

Условный оператор.

Условный оператор позволяет проверить некоторое условие и в зависимости от результатов проверки выполнить то или иное действие. Таким образом, *условный оператор* – это средство *ветвления* вычислительного процесса.

Оператор	Модуль	Назначение	Пример
IF	SYSTEM	Предназначен для <i>ветвления</i> процесса выполнения программы в зависимости от условия. Форма записи: IF <условие> THEN <оператор1> ELSE <оператор2>; IF , THEN , ELSE – <i>если, то, иначе</i> ; <условие> – произвольное выражение логического типа; <оператор1>, <оператор2> – любые исполняемые операторы.	VAR X, Y, MAX: INTEGER; BEGIN ... IF X>MAX THEN Y:=MAX ELSE Y:=X; ...

Условный оператор работает по следующему алгоритму. Вначале вычисляется условное выражение <условие>. Если результат есть **TRUE** (истина, т.е. *условие выполняется*), то выполняется <оператор1>, а <оператор2> пропускается; если результат есть **FALSE** (ложь, т.е. *условие не выполняется*), наоборот, <оператор1> пропускается, а выполняется <оператор2>.

Часть **ELSE** <оператор2> условного оператора может быть опущена, тогда получается *краткая* форма записи оператора **IF**. Краткая форма записи:

IF <условие> **THEN** <оператор>;

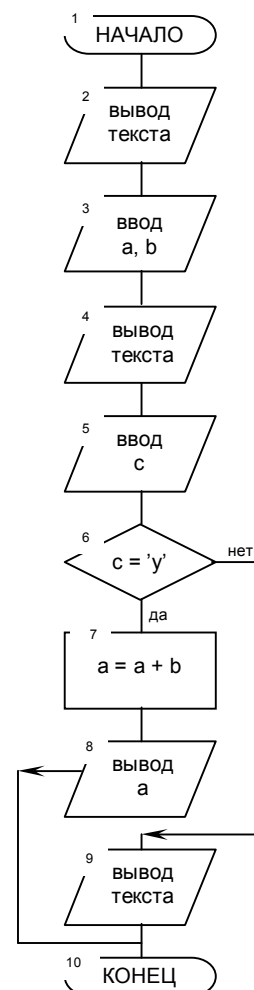
Тогда при выполнении условия выполняется <оператор>, в противном случае этот оператор пропускается.

Пример программы.

```

Program Oper_If;
Var
  a, b : Integer;
  c    : Char;
BEGIN
Write('Введите два числа: ');
Readln(a,b);
Write('Сложить их (y/n)? ');
Readln(c);
If c='y' Then
  Begin
    a:=a+b;
    Writeln('Результат = ', a);
  End
Else
  Writeln('Не очень-то и хотелось. ');
END.

```



Знаки отношений.

Математика	Turbo Pascal
<	<
>	>
=	=
≤	<=
≥	>=
≠	<>

Логические операции.

NOT – логическое **НЕ** (логическое отрицание);
AND – логическое **И** (логическое умножение);
OR – логическое **ИЛИ** (логическое сложение).

В отличие от многих других языков программирования в **Turbo Pascal** логические операции имеют более *высокий* приоритет, чем операции *отношения*. В связи с этим, в сложных логических выражениях обычно необходимо расставлять *скобки*. Выражение типа:

```
IF A = B AND C < D THEN ...
```

вызовет сообщение о *синтаксической ошибке*, так как сначала выполнится операция **B AND C**. Правильным будет выражение:

```
IF (A = B) AND (C < D) THEN ...
```

Условие – выражение логического типа (отношение одного значения или выражения к другому), которое можно заменить на логическую переменную (переменная типа **Boolean**). Приведённые ниже записи равнозначны между собой:

<pre>Var a, b : Integer; ... If a > b Then Writeln(a);</pre>	<pre>Var a, b : Integer; u : Boolean; ... u:=a > b; If u Then Writeln(a);</pre>
---	--

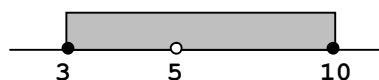
Таблица использования логических операций.

A	B	A and B	A or B	not
0	0	0 and 0 = 0	0 or 0 = 0	not 1 = 0
0	1	0 and 1 = 0	0 or 1 = 1	not 0 = 1
1	0	1 and 0 = 0	1 or 0 = 1	
1	1	1 and 1 = 1	1 or 1 = 1	

(0 – False; 1 – True)

Пример применения логических операций.

На числовой оси выделен диапазон значений (от 3 до 10, за исключением 5). Составить условие для этого диапазона.



Математическая форма условия: $[3,5) \cup (5,10]$ или $3 \leq x \leq 10, x \neq 5$

Запись условий для **If**: $((3 \leq x) \text{ and } (x < 5)) \text{ or } ((5 < x) \text{ and } (x \leq 10))$ или $(3 \leq x) \text{ and } (x \leq 10) \text{ and } (x <> 5)$

Составной оператор.

Составной оператор – это группа исполнительных операторов, заключённых между **BEGIN** и **END**;, которые воспринимаются компилятором как один «*большой*» оператор (см. пример программы).