

MQL4 COURSE

By Coders' guru
www.forex-tsd.com

-5
Smyčky & Rozhodnutí
Část 2

Vítejte v šesté lekci mého kurzu MQL 4.

Doufám, že se vám předchozí lekce líbily.

V předchozí lekci jsme se bavili o smyčkách. Také jsme viděli, že smyčky jsou jedním ze dvou způsobů sloužících ke změně normálního průběhu provádění programu – seshora dolů. Druhým způsobem jsou rozhodnutí.

Rozhodnutí způsobují v programu jednorázový skok do jiné jeho části, v závislosti na hodnotě výrazu.

Toto jsou druhy příkazů rozhodnutí dostupné v MQL4.

Příkaz IF

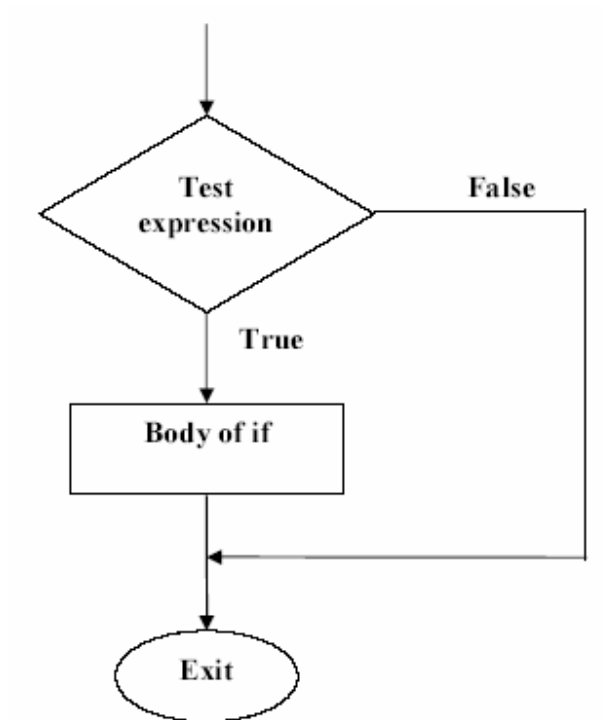
.....

Příkaz **if** je nejjednodušším příkazem rozhodnutí, zde je jeho příklad:

```
if( x < 100 )  
Print("hi");
```

Zde je klíčové slovo **if** následováno závorkami, mezi nimiž je **testovací výraz** (**x < 100**). Pokud je výsledek dosažený testovacím výrazem true, tělo příkazu **if** provede funkci (**Print("hi");**), pokud je false, řízení je předáno příkazu následujícím po bloku **if**.

Obr. 1 znázorňuje průběhový graf příkazu **if**:



Graf

Testovací výraz

False

True

Tělo příkazu if

Výstup

Obr. 1 – Průběhový graf příkazu if

Vícenásobné příkazy:

Jako u smyček, i tělo příkazu **if** může obsahovat více než jeden příkaz. Příkazy jsou vymezeny svorkami.

Např.:

```
if(current_price==stop_lose)
{
    Print("you have to close the order");
    PlaySound("warning.wav");
}
```

Poznámka: Symbol ==v testovacím výrazu; jedná se o jeden ze vztažných operátorů, které jsme probírali v lekci 4, „operace a výrazy“.
Bývá zdrojem mnoha chyb, pokud se zapomenete a použijete operátor =.

Vkládání:

Struktury smyček a rozhodnutí mohou být přesunovány jedna do druhé. Můžete vnořit příkazy **if** do smyčky, smyčky do příkazů **if**, apod.

Zde je uveden příklad:

```
for(int i=2 ; i<10 ; i++)
    if(i%2==0)
    {
        Print("It's not a prime number");
        PlaySound("warning.wav");
    }
```

V tomto příkladu byl příkaz **if** vnořen do smyčky.

*Poznámka: Povšimnete si, že kolem těla smyčky se nevyskytují žádné závorky, což je z toho důvodu, že příkaz **if** a příkazy uvnitř těla jsou považovány za samostatné příkazy.*

Příkaz if...else

.....

Příkaz **if** vás nechá něco vykonat, pokud je podmínka **true**, ale předpokládejme, že v případě hodnoty **false** budeme chtít vykonat něco jiného. V tomto okamžiku přichází čas na příkaz **if...else**.

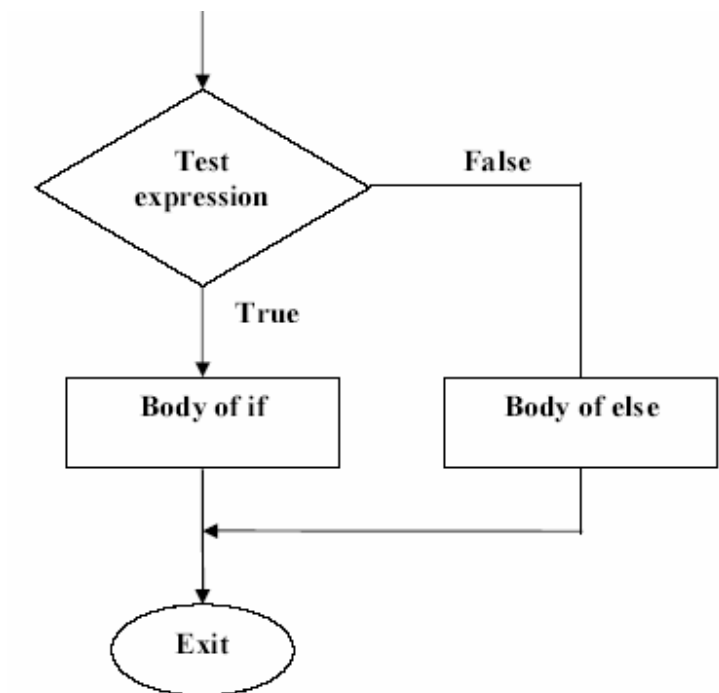
Ten sestává z příkazu **if** následovaného dalším příkazem nebo blokem příkazů, pak přijde na řadu klíčové slovo **else**, následováno dalším příkazem nebo blokem příkazů.

Příklad:

```
if(current_price>stop_lose)
    Print("It's too late to stop, please stop!");
else
    Print("you playing well today!");
```

Pokud je testovací výraz v příkazu **if** **true**, program vyobrazí jednu zprávu, pokud **true** není, vyobrazí druhou.

Obr. 2 vyobrazuje průběhový graf příkazu **if...else**:



Graf

Testovací výraz

False

True

Tělo příkazu if

Tělo příkazu else

Výstup

Obr. 2 – průběhový graf příkazu if...else

Vnořené příkazy if...else

Příkazy **if...else** můžeme vnořit do příkazů **if**, můžete je rovněž vnořit i do vlastních příkazů **if...else** apod.

Jako zde:

```

if(current_price>stop_lose)
    Print("It's too late to stop, please stop!");
if(current_price==stop_lose)
    Print("It's time to stop!");
else
    Print("you playing well today!");

```

U vnořených příkazů **if...else** existuje potenciální problém, když z nepozornosti přiřadíte příkaz **else** ke špatnému **if**.

K vyřešení tohoto problému můžete provést jednu ze dvou věcí:

1- příkaz **if...else** můžete ohraničit závorkami, jako je tomu zde:

```

if(current_price>stop_lose)
{
    Print("It's too late to stop, please stop!");
if(current_price==stop_lose)
    Print("It's time to stop!");
else
    Print("you playing well today!");
}

```

2 – Pokud nemůžete použít první řešení (v případě příliš četného výskytu příkazů **if...else**, nebo se vám do toho nechce), berte toto jako pravidlo.

Přiřaďte **else** nejbližšímu **if** (tento řádek **if(current_price==stop_lose)**)

Příkaz switch

.....

Ppokud máte rozsáhlý strom rozhodnutí a veškerá rozhodnutí jsou závislá na hodnotě stejné proměnné, můžete použít příkaz **switch**.

Příklad:

```
switch(x)
{
  case 'A':
    Print("CASE A");
    break;
  case 'B':
  case 'C':
    Print("CASE B or C");
    break;
  default:
    Print("NOT A, B or C");
    break;
}
```

V příkladu je klíčové slovo **switch** následováno svorkami, uvnitř svorek naleznete **konstantu switch**, tato konstanta bude celým číslem, konstantou znaku nebo konstantou výrazu. Konstanta výrazu nesmí obsahovat proměnnou. Např. **case X+Y** je neplatnou konstantou switch.

Jak výše uvedený příklad funguje?

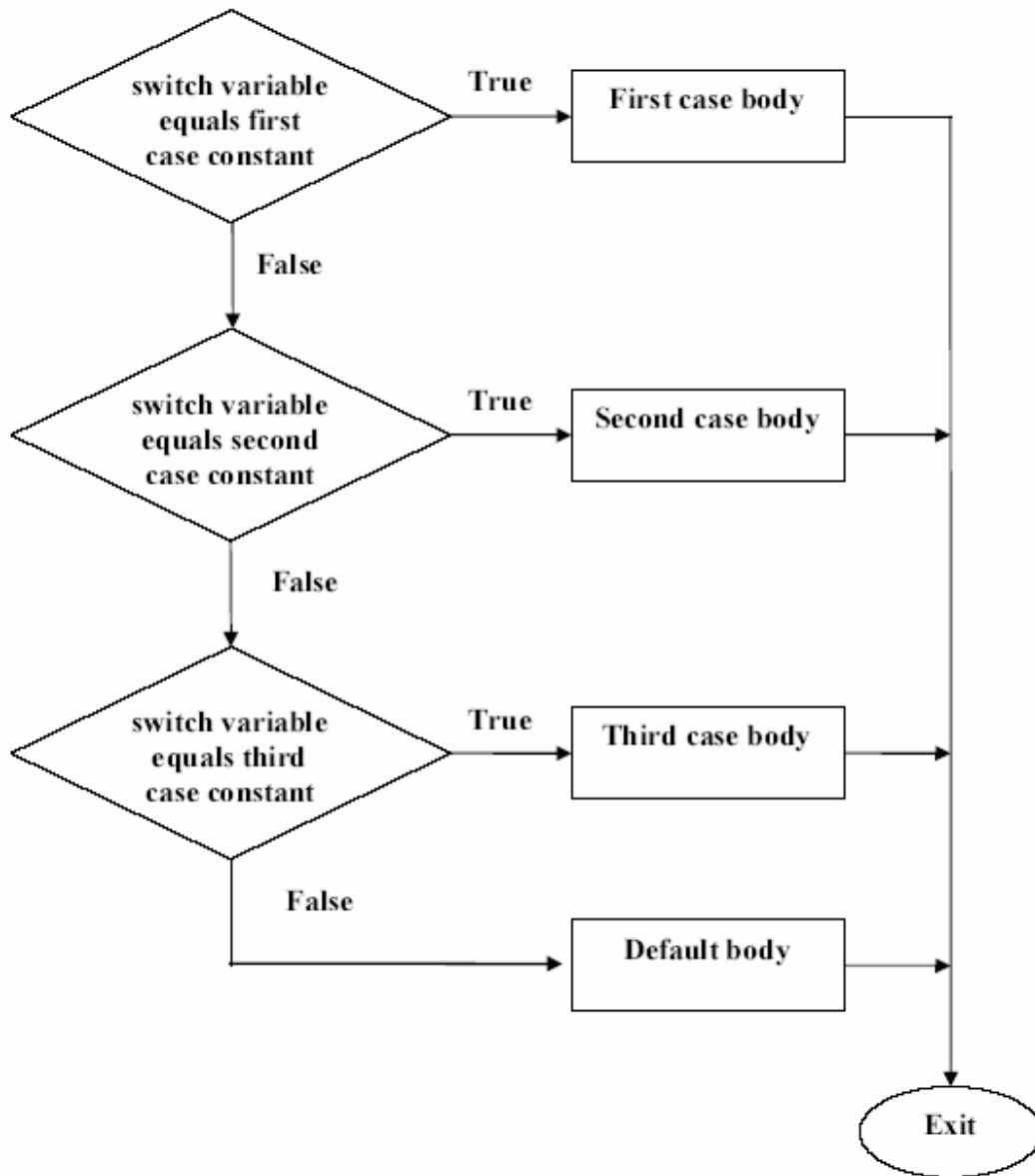
Příkaz switch přiřazuje konstantu **x** konstantám **cases**.

V případě **x=='A'** příkaz vyobrazí "**CASE A**" a příkaz **break** převezme řízení ze switch bloku.

V případech, kdy **x=='B'** nebo **x=='C'** vyobrazí program "**CASE B** nebo **C**". To proto, že v **case 'B'** se nevyskytuje žádný příkaz **break**.

V případě, že **x!=** jakákoliv konstanta case, příkaz switch provede case **default** a vyobrazí "**NOT A, B** nebo **C**".

Obr.3 znázorňuje průběhový graf příkazu switch



Graf

Proměnná switch je rovna první konstantě case

True

Tělo prvního případu case

False

Proměnná switch je rovna druhé konstantě case

True

Tělo druhého případu case

False

Proměnná switch je rovna třetí konstantě case

True

Tělo třetího případu case

False

Tělo default

Výstup

Obr.3 - průběhový graf příkazu switch

Doufám, že se vám lekce líbila.
Uvítám jakékoliv dotazy připomínky.

S pozdravem

Coders'Guru
25-10-2005