

Программирование на MQL 4: Переменные

Александр Иванов (AKA HORN)
<http://forex.tomsk.ru>

В предыдущем номере журнала Forex Magazine мы начали рассказ о программировании на языке MQL 4, встроенном в новую торгово-информационную систему Meta Trader 4. Напомним, что мы уже обсудили то, какие бывают типы пользовательских программ; также мы коснулись того, как организована программа вообще, и проиллюстрировали статью реальным кодом.

Продолжим изучение MQL 4 с того, что рассмотрим более детально "кирпичики", из которых складывается программа.

Представьте, что нам потребовалось сложить два числа и на некоторое время сохранить результат вычисления. Как бы мы поступили в таком случае в обычной жизни? Наверное, проще всего было бы записать результаты на бумажку и обратиться к ним позже, когда потребуется. Но предположим, что таких вычислений мы делаем множество и все результаты записываем на бумажку. Как потом, допустим, через неделю, мы сможем вспомнить, что означает каждое из записанных чисел? Можно перед каждым из записанных чисел написать словами, что же оно для нас означает - в таком случае, и через неделю, и через более длительный срок мы сможем восстановить смысл записанных чисел.

Но вернёмся к программированию. Первым из "кирпичиков", с помощью которых создаются программы, являются так называемые "переменные". В процессе написания программ, так или иначе, приходится оперировать какими-то данными, из-за чего появляется потребность сохранять промежуточные результаты в памяти компьютера. Так же как и в жизни, для того, чтобы не путаться с тем, что где сохранено в памяти компьютера, мы будем давать имена участкам памяти, где будут храниться данные программы. Переменные - это именованные участки памяти компьютера, куда мы можем "складывать" некоторые числа. Упоминание "участков памяти компьютера" не должно смущать начинающих программистов - это было сделано, чтобы дать наиболее наглядное объяснение понятию "переменная". Реально же для создания какой-либо переменной достаточ-

но описать её тип, дать ей имя и, при необходимости, задать ей первоначальное значение, например:

```
string g_sNameOfChart = "EURUSD 4 hour";
```

В данном примере создаётся переменная `g_sNameOfChart` для хранения строк типа `string` и ей присвоено начальное значение "EURUSD 4 hour". Пока не обращайтесь пристального внимания на название этой переменной, отметьте лишь то, что в него входит понятное описание предназначения этой переменной "Name Of Chart". Сразу сделаем одно замечание, дабы предвосхитить возможные заблуждения и вопросы читателей: переменная может менять своё значение по желанию программиста. То есть, то значение, которое однажды было присвоено переменной, может быть изменено программистом в любой момент, поэтому то она и носит название "переменная". Сделать это можно, например, вот так:

```
g_sNameOfChart = "GBPJPY 1 week";
```

Теперь настало время поговорить о типах переменных. Выше приведённый пример содержит объявление переменной строкового типа. Кроме типа `string` существуют ещё `int`, `bool`, `double`, `datetime` и `color`. И так обо всех по порядку:

Тип переменных `int` - это сокращение от английского слова "integer" - целое число. Переменные этого типа, как следует из его названия, могут содержать только целые числа.

Тип переменных `bool` - это сокращение от слова "boolean" - булевский тип, названный по имени английского математика и логика Джорджа Буля (George Boole) разработавшего дискретную алгебру логики. Переменные этого типа могут принимать значения только "true" - истина и "false" - ложь.

Тип переменных `double` - переменные этого типа могут хранить рациональные числа с плавающей точкой двойной точности (`double float`).

Тип переменных `datetime` - переменные этого типа, могут хранить дату и время. Дата и время хранятся в них в

количестве секунд, прошедших с 0 часов 0 минут 0 секунд 1-го января 1970-го года.

Тип переменных color - переменные этого типа служат для работы с цветами.

Тип переменных string - переменные этого типа могут хранить строки.

Ниже приведён пример объявления переменных каждого из рассмотренных типов данных языка MQL 4:

```
int      SomeNumber   = 123456;
bool     IsRightPrice = true;
double   CurrentPrice = 1.2317;
datetime CurrentDate  = D'2004.04.07 8:00';
color    ChartColor   = Olive;
string   MyName        = "Alexander";
```

Переменные можно называть как угодно, главное, чтобы, встретив переменную в программе, вы смогли легко понять, что за значения могут содержаться в ней, и что это обозначает. Мы рекомендуем использовать описательные имена переменных, типа тех, что приведены в примере. Но приведённый выше пример содержит объявления переменных говорящих только о том, что обозначают содержащиеся в переменных значения. Для более детального описания переменной, можно следовать ещё одной нашей рекомендации и использовать суффиксы, описывающие не только значение переменной, но и тип этой переменной. Такой способ описания переменных называется "венгерской нотацией". Её происхождение приписывается одному из программистов фирмы Microsoft, который по национальности был венгром.

Смысл венгерской нотации в применении к типам MQL 4 заключается в том, что перед именами переменных типа string нужно ставить суффикс "s", перед именами переменных типа int - суффикс "n", перед bool - суффикс "b", перед double - суффикс "f", перед datetime - суффикс "t", а перед именами переменных типа color - суффикс "c".

Ниже приведён пример объявления тех же самых переменных, но уже с применением венгерской нотации:

```
int      nSomeNumber   = 123456;
bool     bIsRightPrice = true;
double   fCurrentPrice = 1.2317;
datetime tCurrentDate  = D'2004.04.07 8:00';
color    cChartColor   = Olive;
string   sMyName        = "Alexander";
```

Это простое изменение в названии переменной существенно облегчает написание больших программ и, как следствие, уменьшает вероятность появления ошибок.

Существует ещё один тип - void, но он не должен использоваться для объявления переменных, и о нём мы поговорим позже.

Читатель может спросить: "зачем существует так много различных типов?". Дело в том, что, во-первых, каждый из этих типов по-разному хранится в памяти компьютера, а во-вторых, операции над различными типами данных выполняются процессором компьютера с разными скоростями. Например, предыдущая версия языка MQL предписывала хранить все числа в переменных типа double, что обуславливало использование медленных процессорных операций даже в тех случаях, когда это было не обязательно. Теперь же программист в явном виде может указать переменные, которые будут хранить целочисленные данные и, соответственно, будут в десятки раз быстрее обрабатываться процессором.

Числовые типы можно смешивать, выполняя арифметические операции, однако следует помнить, что когда мы будем в целочисленную переменную записывать значение типа double, то дробная часть исчезнет, и будет учтена только целая часть рационального числа.

Тип string не удастся присвоить переменной какого-либо другого типа, а вот наоборот можно. Когда мы в переменную типа string попытаемся записать числовое значение, то оно автоматически преобразуется к строке.

Если попытаться использовать числовые переменные в том месте, где подразумевается использование булевых переменных, то следует помнить, что все значения, отличные от нуля, будут восприниматься как "true", и только нулевые значения - как "false".

Вторым из "кирпичиков", составляющих программы, являются функции. Функции - это одно из новшеств, отсутствовавших в MQL II и появившихся в MQL 4. Их мы подробно обсудим в следующем номере журнала Forex Magazine.

*Александр Иванов
для Forex Magazine*