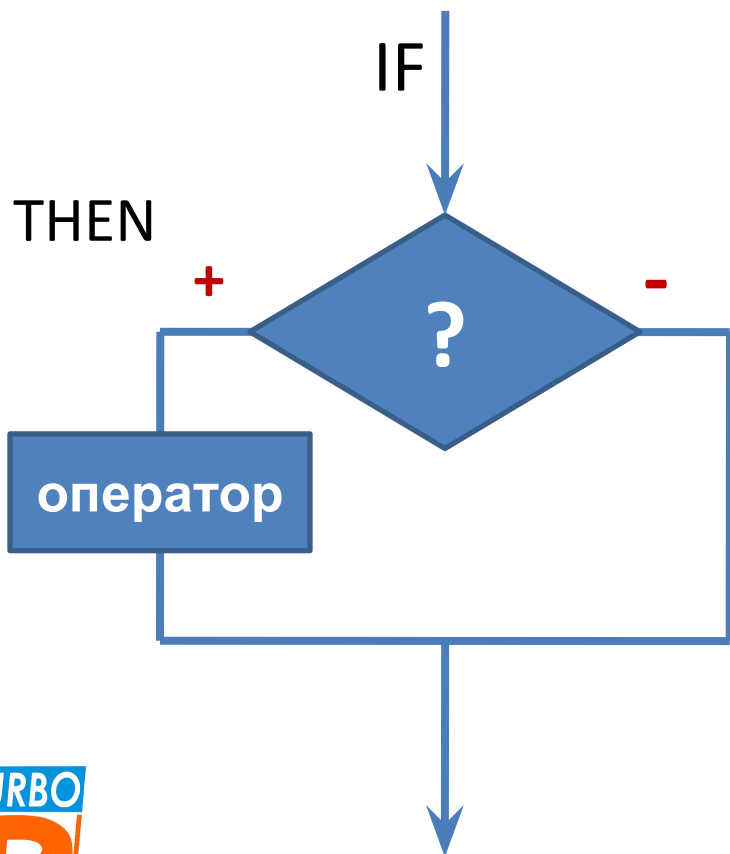
- Полная форма условного оператора



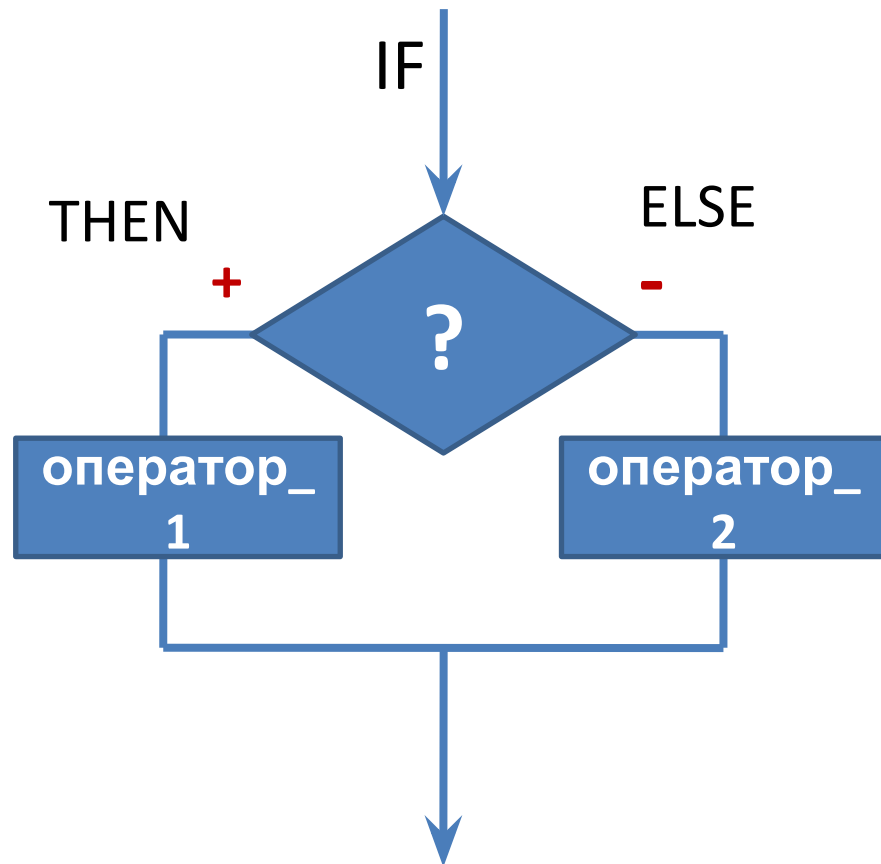


Структуры условного оператора

Неполная форма
условного оператора



Полная форма
условного оператора





Структуры условного оператора

- Каждая структура имеет один вход и один выход.
- 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 проверятся 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;



Условный оператор

- Если после слов **THEN** и **ELSE** надо выполнить не один оператор, а несколько, то эти операторы заключают в так называемые **операторные скобки**, открывающая скобка которых – слово **BEGIN**, а закрывающая - **END**:

begin
(операторы)

end;

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

В операторных скобках **BEGIN - END**

рекомендуется каждую пару записывать в одном столбце: так легче проверить соответствие каждой открывающей скобке закрывающую.



Примеры условного оператора

- **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 форме 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. Транслятор поступает следующим образом. Встретив сложную конструкцию из вложенных условных операторов, он анализирует её с конца, приписывая последнее найденное **ELSE** первому встреченному при просмотре справа налево

IF.



Пример программы

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

```
program E3;  
var a,b: integer;  
begin  
  writeln ('введите два целых числа' );  
  readln (a,b);  
  if a mod 2 = 0 then writeln ('a-четное')  
    else writeln ('b-нечетное');  
  if b mod 2 = 0 then writeln ('b-четное')  
    else writeln ('a-нечетное')  
end.
```



Логические выражения

- Алгоритм решения квадратного уравнения содержит проверку условия **$d < 0$** .
- Два значения, d и 0 , связаны отношением $<$ - меньше.
- Если условие выполняется, то говорят, что соответствующее выражение **ИСТИННО**, если не выполняется – выражение **ЛОЖНО**. Речь идет о логическом выражении.
- Для построения сложных условий в Паскале имеются логические операции **and** (и), **or**(или) и **not**(не).
- Обозначим истинное значение через 1 или ложное через 0, построим таблицы истинности для этих операций.



Таблицы истинности для логических операций

X	Y	X and Y
1	1	1
1	0	0
0	1	0
0	0	0

X	Y	X or Y
1	1	1
1	0	1
0	1	1
0	0	0

X	not X
1	0
0	1



Примеры построения сложных логических выражений.

Пример 1.

- Пусть требуется определить, принадлежит ли точка с координатой x отрезку $[a;b]$.
- Если записать это условие двойным неравенством, то читать его надо так:
 x меньше либо равен b
и больше либо равен a
 $(a \leq x \leq b)$.
- Отношение «меньше либо равно» в Паскале записывается двумя знаками.
- Аналогично записывается и «больше либо равно».
- Однако в Паскале нельзя записывать двойное неравенство. Используя логическую операцию and (и), запишем:
 $(x \geq a) \text{ and } (x \leq b)$.
- Отношения, между которыми стоит логическая операция, заключается в круглые скобки.



Примеры построения сложных логических выражений.

Пример 2.

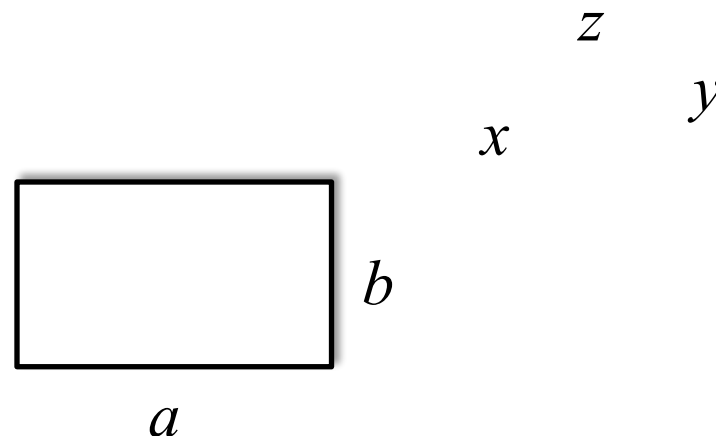
- Имеется прямоугольное отверстие со сторонами a и b и кирпич с ребрами x , y , z .

Требуется составить условие прохождения кирпича в отверстие. (рис.3)

- Кирпич пройдет в прямоугольное отверстие, если выполняется сложное условие:

$(a \geq x) \text{ and } (b \geq y) \text{ or}$
 $(a \geq y) \text{ and } (b \geq x) \text{ or}$
 $(a \geq x) \text{ and } (b \geq z) \text{ or}$
 $(a \geq z) \text{ and } (b \geq x) \text{ or}$
 $(a \geq y) \text{ and } (b \geq z) \text{ or}$
 $(a \geq z) \text{ and } (b \geq y)$

Рис.3



- Для трех граней шесть условий получается потому, что можно каждую грань повернуть на 90° и проверить для каждой грани два случая.

Примеры построения сложных логических выражений.

Пример 3.

- Определить принадлежность точки фигуре.
- Пусть фигура задана ограничивающими ее прямыми (рис.4).
- Для каждой прямой определим полуплоскость, в которой находится фигура – треугольник ABC. Полуплоскость задается неравенством.

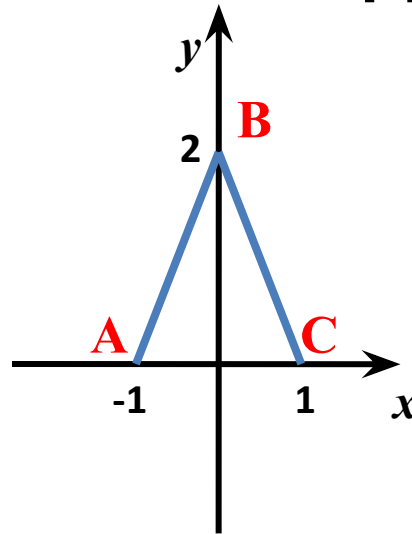
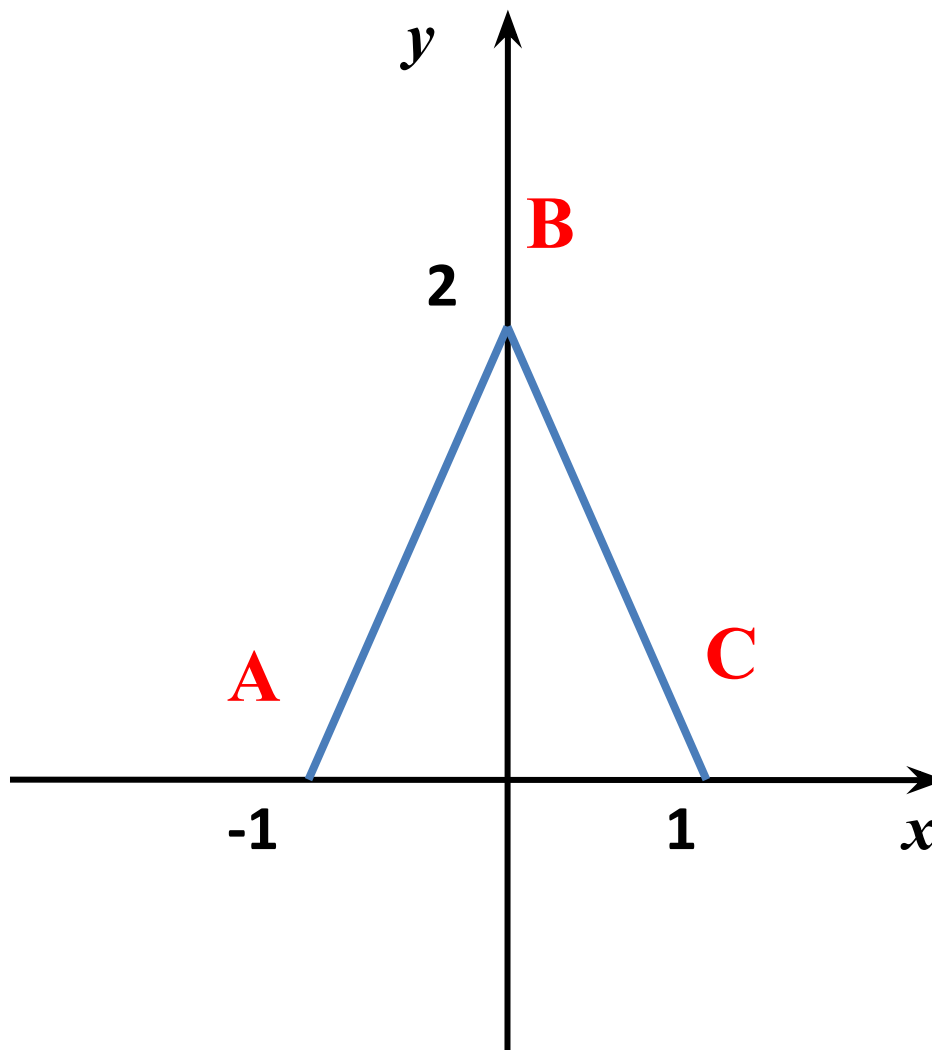


Рис.4

- Полуплоскость, находящаяся выше оси x определяется неравенством $y > 0$.
- Полуплоскость, находящаяся справа от прямой, соединяющей точки $A(-1;0)$ и $B(0;2)$ задается неравенством $y - 2x - 2 < 0$.
- Полуплоскость, находящаяся слева от прямой, соединяющей точки $C(1;0)$ и $B(0;2)$ задается неравенством $y + 2x - 2 < 0$.

Условие принадлежности точки $(x;y)$ фигуре:
 $(y > 0) \text{ and } (y - 2 * x - 2 < 0) \text{ and } (y + 2 * x - 2 < 0)$



Условие принадлежности точки $(x;y)$ фигуре:
 $(y>0)\text{and}(y-2*x-2<0)\text{and}(y+2*x-2<0)$



Примеры построения сложных логических выражений.

Пример 4.

- Приведем пример программы определения существования треугольника со сторонами a , b и c .
- Условие существования треугольника известно из геометрии: сумма двух сторон должна быть больше третьей.
- Следовательно, для всех сторон «сумма двух больше третьей» должно выполняться.

```
program E4;  
var a,b,c: real;  
begin  
  writeln ('введите длины трех  
  сторон треугольника');  
  readln (a,b,c);  
  write ('треугольник со  
  сторонами', a,',',b,',',c);  
  if (a+b>c) and (b+c>a) and  
  (a+c>b)  
  then write (' существует')  
  else write (' не существует');  
  readln  
end.
```

В Турбо
Паскале



Pascal ABC

Файл Правка Вид Программа Сервис Помощь

•E4.pas

```
program E4;
var a,b,c: real;
begin
    writeln ('введите длины трех сторон треугольника');
    readln (a,b,c);
    write ('треугольник со сторонами ', a, ', ', b, ', ', c);
    if (a+b>c) and (b+c>a) and (a+c>b)
    then write (' существует')
    else write (' не существует');
end.
```

```
введите длины трех сторон треугольника
2 3 6
треугольник со сторонами 2,3,6 не существует
```





Вопросы и задания

1. Как транслятор анализирует вложенные условные операторы?
2. Как работает неполный условный оператор?
3. Как проверить, является ли целое число нечетным?
4. Как выполняются логические операции **и**, **или**, **не**?



Напишите программы на Паскале для решения следующих задач.

5. Даны три числа a , b и c . Выясните, верно ли $a < b < c$. Ответ получите в текстовой форме: верно или неверно.
6. Даны положительные числа a , b , c , x . Выясните, пройдет ли кирпич с ребрами a , b , c в квадратное отверстие со стороной x .
7. Выясните, принадлежит ли числа a и b промежутку $(-1; 1)$.
8. Даны числа x и y . Вычислите число z , равное $x+y$, если $x \leq y$, и $1-x+y$ в противном случае.
9. Присвойте z значение большего из чисел x и y в том случае, если $x < 0$, и меньшего, если $x \geq 0$.



Напишите программы на Паскале для решения следующих задач.

10. Даны три действительных числа. Выберите те из них, которые принадлежат отрезку $[1;3]$.
11. Присвойте величине a значение наибольшего из трех заданных чисел.
12. Даны два числа. Выведите первое из них, если оно больше второго, и оба числа, если это не так.
13. Проверьте, есть ли среди трех заданных чисел равные.
14. Даны два действительных числа. Меньшее из них замените полусуммой этих чисел, а большее – их произведением.



Напишите программы на Паскале для решения следующих задач.

15. Вычислите наименьшее из трех заданных чисел.
16. Найдите решение уравнения $ax+b=0$, если оно существует.
17. Если данное число x меньше 0, то z присвойте значение большего из двух чисел x и y , иначе z присвойте значение полусуммы этих чисел.
18. Даны три действительных числа. Найдите наибольшее значение из их суммы и произведения.
19. Даны действительные числа a , b и c . Удвойте эти числа, если они являются упорядоченными по возрастанию.



Литература

- **А.А.Кузнецов, Н.В.Ипатова**
«Основы информатики», 8-9 кл.:
 - Раздел 3. ОСНОВЫ
ПРОГРАММИРОВАНИЯ,
С.93-99