

Алгоритмическая структура ветвление (разветвление)

**Оператор разветвляющейся
структуры If...Then**

- Алгоритмическая структура ветвление реализуется в VBA оператором **If...Then...Else** (ЕСЛИ...ТО...ИНАЧЕ). Этот управляющий оператор осуществляет ветвление программы в зависимости от выполнения условия, задаваемого логическим выражением. Логическое выражение является *истинным*, если условие выполняется, и *ложным*, если не выполняется. Оператор **If...Then...Else** (или просто оператор **If**) имеет две формы записи (синтаксис) – линейную и блочную.

Линейный синтаксис оператора If ... Then

- При линейном синтаксисе весь оператор записывается *в одну строчку* (перенос на новую строку не допускается)

Полная форма записи



Здесь обозначены:

– логическое_выражение – любое логическое выражение, допустимое на языке Бейсик. Логическое выражение может возвращать либо ненулевое значение (True – «истина») – при выполнении условия, либо ноль (False – «ложь») – если условие не выполняется.

Если логическое выражение состоит из цепочки нескольких логических выражений, то они соединяются друг с другом посредством логических операций – And («И») или Or («ИЛИ»);

– операторы_1 – один или несколько операторов, которые будут выполняться при выполнении условия, заданного логическим выражением («истина»). Если операторов несколько, то они записываются в одну строку и отделяется между собой двоеточием;

– операторы_2 – операторы, выполняющиеся при значении логического выражения «ложь»

Краткая форма записи оператора **If** не содержит часть **Else**

ИСТИНА

If логическое_выражение **Then** операторы_1

ЛОЖЬ

(следующая строка)

алгоритмическая структура «ветвление».

VISUAL BASIC	Блок-схема
<p>1. Полная форма ветвления IF <условие> THEN <действие1> ELSE <действие2> END IF</p> <p>2. Неполная форма ветвления IF <условие> THEN <действие1> END IF</p>	<pre>graph TD subgraph Full_Form [1. Полная форма ветвления] C1{условие} A1[действие1] A2[действие2] C1 -- нет --> A1 C1 -- да --> A2 A1 --> J1(()) A2 --> J1 J1 --> O1[] end subgraph Incomplete_Form [2. Неполная форма ветвления] C2{условие} A3[действие1] C2 -- да --> A3 C2 -- нет --> J2(()) A3 --> J2 J2 --> O2[] end</pre>

Например:

a = 0: b = 1

IF a < b THEN t = 15: v = 16 ELSE t = 17: v = 17

Результат: t = 15, v = 16

Вычислить функцию $y(x)$:

$$y = \begin{cases} |x|, & \text{если } x < 0 \\ x^2, & \text{если } x \geq 0 \end{cases}$$

X=-2

If x < 0 Then y = Abs(x) Else y = x ^ 2

Результат: $y=2...$

Или

X=2

If x < 0 Then y = Abs(x)

If x > = 0 Then y = x ^ 2

Результат: $y=4$

Блочный синтаксис оператора If...Then

Блочная форма применяется в случае, когда после ветвления алгоритма необходимо выполнить несколько операторов, или если в алгоритме программы требуется выполнить более сложный выбор (2,3... условия).

При этом исполняемые операторы записываются в несколько строк – блок. Блок должен заканчиваться ключевым словом **End If** (конец блока If).

Минимальная блочная форма

If логическое_выражение **Then**

Оператор_1

Оператор_2

.....

Else

Операторы_N

End If

Максимальная блочная форма

Применяется, когда необходимо
анализировать не одно или два условия, а
целый ряд взаимосвязанных условий

```
If логическое_выражение_1 Then  
    [блок_операторов_1]  
ElseIf логическое_выражение_2 Then  
    [блок_операторов_2]  
.....  
ElseIf логическое_выражение_N -1 Then  
    [блок_операторов_N-1]  
Else  
    [блок_операторов_N]  
End If
```

(**ElseIf** пишется слитно, а в слове **End If** ставится пробел)

При полной блочной форме проверяется сначала первое логическое выражение. Если оно истинное, выполняются операторы блока, следующего за ключевым словом **Then** (блок_операторов_1). Если логическое выражение ложно, происходит переход к оценке условия, заданного в последующей строке **ElseIf**. И т.д.

Если ни одно из условий **ElseIf** не выполняется, то выполняются операторы блока **Else** (блок_операторов_N).

Например:

$$y = \begin{cases} |x|, & \text{если } x < 0 \\ x^2, & \text{если } 0 \leq x \leq 5 \\ x + 2, & \text{если } x > 5 \end{cases}$$

.....

x=2

If x < 0 Then

y = Abs(x)

Elseif x >= 0 And x <= 5 Then

y = x ^ 2

Else

y = x + 2

End If

.....

Результат: y=4

Операции отношения и логические операции

Логическое выражение может быть истинным или ложным. В отличие от арифметических выражений логические выражения принимают не числовые значения, а логические значения – "True" (истина) или "False" (ложь).

Операции отношения

Операция сравнения	Описание
$A = B$	Равно. Результат True, если $A = B$
$A <> B$	True, если A не равно B
$A < B$	True, если A меньше B
$A > B$	True, если A больше B
$A \leq B$	True, если A меньше или равно B
$A \geq B$	True, если A больше или равно B

Например:

выражение $25 <> 30 \rightarrow True$;

$25 < 30 \rightarrow True$;

$25 > 30 \rightarrow False$.

В логическом выражении допустимы два или несколько условий (операций отношения), которые связываются *логическими операциями And, Or, Not* (И, ИЛИ, НЕ) . Логические операции возвращают значение "Истина" (не нулевое) и "Ложь" (нулевое) значения.

Логические операции

Логическая операция	Действие
$A \text{ And } B$	Лог. выражение истинно если оба условия принимают значение True, то и результат тоже True
$A \text{ Or } B$	Если хотя бы одно из значений A или B будет True, то результат тоже True
$\text{Not } A$	Если A True, то результат False

Например

:если $a=2$: $b=2$: $c=3$, то

$a=b$ *And* $a \neq c \rightarrow \text{True}$

$a < b$ *And* $a \neq c \rightarrow \text{False}$

$a=b$ *Or* $a > c \rightarrow \text{True}$

$a < b$ *Or* $a > c \rightarrow \text{False}$

***Not* ($a=b$) $\rightarrow \text{False}$.**

(Операции отношения имеют бóльший приоритет, чем логические операции)

