

# MQL4 COURSE

By Coders' guru  
[www.forex-tsd.com](http://www.forex-tsd.com)

-8-

## Proměnné

Vítejte v lekcí „proměnné“ mého MQL 4 kurzu.

Doufám, že se vám předchozí lekce líbily a že jste připraveni na boj s proměnnými:  
Doporučuji, abyste si před touto lekcí přečetli lekcí “DATOVÉ TYPY”.  
Stáhnout si ji můžete zde:

<http://forex-tsd.com/attachment.php?attachmentid=469>

Nyní přejdeme k vlastní lekcí.

Co znamená pojem proměnné?

Jak jsem vám již odhalil dříve, proměnné jsou jména, která odkazují na sekce v paměti, do které mají být data ukládána.

Abyste si to dokázali lépe vybavit, zkuste si představit, že paměť je série různě velkých přihrádek. Velikost přihrádky odpovídá rozsahu požadované paměti v bajtech.

- Abyste mohli do přihrádky uložit data, musíte jí přiřadit jméno; tento proces je známý jako deklarace.
- V procesu deklarování informujete počítač o typu a rozměru přihrádky, kterou si přejete použít, a to slovem, které se nazývá keyword (klíčové slovo).
- Pomůže vám, pokud přihrádce přiřadíte smysluplné pojmenování, které vám později usnadní vyhledávání dat, jméno se nazývá konstanta proměnné.
- Data jsou umístěna do přihrádky prostřednictvím přiřazení.
- Pokud použijeme hodnotu boxu, který jsme vytvořili, ve stejném řádku, ve kterém jsme deklarovali proměnnou.

Když vytvoříme proměnnou, sdělujeme tím počítači, že po něm chceme, aby specifikoval délku paměti (v bajtech) naší proměnné. Uložení jednoduchého čísla, velkého počtu čísel nebo písmene,

nezabere stejné množství paměti. Počítač se nás tedy dotáže, o jaký typ dat se jedná a jaká bude jejich délka. Za tímto účelem se používá datový typ.

Počítači sdělíme například řádek s tímto kódem:

```
int MyVariable=0;
```

To znamená, že po počítači žádáme sestavení bloku o délce 4 bajty pro naši proměnnou, pojmenovanou “MyVariable”.

V přechozím případě jsme použili:

int: Klíčové slovo

int: Datový typ celého čísla.

int: Deklarace

MyVariable: Konstanta proměnné.

=0: Inicializace

Více se o proměnných dozvíme v další lekci.

V MQL4 se vyskytují tyto druhy Datových typů:

- Integer(int) – celé číslo
- Boolean ( bool)
- Character(char) - znak
- String (string) - řetězec
- Floatin g-point number(double) – číslo plovoucího bodu (desetinné)
- Color (color) - barva
- Datetime (datetime) – datum a čas

Předchozích několik řádků jsem pro vás zkopíroval z lekce DATOVÉ TYPY, abyste věděli, jak proměnnou vytvořit, nyní se dozvíme, jak proměnnou deklarovat:

**Deklarace :**

Deklarování proměnné znamená, že ji přivedete na svět a určíte její typ. To učiníte zvolením klíčového slova, které jste se naučili v lekci DATOVÉ TYPY (int, double, char, bool, string, color a datetime) spolu se jménem, které jste proměnné přidělili.

Příklad: Zde jste deklarovali proměnnou pod názvem MyVariable, která je typem integer. Před deklarováním jméno MyVariable v kódu nemůžete použít. Pokud jste jej použili bez deklarování v MQL4, kompilátor si bude stěžovat a vyobrazí takovou zprávu: 'MyVariable'- variable not defined. 1 error(s),0 warning(s).

```
int MyVariable;
```

### Inicializace:

Inicializace proměnné znamená přiřadit k ní hodnotu. Např. `MyVariable = 0` ; Inicializovat proměnnou můžete ve stejném řádku, v jakém ji deklaruujete. Např.:

```
int MyVariable= 0 ;
```

A můžete ji deklarovat na jednom místě a inicializovat na jiném:

```
int MyVariable;  
...  
...  
MyVariable=5;
```

Mějte však na paměti tento fakt: Deklarace musí být provedena před inicializací.

### Zaměření proměnných:

Existují dva druhy zaměření proměnných: Lokální a Globální.

Zaměření znamená, která část kódu bude o proměnné vědět a používat ji.

Lokální proměnná znamená, že vnější svět nebude vidět, kde byla deklarována. Např. proměnné deklarované uvnitř funkce jsou lokální pro funkci kódu bloku, a proměnné deklarované uvnitř smyčky nebo bloku kódu funkce „rozhodnutí“, jsou lokální pro tyto bloky a nemohou být viditelné a používané mimo ně.

Příklad:

```
double my_func (double a, double b, double c)  
{  
    int d ;  
    return (a*b + c);  
}
```

V příkladě jsou proměnné a,b,c a d lokálními proměnnými, které mohou být použity pouze uvnitř funkce bloku kódu (cokoliv mimo závorek) a nemohou být použity vně. Nemůžeme tedy zapsat např. řádek: `d = 10` ; jelikož není vnímán následujícím řádkem funkce, protože je umístěn vně.

Druhým typem zaměření jsou Globální proměnné, které byly deklarovány vně jakéhokoliv bloku kódu, a mohou být viditelné v mnoha částech vašeho kódu.

Příklad:

```

int Global_Variable;
double my_func (double a, double b, double c)
{
    return (a*b + c + Global_Variable);
}

```

Zde je proměnná `Global_Variable` deklarována vně funkce (úroveň deklarace funkce), takže může být vnímána všemi funkcemi vašeho programu. Pokud nejsou globální proměnné inicializovány, budou automaticky nastaveny na nulu.

Externí proměnné:

Klíčové slovo “extern” se používá k deklarování zvláštního druhu proměnných; tyto proměnné se používají k určení vstupních dat funkcí Expert advisors nebo Custom indicators.

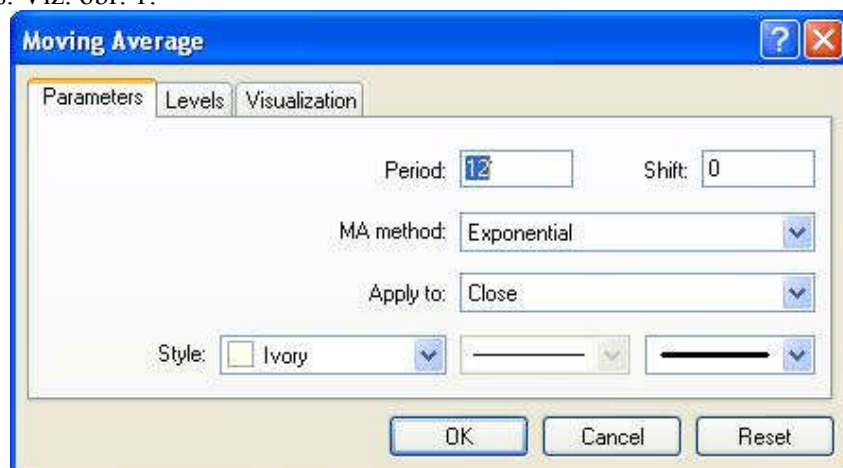
Příklad:

```

extern color Indicator_color = C'0x00,0x00,0xFF'; // blue
int init()
{
    ...
}

```

Zde byla proměnná `Indicator_color` definována jako externí proměnná, kterou uvidíte při prvním přiřazení vašeho indikátoru (nebo EA) ke grafu MetaTraderu, který můžete měnit prostřednictvím okna properties. Viz. obr. 1.



Obr1: Okno Property MA indikátoru

Zde jsou proměnné `Period`, `Shift`, `MA_method`, `Apply_to` a `Style` definovány pomocí klíčového slova “`extern`”, takže se objeví v okně `property`. Jakékoliv proměnné, u kterých si přejete, aby se daly uživatelem měnit, sestavte jako externí proměnné.

Doufám, že se vám lekce líbila.

Uvítám jakékoliv dotazy či připomínky.

S pozdravem  
Coders'Guru  
29 -10 -20 05