

Урок 6. Условный оператор.

Условный оператор позволяет организовать в программе *ветвления*. Определённая команда (набор команд) выполняется в зависимости от истинности некоторого логического выражения. Условный оператор имеет *полную* и *краткую* формы.

1. Полная форма условного оператора

<code>if условие then</code>	<code>if x > 5 then</code>
<code> оператор1</code>	<code> x := 0</code>
<code>else</code>	<code>else</code>
<code> оператор2;</code>	<code> x := x + 1;</code>

if (англ.) – "если"
then (англ.) – "тогда"
else (англ.) – "иначе"

В качестве *условия* указывается некоторое логическое выражение. Если *условие* оказывается истинным, то выполняется *оператор1*, в противном случае выполняется *оператор2*.

Перед `else` точка с запятой не ставится!

2. Краткая форма условного оператора

<code>if условие then</code>	<code>if y <> 0 then</code>
<code> оператор;</code>	<code> Sleep(1000);</code>

Если *условие* оказывается истинным, то выполняется *оператор*, в противном случае происходит переход к следующему оператору программы.

3. Условный оператор с несколькими условиями

<code>if условие1 then</code>	<code>if x < 100 then</code>
<code> оператор1</code>	<code> z := 1</code>
<code>else if условие2 then</code>	<code>else if x < 400 then</code>
<code> оператор2</code>	<code> z := 2</code>
<code>else if условие3 then</code>	<code>else if x < 1000 then</code>
<code> оператор3</code>	<code> z := 3</code>
<code>else</code>	<code>else</code>
<code> оператор4;</code>	<code> z := 0;</code>

Если *условие1* истинно, то выполняется *оператор1*, в противном случае если *условие2* истинно, то выполняется *оператор2*, в противном случае если *условие3* истинно, то выполняется *оператор3*. Если все условия оказываются ложными, то выполняется *оператор4*.

Данная конструкция языка состоит из полных форм условного оператора, "сцепленных" друг с другом.

4. Составной оператор (begin end)

Составной оператор предназначен для объединения нескольких операторов в один.

```
if условие then
begin
    оператор1;
    оператор2;
    оператор3
end;
```

```
if x = y then
begin
    SetPenColor(clGreen);
    SetBrushColor(clRed);
    Circle(x, y, 50)
end;
```

begin (англ.) – "начинать"

end (англ.) – "конец", "заканчивать"

Ключевые слова *begin* и *end*, окаймляющие операторы, называются *операторными скобками*.

5. Вложенные условные операторы

```
if условие1 then
    if условие2 then
        оператор1
    else
        оператор2;
```

else всегда относится к ближайшему предыдущему оператору *if*, для которого ветка *else* еще не указана. Если в предыдущем примере требуется, чтобы *else* относилась к первому оператору *if*, то необходимо использовать составной оператор:

```
if условие1 then
begin
    if условие2 then
        оператор1
    end
else
    оператор2;
```