

MQL4 COURSE

By Coders' guru
www.forex-tsd.com

Vítejte ve světě MQL 4 funkcí.
Funkce v jakémkoliv jazyce mají dvě fáze:

Jeich učení, které někdy bývá nudné;

Jeich používání, které vždy slouží jako záchranný člun.

Nyní vám chci sdělit svou tradiční větu:

Doufám, že se vám předchozí lekce líbily, stáhnout si je můžete z těchto odkazů:

- 1- Lesson 1 - Welcome to the MQL4 course.
<http://www.forex-tsd.com/attachment.php?attachmentid=372>
- 2- Lesson 2 – SYNTAX.
<http://www.forex-tsd.com/attachment.php?attachmentid=399>
- 3- Lesson 3 - MQL4 Data types.
<http://www.forex-tsd.com/attachment.php?attachmentid=469>
- 4- Lesson 4 - MQL4 Operations & Expressions.
<http://www.forex-tsd.com/attachment.php?attachmentid=481>
- 5- Lesson 5- Loops & Decisions (Part1).
<http://www.forex-tsd.com/attachment.php?attachmentid=504>
- 6- Lesson 6 - Loops & Decisions (Part2).
<http://www.forex-tsd.com/attachment.php?attachmentid=547>

Začněme tedy se sedmou lekcí:

Co je významem slova funkce?

Funkce jsou něco jako zařízení na výrobu klobás. Do jedné části vložíte maso a koření a z druhé vám vyjede klobása.

Masem a kořením jsou zde **funkční parametry**, klobásou je **vrácená hodnota funkce**.

Zařízením je v tomto případě **tělo funkce**.

Mezi funkcí a mašinou na klobásy je pouze jeden rozdíl, a to, že některé funkce nevrátí nic (nic se v MQL 4 nazývá **void**).

Předvedme si několik příkladů:

```
Double //typ klobásy – vrácená hodnota
my_func (double a, double b, double c) // jméno funkce a seznam parametrů (maso a koření)
{
    return (a*b + c);
} // klobása je venku – vrácená hodnota
```

Jak vidíte, funkce začíná **typem** vrácené hodnoty „**double**“, následovaná **jménem funkce** a zakončená svorkami. Mezi svorky vložíte maso a koření, pardon, parametry funkce.

Zde vkládáme trojí parametry: **double a. double b. double c.**

Tělo funkce je pak započato a zakončeno závorkami. V našem případě tělo funkce vyprodukuje operaci (**a*b + c**).

Klíčové slovo return je zde odpovědné za vrácení konečné hodnoty.

Klíčové slovo return

Klíčové slovo return ruší funkci (jako např. **break** u smyčky) a předává řízení **volání funkce** (s tím se brzy seznámíme).

V klíčovém slově **return** může být obsažen výraz ohraničený svorkami jako v příkladu uvedeném výše **return (a*b + c)**. To znamená, že funkce bude zrušena a vrácen výsledek výrazu. Může být také bez výrazu a mít za úkol pouze zrušení funkce.

Poznámka: Ne všechny funkce používají klíčové slovo return, zejména, pokud zde není hodnota ke vrácení.

Jako v tomto příkladě:

```
void // void znamená, že se zde nevyskytují žádné klobásy – vrácené hodnoty.
my_func (string s) // jméno funkce a seznam parametrů (maso a koření)
{
    Print(s);
}
```

Výše uvedená funkce nebude vracet hodnotu, vyobrazí však parametry, které jste jí poskytli. Pokud nemá funkce hodnotu pro vrácení, použijete slovo „**void**“, jako když funkce vrací typ. Tyto typy funkcí jsou u některých programů nazývány **metodami**, v MQL4 je však nazýváme funkcemi.

Volání funkce:

Nyní již velmi dobře víme, co znamená slovo funkce (doufám). Jak používat funkce ve vašem MQL4?

Po zapsání funkce se zde vyskytuje extra krok pro použití funkce ve vašem programu. Krok se nazývá **volání** (její použití).

Předpokládejme, že máme funkci, která shromažďuje součet dvou celých čísel.

Funkce je zde:

```
int collect (int first_number, int second_number)
{
    return(first_number+ second_number);
}
```

Jak funkce pracuje již víte, ale vy ji chcete použít.

To provedete takto:

```
int a = 10;
int b = 15;
int sum = collect(a,b);
Print (sum);
```

V příkladu bude vyobrazena hodnota 25 (není to kouzelné?). Ale jak k tomu došlo? Tím kouzelným řádkem je **int sum = collect(a,b)**; zde jste deklarovali proměnnou (**sum**) k zajištění hodnoty vrácení funkce a funkci jste poskytli dva parametry (**a,b**).

V zásadě jste vyvolali funkci.

Když MQL4 uvidí jméno funkce, převezme vaše parametry, vstoupí do funkce a vrátí – brzy – výsledky a umístí je do stejného řádku.

Je to velmi podobné kopírování všech řádků funkce na místo, kde jste funkci vyvolali, jednoduché, že?

Vnoření funkcí do funkce:

Funkci (nebo více funkcí) můžete vložit do těla jiné funkce. Je to proto, že řádek pro vyvolání je ošetřen jako u běžného příkazu (jedná se vlastně o příkaz).

Příklad:

Použijeme výše popsanou funkci **collect** uvnitř jiné, nové funkce, jejímž úkolem vyobrazení výsledku sběru

```
void print_collection (int first_number, int second_number)
{
    int sum = collect(first_number, second_number);
    Print(sum);
}
```

Zde jsme vyvolali funkci collect uvnitř těla funkce print_collection a vyobrazili výsledek. Void znamená, že se zde nevyskytuje žádná hodnota (pamatujete si to ještě?).

MQL 4 – zvláštní funkce init(), deinit() a start():

V MQL 4 začíná každý program funkcí **init()** (inicializace), která proběhne při přiřazení vašeho programu (Expert Advisor nebo Custom indicator) ke grafům Metatraderu nebo v případě změny finančního symbolu nebo periodicity grafu. Jejím úkolem je inicializace hlavních proměnných vašeho programu (o inicializaci proměnných se dozvíte více v další lekci).

Jakmile program dokončí svou práci nebo pokud zavřete okno grafu či změníte finanční symbol, periodicitu grafu, uzavřete terminál MetaTraderu, provede se funkce **deinit()** (deinicializace).

Třetí funkce (ta nejdůležitější) – **start()** – se aktivuje při každé nově obdržené cenové nabídce. S touto funkcí strávíte 90% vašeho programátorského života.

Více se o těchto funkcích dovíme v opravdových lekcích, kde si zapíšeme vlastní funkce Expert Advisor a Custom indicator).

Doufám, že se vám lekce líbila.

Uvítám jakékoliv dotazy a návrhy.

S pozdravem

Coders'Guru
25-10-2005