FOREX MAGAZINE

Программирование на MQL 4: Переменные. Часть 2 Июль 2004 №27

Программирование на MQL 4: Переменные. Часть 2

Александр Иванов (AKA HORN)
http://fxtrade.tomsk.ru

В этой статье мы продолжим разговор о переменных, объясним их предназначение и некоторые нюансы, связанные с их использованием.

Pre-defined variables

В MQL 4 встроены предопределённые переменные (pre-defined variables), такие как Ask, Bid, Bars, Close, Open, High, Low, Time и Volume.

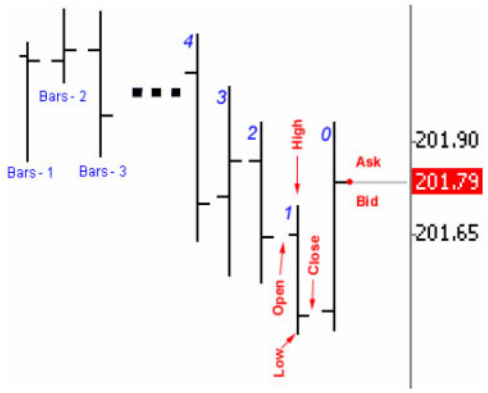
Эти переменные "привязаны" к конкретному графику; любой из открываемых вами инструментов на каждом из своих временных интервалов имеет собственный набор упомянутых выше предопределённых переменных. Доступ к значениям этих переменных из пользовательской программы, прикреплённой к графику некоторого временного периода, можно осуществить в любой момент времени. Изменять значения этих переменных нет возможности.

Следует, отметить, что переменные Close, Open, High, Low, Time и Volume представляют не единичные значения, а массивы, содержащие в себе исторические данные для рассматриваемого инструмента. Доступ к последнему значению (мы называем такое значение "самым молодым") любой из этих переменных осуществляется по имени переменной с указанием нулевого индекса. Например, самая последняя цена продажи торгуемого инструмента будет равняться Ask[0].

Переменные Bid и Ask актуальны только для формируемого в текущий момент бара, поэтому они не являются массивами. В них хранится информация о предлагаемой продавцом цене и запрашиваемой цене на покупку.

Ценовые графики принято рассматривать в масштабах некоторых временных периодов - баров.

Бар - это графическое изображение одного ценового периода. Бар имеет свою цену открытия (переменная Open) - цену, которая, была на рынке на момент начала временного периода, изображаемого баром; максимальную (переменная High) и минимальную (переменная Low) цену, достигнутую на промежутке времени бара и цену закрытия бара (переменная Close) - цену торгуемого инструмента на рынке на конец временного периода, изображаемого баром. Приведённый ниже рисунок объясняет, чему на графике соответствуют эти переменные.



Как мы уже упомянули, о номерах баров удобно думать, как об их возрасте. Текущий бар ещё не сформировался до конца, и поэтому его возраст равен нулю. Бар, который сформировался на предыдущем интервале времени, имеет возраст один период, и так далее. На приведённом рисунке номера баров указаны синими цифрами, и видно, как справа налево с увеличением возраста баров увеличиваются их номера.

Для того, чтобы узнать, сколько всего баров доступно нам в программе, мы можем пользоваться предопределённой переменной Bars. Для наглядности на рисунке пропущены средние бары (вместо них вы видите три жирные точки) и оставлены несколько последних баров. Как вы можете видеть, первый слева бар (самый старший по возрасту) имеет номер равный Bars-1, второй слева бар имеет номер равный Bars-2 и так далее.

Многих ставит в тупик тот факт, что самый старший бар имеет номер равный Bars-1. Это происходит из-за того, что нумерация баров осуществляется не с единицы, а с нуля. Допустим, имея десять баров, пронумерованных с нуля, мы получим следующий ряд из их индексов:

9, 8, 7, 6, 5, 4, 3, 2, 1, 0

В итоге, переменная Bars будет содержать количество баров (десять), а самый старший бар будет иметь индекс равный Bars-1 (девять).

Массив значений Time содержит времена открытия периодов изображённых баров.

Массив значений Volume содержит тиковые объёмы баров. По ним можно судить только о количестве сделок, а не о реальном объёме проданного или купленного товара, будь то валюта, акции или какой-либо другой торгуемый инструмент.

Global Variables

Глобальные переменные (global variables) - это удобный инструмент для организации обмена информацией между программами или для сохранения числовых значений вашей программы на время выключения MetaTrader'a или компьютера.

Для создания или изменения значения глобальной переменной, нужно вызвать функцию GlobalVariableSet(). Эта функция сама создаст глобальную переменную, если вдруг её ещё не существует, а в том случае, когда переменная уже существует, изменит её значение. Первым параметром этой функции является имя глобальной переменной, вторым параметром - числовое значение.

Чтобы не гадать, была ли уже создана глобальная переменная с некоторым именем, можно использовать функцию GlobalVariableCheck(). Обычно функции GlobalVariableSet() и GlobalVariableCheck() используют вместе для первичной инициализации некоторой глобальной переменной следующим образом:

```
// если не существует глобальной
// переменной с именем "g1", то
// создать её и присвоить ей
// значение 12345

if(!GlobalVariableCheck("g1")) {
   GlobalVariableSet("g1",12345);
}
```

Получить значение глобальной переменной можно с помощью функции GlobalVariableGet().

Удалить глобальную переменную можно с помощью функции GlobalVariableDel(). Если же появится необходимость удалить сразу все глобальные переменные, присутствующие на текущий момент в системе, то сделать это можно, вызвав GlobalVariableDeleteAll().

Область видимости

Уделим немного внимания тем переменным, которые используются для хранения значений при написании программ. В статье "Программирование на MQL 4: Переменные", напечатанной в 21 номере журнала Forex Magazine, мы давали краткие рекомендации по применению правил именования, и сейчас мы их немного расширим. Наверное, читатели уже обратили внимание на то, что во многих программах некоторые переменные имеют префикс "g_", а некоторые не имеют этой приставки. Постараемся объяснить, с чем это связано.

При написании программ обязательно нужно учитывать так называемую "область

видимости" переменной. Дело в том, что на уровне исходного текста программы тоже есть свои глобальные переменные и локальные переменные. Локальные переменные это те, которые объявлены внутри какой-нибудь функции. Областью видимости таких переменных является тело функции, заключённое в фигурные скобки, внутри которой они объявлены. Глобальные же переменные - это те, которые объявлены вне тела какой-либо функции, они доступны всем функциям файла, в котором сохранена программа.

Для того, чтобы облегчить понимание исходного текста и процесс отладки программы, имена глобальных переменных, то есть переменных, которые имеют областью видимости весь файл, начинают с префикса "g_". Это не обязательное требование, но поверьте, что следование этой рекомендации существенно облегчит вам процесс написания программы и сбережёт нервы при отладке. Пример ниже показывает объявление глобальных и локальных переменных:

```
// глобальная переменная
int  g_nLots;

void Function1(int param1)
{
   // локальная переменная
   int nLots;

   // ...
   // выполнять какие-то действия
   // ...
};
```

Кстати, параметры, передаваемые в функцию (в данном случае это param1), имеют статус локальных переменных и область их видимости - тело функции.

Если все-таки получится так, что глобальная и локальная переменные будут названы одинаково, то пока локальная переменная будет в области своей видимости, использоваться будет она. После выхода из области видимости локальной переменной, программа будет использовать глобальную переменную.

```
// объявляем глобальную
// переменную nLots
int  nLots = 0;

void Function1()
{
   // объявляем локальную
   // переменную nLots
   int nLots;
   // ...
   // обращение к локальной
   // переменной nLots
   nLots = 100;
};

void Function2()
{
   // обращение к глобальной
   // переменной nLots
   nLots = 3;
}
```

Очевидно, что давать одинаковые имена глобальным и локальным переменным не лучшая практика, поэтому рекомендуем вам придерживаться данных в этой статье дополнений к правилам именования переменных.

Александр Иванов
для Forex Magazine