

Программирование на MQL 4: Условные операторы

Александр Иванов (АКА HORN)

<http://www.fxtrade.tomsk.ru>

Сегодня, как и обещали прошлым номере журнала Forex Magazine, мы рассмотрим ещё одно недостающее звено для создания полноценных программ: "условные операторы".

Если попытаться провести параллели между ситуациями, возникающими в программировании, и повседневной жизнью, то "условные операторы" можно было бы объяснить на примере путника, оказавшегося на перепутье и размышляющего, куда пойти дальше. Попробуем представить себе ход рассуждений, подталкивающих путника к выбору того или иного направления пути. Если опустить все детали рассуждений нашего "путника", то ход его мыслей можно представить фразой: "если выполняется некое условие, то сделать такое-то действие, иначе сделать другое действие". На данный момент нам не так важно, на основании чего он может решить, в каком направлении продолжить дальнейший путь, так как сейчас для нас самое главное то, что в подобной ситуации возникает момент выбора. Вкратце, такую схему поведения будем называть "если, то, иначе".

Рано или поздно в алгоритме почти любой программы появляется ветвление дальнейшего хода действий, это как раз и есть тот момент, когда требуется определить, в каком из направлений будет дальше двигаться программа. Для иллюстрации сказанного, приводим схему кода, написанную на MQL 4:

```
if(condition) {  
    // выполнить этот блок,  
    // если condition равен true.  
} else {  
    // выполнить этот блок,  
    // если condition равен false.  
}
```

Переменная "condition" в данном случае должна быть типа "bool". Она может принимать значения true или false. Кроме того, вместо переменной "condition" может быть подставлено какое-либо логическое выражение, которые составляют с помощью операторов сравнения (>, >=, ==, <=, <), оператора отрицания (!) и операторов and и or.

В том случае, когда переменная "condition" или подставленное вместо неё логическое выражение принимают значение "true", будет выполнен код, заключённый в первые операторные скобки. Если же переменная "condition" или подставленное вместо неё логическое выражение принимают значение "false", выполняется код, заключённый во вторые операторные скобки, то, что после слова "else".

Далее переменную "condition" или подставленное вместо неё логическое выражение будем называть условием оператора "if".

Приведённая в предыдущем примере конструкция называется полной формой условного оператора "if() {} else {}". Однако, не всегда необходимо выполнять какой-либо код, если условие оператора "if" ложно. Следующий пример показывает использование краткой формы оператора "if() {}":

```
if(condition) {  
    // выполнить этот блок,  
    // если condition равен true.  
}
```

Краткой формой оператора "if" удобно пользоваться, когда происходит проверка начальных параметров каких-либо функций.

Например, на приведённом ниже куске кода показаны начальные строки функции `init()` индикатора `Moving Averages`, в котором происходит проверка входных параметров для построения индикатора. В случае, когда расчетный период для этого индикатора меньше, чем один бар, или указанный при инициализации тип не совпадает ни с одним из четырёх позволенных типов МА, индикатор ещё на этапе инициализации прекращает свою работу.

```
if(MAPeriod<1 ||
    MAType<0 || MAType>3) {

    g_Failed=true;
    return(0);
}
```

Если же все указанные проверки пройдены, то не нужно дальнейшие действия оформлять в виде "else" части оператора, инициализация и в том и в другом случае будет продолжена.

Вообще, если мы стремимся к написанию быстрых программ, то полной формой оператора "if" следует пользоваться как можно реже. Это связано с особенностями выполнения программ на уровне процессора.

Для полноты изучения материала следует обратить наше внимание на то, что в условии оператора "if" можно использовать не только логические выражения. Если в качестве условия указать числовую переменную или выражение, результатом работы которого будет число, то оператор "if" будет воспринимать все числа отличные от нуля как значение "true", иначе, если ноль, то условие будет считаться ложным. Это можно использовать в тех случаях, когда значение некоторой переменной проверяется на равенство (неравенство) нулю.

На приведённом ниже примере показано два варианта проверки переменной на равенство нулю. Здесь оба оператора "if" выполняют одно и то же:

```
int i = 1234;

// первый вариант
if(i != 0) {
    Alert("i != 0");
}

// второй вариант
if(i) {
    Alert("i != 0");
}
```

На этом мы пока закончим рассматривать способы и примеры использования условного оператора "if", потому что его использование является неотъемлемой частью любой более или менее развитой программы, и в дальнейшем мы ещё неоднократно с ним столкнёмся.

На сегодня всё. Хотелось бы обратить внимание читателей, желающих обсудить статьи, задать вопросы по программированию на MQL или просто поговорить на около-финансовые темы, что с этой недели начинает работу форум на сайте www.fxtrade.tomsk.ru, добро пожаловать!

*Александр Иванов
для Forex Magazine*