

MQL4 COURSE

By Coders' guru
www.forex-tsd.com

(Dodatek 2)

Obchodní Funkce

V tomto dodatku najdete popis 25 MQL4 obchodních funkcí. Rozhodl jsem se napsat tento dodatek před napsáním třetí části lekce "**Váš první Expert Advisor**", protože musíte znát tyto důležité funkce než začnete louskat zbývající kód.

OrderSend:

Syntaxe:

```
int OrderSend( string symbol, int cmd, double volume, double price, int slippage, double stoploss, double takeprofit, string comment=NULL, int magic=0, datetime expiration=0, color arrow_color=CLR_NONE)
```

Popis:

Funkce **OrderSend** se používá pro otevření prodejní / nákupní objednávky nebo nastavení budoucí objednávky.

Funkce vrací: Pokud **proběhla**, vrací **číslo ticketu** této objednávky
Pokud **neproběhla**, vrací **-1**.

Použitím funkce **GetLastError** získáte více informací o chybě.

Pozn: Číslo ticketu je unikátní číslo vrácené funkcí **OrderSend**, toto číslo můžete použít později jako odkaz pro otevření nebo zrušení objednávky (například jej můžete použít pro funkci **OrderClose**, která uzavře příslušnou objednávku)

Pozn: Funkce **GetLastError** vrací předdeklarované číslo posledně vyskytnutých chyb v příslušné operaci (například když zavoláte **GetLastError** po operaci **OrderSend** dostanete chybové číslo které se vyskytnulo pro provedení funkce **OrderSend**).

Volání **GetLastError** nastaví poslední chybové číslo na 0. Uplný seznam MQL4 chyb najdete v souboru **stderr.mqh**.

Použitím funkce **ErrorDescription** (definované v **stdlib.mqh**), můžete získat popis chyby pro určité chybové číslo.

Parametry:

Tato funkce má **11** parametrů.

string symbol:

Symbol příslušného měnového páru který obchodujete (např. EURUSD a USDJPY)

Pozn: Použití funkce *Symbol()* dostane aktuálně používaný symbol a funkcí *OrderSymbol* dostanete symbol pro příslušnou objednávku.

int cmd:

Číslo signalizuje typ operace, kterou chcete provést:

Konstanta	Hodnota	Popis
OP_BUY	0	Koupě pozice
OP_SELL	1	Prodej pozice
OP_BUYLIMIT	2	Nákupní limitní pozice
OP_SELLLIMIT	3	Prodejní limitní pozice
OP_BUYSTOP	4	Buy STOP pro limitní pozici
OP_SELLSTOP	5	Sell stop pro limitní pozici

Pozn: Můžete použít číslo ze sloupce hodnota nebo použít konstantu. Například: *OrderSend(Symbol(),0,...)* je stejné jako *OrderSend(Symbol(),OP_BUY,...)* . Ale doporučuji používat konstanty kód je pak více přehledný.

double volume:

Počet lotů, které chcete obchodovat.

double price:

Cena za kterou chcete otevřít objednávku.

Použijte funkce *Bid* a *Ask* získání konkrétní bid nebo ask ceny.

int slippage:

Hodnota slippage

Pozn: *slippage* je rozdíl mezi předpokládanými náklady za transakci a skutečnými náklady. *Slippage* je obvykle označována jako změna spreadu. (viz. Investopedia.com)

double stoploss:

Cena za kterou chcete uzavřít objednávku v případě ztráty.

double takeprofit:

Cena za kterou chcete uzavřít objednávku v případě profitu.

string comment:

Komentář který chcete uvést u vaší objednávky. (Příklad 1)

Default hodnota je NULL, což znamená bez komentáře.

Pozn: Defaultní hodnota parametru znamená že když příslušný parametr neuvedete použije se právě defaultní hodnota.

int **magic:**

Magic číslo které přiřadíte objednávce.

Pozn: Magic číslo je číslo, které přiřadíte k vaší objednávce, tento odkaz vám umožní rozlišit různé objednávky. Například objednávky které máte otevřeny vašim expert advisor a objednávky které jste otevřeli ručně.



Type	L...	Sy...	Price	S / L	T / P	Time	Price	Swap	Profit	Comment
sell	0.10	eurUSD	1.1722	0.0000	1.1672	2005....	1.1672	0.00	50.00	macc sample[tp]
sell	0.10	eurUSD	1.1722	0.0000	1.1672	2005....	1.1672	0.00	50.00	macc sample[tp]
sell	0.10	eurUSD	1.1721	0.0000	1.1671	2005....	1.1671	0.00	50.00	macc sample[tp]
buy	0.10	eurUSD	1.1724	0.0000	1.1774	2005....	1.1774	0.00	50.00	macc sample
sell	0.10	eurUSD	1.1721	0.0000	1.1671	2005....	1.1671	0.00	50.00	macc sample[tp]

Příklad 1

datetime **expiration:**

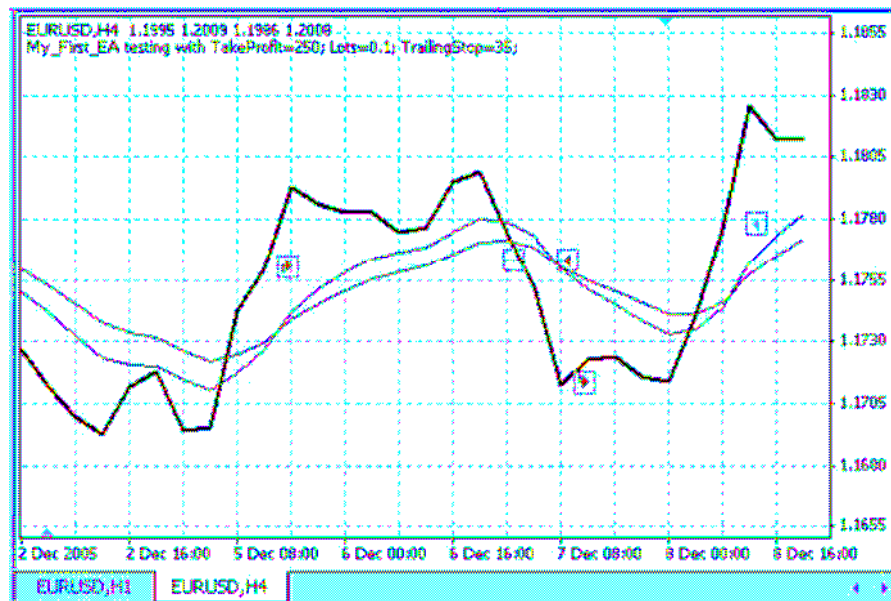
Doba, kdy chcete vaši nevyřešenou objednávku (limitní) označit za neplatnou.

Default hodnota je **0** tzn. že nikdy.

Pozn: Čas který se zda zadává je serverový čas ne váš lokální, aby jste získali aktuální serverový čas použijte funkci *CurTime* a pokud potřebujete lokální čas použijte funkci *LocalTime*.

color **arrow_color:**

Barva otevírací šipky (příklad 2), defaultní hodnota je CLR_NONE , což znamená nebude zobrazena žádná šipka na grafu.



Příklad 2

Příklad:

```

int ticket;
if(iRSI(NULL,0,14,PRICE_CLOSE,0)<25)
{
    ticket=OrderSend(Symbol(),OP_BUY,1,Ask,3,Ask-25*Point,Ask+25*Point,"My
    order #2",16384,0,Green);
    if(ticket<0)
    {
        Print("OrderSend failed with error #",GetLastError());
        return(0);
    }
}

```

OrderModify:

Syntax:

```

bool OrderModify( int ticket, double price, double stoploss, double takeprofit, datetime expiration,
color arrow_color=CLR_NONE)

```

Popis:

Funce **OrderModify** se používá pro upravení vlastností pro určitou otevřenou objednávku nebo limitní objednávku. **Vrací true** pokud byla objednávka úspěšně upravena a **false** když to selže.
Použitím funkce **GetLastError** získáte více detailu o chybě.

Parametry:

Funkce má **6** parametrů

int **ticket**:

Číslo ticketu pro objednávku kterou chcete upravit.

Pozn: To číslo bylo přiřazeno k objednávce funkcí **OrderSend**. Můžete použít funkci **OrderTicket** k získání čísla ticketu aktuální objednávky.

double **price**:

Cena kterou chcete nastavit na objednávce.

Pozn: Použijte fce **OrderPrice** pro získání otevírací ceny aktuální objednávky.

double **stoploss**:

Cena za kterou chcete uzavřít objednávku v případě ztráty.

double **takeprofit**:

Cena za kterou chcete uzavřít objednávku v případě profitu.

datetime **expiration**:

Doba, kdy chcete vaši nevyřešenou objednávku (limitní) označit za neplatnou.

Default hodnota je **0** tzn. že nikdy.

color **arrow_color**:

Barva šipky (příklad 2), defaultní hodnota je CLR_NONE , což znamená nebude zobrazena žádná šipka na grafu.

Příklad:

```
if(TrailingStop>0)
{
    SelectOrder(12345,SELECT_BY_TICKET);
    if(Bid-OrderOpenPrice()>Point*TrailingStop)
    {
        if(OrderStopLoss()<Bid-Point*TrailingStop)
        {
            OrderModify(OrderTicket(),Ask-10*Point,Ask-35*Point, OrderTakeProfit(),0,Blue);
            return(0);
        }
    }
}
```

OrderClose:

Syntax:

```
bool OrderClose( int ticket, double lots, double price, int slippage, color Color=CLR_NONE)
```

Popis:

Funkce **OrderClose** se používá k uzavření příslušné objednávky (číslem ticketu)

Vrací true když je objednávka uzavřena, false když operace selže.

Použijte fce **GetLastError** pro získání více podrobností o chybě.

Parametry:

Funkce má 5 parametrů

int **ticket**:

Číslo ticketu pro příslušnou objednávku kterou chcete uzavřít.

double **lots**:

Počet lotů

Pozn: Použitím funkce **OrderLots** získáte počet lotů pro aktuální objednávku

double **price**:

Cena za kterou chcete uzavřít objednávku.

Použitím fce **Bid** a **Ask** získáte aktuální bid nebo ask cenu

int **slippage**:

Hodnota slippage

color **Color**:

Barva uzavírací šipky, default hodnota je **CLR_NONE** což znamená že žádná šipka se nezobrazí.

Příklad:

```
if(iRSI(NULL,0,14,PRICE_CLOSE,0)>75)
{
    OrderClose(order_id,1,Ask,3,Red);
    return(0);
}
```

OrderSelect:

Syntax:

```
bool OrderSelect( int index, int select, int pool=MODE_TRADES)
```

Popis:

Funkce **OrderSelect** vybere příslušnou objednávku a to na základě ticketu nebo indexu.

Vrací true pokud je objednávka vybrána a false pokud operace selže.

Použijte fce **GetLastError** pro získání více podrobností o chybě.

Pozn: Musíte použít funkci **OrderSelect** před použitím obchodních funkcí které nemají žádný parametr:
OrderMagicNumber, OrderClosePrice, OrderCloseTime, OrderOpenPrice, OrderOpenTime, OrderComment, OrderCommission, OrderExpiration, OrderLots, OrderPrint, OrderProfit, OrderStopLoss, OrderSwap, OrderSymbol, OrderTakeProfit, OrderTicket and OrderType

Parametry:

Tato funkce má **3** parametry.

int **index:**

Index nebo číslo ticketu pro objednávku kterou chcete vybrat.

Záleží na druhém parametru.

int **select:**

Podle čeho vybírat (dle indexu nebo ticketu)

Existují dvě hodnoty:

SELECT_BY_POS: použije pozici (index) pro objednávku.

SELECT_BY_TICKET – použije ticket pro objednávku.

int **pool:**

Pokud použijete *SELECT_BY_POS*, musíte určit kde je zdroj (data):

MODE_TRADES: výběr z aktuálně obchodovaných objednávek (otevřené a limitní). **Tato hodnota je defaultní.**

MODE_HISTORY: výběr z historie (uzavřené a zrušené objednávky).

Příklad:

```
if(OrderSelect(12470, SELECT_BY_TICKET)==true)
{
    Print("order #12470 open price is ", OrderOpenPrice());
    Print("order #12470 close price is ", OrderClosePrice());
}
else
    Print("OrderSelect failed error code is", GetLastError());
```

OrderDelete:

Syntax:

```
bool OrderDelete( int ticket)
```

Popis:

Funkce **OrderDelete** se používá k smazání limitní objednávky. Vrací true pokud je smazána jinak false. Použijte fce **GetLastError** pro získání více podrobností o chybě.

Parametry:

Tato funkce má **1** parametr.

int **ticket**:

Číslo ticketu pro objednávku kterou chcete smazat.

Příklad:

```
if(Ask>var1)
{
    OrderDelete(order_ticket);
    return(0);
}
```

OrderCloseBy:

Syntax:

```
bool OrderCloseBy( int ticket, int opposite, color Color=CLR_NONE)
```

Popis:

Funkce **OrderCloseBy** se používá k uzavření určité objednávky otevřením objednávky v opačném směru.

Funkce vrací true pokud je uzavřena, jinak false

Použijte fce **GetLastError** pro získání více podrobností o chybě.

Parametry:

Tato funkce má **3** parametry.

int **ticket**:

Číslo ticketu pro objednávku kterou chcete uzavřít

int **opposite**:

Číslo ticketu objednávky kterou chcete otevřít v opačném směru.

color **Color**:

Barva uzavírací šipky, default hodnota je *CLR_NONE* což znamená že žádná šipka se nezobrazí.

Příklad:

```
if(iRSI(NULL,0,14,PRICE_CLOSE,0)>75)
{
    OrderCloseBy(order_id,opposite_id);
    return(0);
}
```

OrderType:

Syntax:

```
int OrderType( )
```

Popis:

Funkce *OrderType* vrací typ vybrané objednávky, může to být jeden z následujících: *OP_BUY*, *OP_SELL*, *OP_BUYLIMIT*, *OP_BUYSTOP*, *OP_SELLLIMIT* or *OP_SELLSTOP* (viz *OrderSend* funkce)

Před voláním fce *OrderType* musíte vybrat objednávku pomocí fce *OrderSelect*.

Parametry:

Tato funkce nemá žádný parametr.

Příklad:

```
int order_type;
if(OrderSelect(12, SELECT_BY_POS)==true)
{
    order_type=OrderType();
    // ...
}
else
    Print("OrderSelect() returned error - ",GetLastError());
```

HistoryTotal:

Syntax:

```
int HistoryTotal( )
```

Popis:

Funkce *HistoryTotal* hledá v account history nahrané v terminálu a vrací počet uzavřených objednávek

Pozn: Obvykle používáme tuto funkci s funkcí *OrderSelect* abychom získali informace o určité objednávce z historie.

Parametry:

Tato funkce nemá žádné parametry a vrací počet uzavřených objednávek.

Příklad:

```
// retrieving info from trade history
int i,hstTotal=HistoryTotal();
for(i=0;i<hstTotal;i++)
{
    //---- check selection result
    if(OrderSelect(i,SELECT_BY_POS,MODE_HISTORY)==false)
    {
        Print("Access to history failed with error (",GetLastError(),"");
        break;
    }
    // some work with order
}
```

OrderClosePrice:

Syntax:

```
double OrderClosePrice( )
```

Popis:

Funkce *OrderClosePrice* vrací uzavírací cenu vybrané objednávky.

Objednávky musí být vybrána funkcí *OrderSelect* ještě před tím než se volá funkce *OrderClosePrice*

Parametry:

Tato funkce nemá žádné parametry a vrací data type double (uzavírací cenu vybrané objednávky)

Příklad:

```
if(OrderSelect(ticket,SELECT_BY_POS)==true)
    Print("Close price for the order ",ticket," = ",OrderClosePrice());
else
    Print("OrderSelect failed error code is",GetLastError());
```

OrderCloseTime:

Syntax:

```
datetime OrderCloseTime( )
```

Popis:

Funkce *OrderCloseTime* vrací datum a čas uzavření vybrané objednávky.

Pokud fce vrátí **0** znamená to že objednávka nebyla ještě uzavřena

Objednávka musí být vybrána funkcí *OrderSelect* před tím než je volaná fce *OrderCloseTime*

Parametry:

Tato funkce nemá žádné parametry a vrací datetime data typ (datum a čas uzavření vybrané objednávky)

Příklad:

```
if(OrderSelect(10,SELECT_BY_POS,MODE_HISTORY)==true)
{
    datetime ctm=OrderOpenTime();
    if(ctm>0) Print("Open time for the order 10 ", ctm);
    ctm=OrderCloseTime();
    if(ctm>0) Print("Close time for the order 10 ", ctm);
}
else
    Print("OrderSelect failed error code is",GetLastError());
```

OrderComment:

Syntax:

```
string OrderComment( )
```

Popis:

Funkce *OrderComment* vrací komentář pro vybranou objednávku.

Pozn: Tento komentář přiřazujete objednávce když ji otevírate pomocí fce *OrderSend* nebo když je přiřazen serverem. Server někdy přidá jeho komentář na konec vašeho komentáře.

Objednávka musí být vybrána funkcí *OrderSelect* před tím než je volaná fce *OrderComment*

Parametry:

Tato funkce nemá žádné parametry a vrací dat. typ string (komentář příslušné objednávky)

Příklad:

```
string comment;
if(OrderSelect(10,SELECT_BY_TICKET)==false)
{
    Print("OrderSelect failed error code is",GetLastError());
    return(0);
}
comment = OrderComment();
```

OrderCommission:

Syntax:

```
double OrderCommission( )
```

Popis:

Funkce **OrderCommission** vrací hodnotu komise pro zvolenou objednávku. Objednávka musí být vybrána fce **OrderSelect** před tím než je volána fce **OrderCommission**

Parametry:

Tato fce nemá žádné parametry a vrací dat. typ double (hodnotu komise)

Příklad:

```
if(OrderSelect(10,SELECT_BY_POS)==true)
    Print("Commission for the order 10 ",OrderCommission());
else
    Print("OrderSelect failed error code is",GetLastError());
```

OrderExpiration:

Syntax:

```
datetime OrderExpiration( )
```

Popis:

Funkce **OrderExpiration** vrací datum a čas vypršení vybrané limitní objednávky, kterou jste nastavili pomocí fce **OrderSend**. Objednávka musí být vybrána pomocí fce **OrderSelect** před zavolání fce **OrderExpiration**

Parametry:

Tato funkce nemá žádné parametry a vrací datum a čas

Příklad:

```
if(OrderSelect(10, SELECT_BY_TICKET)==true)
    Print("Order expiration for the order #10 is ",OrderExpiration());
else
    Print("OrderSelect failed error code is",GetLastError());
```

OrderLots:

Syntax:

```
double OrderLots()
```

Popis:

Funkce **OrderLots** vrací počet lotů z vybrané objednávky, to jsme nastavili pomocí fce **OrderSend** (parametr volume)
Objednávka musí být vybrána pomocí fce **OrderSelect** před zavolání fce **OrderLots**

Parametry:

Tato funkce nemá žádné parametry a vrací dat. typ double

Příklad:

```
if(OrderSelect(10,SELECT_BY_POS)==true)
    Print("lots for the order 10 ",OrderLots());
else
    Print("OrderSelect failed error code is",GetLastError());
```

OrderMagicNumber:

Syntax:

```
int OrderMagicNumber()
```

Popis:

Funkce **OrderMagicNumber** vrací Magic číslo vybrané objednávky, toto číslo jste nastavili pomocí fce **OrderSend**
Objednávka musí být vybrána pomocí fce **OrderSelect** před zavolání fce **OrderMagicNumber**

Parametry:

Tato funkce nemá žádné parametry a vrací dat. typ integer

Příklad:

```
if(OrderSelect(10,SELECT_BY_POS)==true)
    Print("Magic number for the order 10 ", OrderMagicNumber());
else
    Print("OrderSelect failed error code is",GetLastError());
```

OrderOpenPrice:

Syntax:

```
double OrderOpenPrice()
```

Popis:

Funkce *OrderOpenPrice* vrací otevírací cenu vybrané objednávky. Objednávka musí být vybrána pomocí fce *OrderSelect* před zavolání fce *OrderOpenPrice*.

Parametry:

Tato funkce nemá žádné parametry a vrací dat. typ double

Příklad:

```
if(OrderSelect(10, SELECT_BY_POS)==true)
    Print("open price for the order 10 ",OrderOpenPrice());
else
    Print("OrderSelect failed error code is",GetLastError());
```

OrderOpenTime:

Syntax:

```
datetime OrderOpenTime( )
```

Popis:

Funkce *OrderOpenTime* vrací datum a čas otevření vybrané objednávky.

Parametry:

Tato funkce nemá žádné parametry a vrací dat. typ datetime

Příklad:

```
if(OrderSelect(10, SELECT_BY_POS)==true)
    Print("open time for the order 10 ",OrderOpenTime());
else
    Print("OrderSelect failed error code is",GetLastError());
```

OrderPrint:

Syntax:

```
void OrderPrint( )
```

Popis:

Funkce **OrderPrint** vytiskne data o vybrané objednávce do export log souboru. Objednávka musí být vybrána pomocí fce **OrderSelect** před zavolání fce **OrderPrint**

Parametry:

Tato funkce nemá žádné parametry a nevrací žádnou hodnotu. (void)

Pozn: *void* znamená že funkce nevrací žádnou hodnotu, takže nemůže ji použít k naplnění proměné, např takto:

```
int i = OrderPrint(); //bez významu, ale kompilátoru to nevadí.
```

Příklad:

```
if(OrderSelect(10, SELECT_BY_TICKET)==true)
    OrderPrint();
else
    Print("OrderSelect failed error code is",GetLastError());
```

OrderProfit:

Syntax:

```
double OrderProfit( )
```

Popis:

Funkce **OrderProfit** vrací profit vybrané objednávky. Objednávka musí být vybrána pomocí fce **OrderSelect** před zavolání fce **OrderProfit**

Parametry:

Tato funkce nemá žádné parametry a vrací dat. typ double

Příklad:

```
if(OrderSelect(10, SELECT_BY_POS)==true)
    Print("Profit for the order 10 ",OrderProfit());
else
    Print("OrderSelect failed error code is",GetLastError());
```

OrderStopLoss:

Syntax:

```
double OrderStopLoss( )
```

Popis:

Funkce *OrderStopLoss* vrací hodnotu stoplossu pro vybranou objednávku, stoploss jste nastavili přes fce *OrderSend* nebo *OrderModify*

Objednávka musí být vybrána pomocí fce *OrderSelect* před zavolání fce *OrderStopLoss*

Parametry:

Tato funkce nemá žádné parametry a vrací dat. typ double

Příklad:

```
if(OrderSelect(ticket,SELECT_BY_POS)==true)
    Print("Stop loss value for the order 10 ", OrderStopLoss());
else
    Print("OrderSelect failed error code is", GetLastError());
```

OrdersTotal:

Syntax:

```
int OrdersTotal( )
```

Popis

Funkce *OrdersTotal* vrací počet otevřených a limitních objednávek. Pokud je toto číslo **0**, znamená to že nejsou otevřeny žádné objednávky ani limitní.

Parametry:

Tato funkce nemá žádné parametry a vrací dat. typ integer

Příklad:

```
int handle=FileOpen("OrdersReport.csv",FILE_WRITE|FILE_CSV,"\t");
if(handle<0) return(0);
// write header
FileWrite(handle,"#", "open price", "open time", "symbol", "lots");
int total=OrdersTotal();
// write open orders
for(int pos=0;pos<total;pos++)
{
    if(OrderSelect(pos,SELECT_BY_POS)==false) continue;
    FileWrite(handle,OrderTicket(),OrderOpenPrice(),OrderOpenTime(),OrderSymbol(),OrderLots());
}
FileClose(handle);
```


OrderSwap:

Syntax:

double **OrderSwap**()

Popis:

Funkce *OrderSwap* vrací hodnotu swap pro vybranou objednávku.
Objednávka musí být vybrána pomocí fce *OrderSelect* před zavolání fce *OrderSwap*.

Parametry:

Tato funkce nemá žádné parametry a vrací dat. typ double

Příklad:

```
if(OrderSelect(order_id, SELECT_BY_TICKET)==true)
    Print("Swap for the order #", order_id, " ",OrderSwap());
else
    Print("OrderSelect failed error code is",GetLastError());
```

OrderSymbol:

Syntax:

```
string OrderSymbol( )
```

Popis:

Funkce *OrderSymbol* vrací symbol měnového páru pro vybranou objednávku. (např. EURUSD and USDJPY).
Objednávka musí být vybrána pomocí fce *OrderSelect* před zavolání fce *OrderSymbol*

Parametry:

Tato funkce nemá žádné parametry a vrací dat. typ string

Příklad:

```
if(OrderSelect(12, SELECT_BY_POS)==true)
    Print("symbol of order #", OrderTicket(), " is ", OrderSymbol());
else
    Print("OrderSelect failed error code is",GetLastError());
```

OrderTakeProfit:

Syntax:

```
double OrderTakeProfit( )
```

Popis:

Funkce *OrderTakeProfit* vrací hodnotu profitu nastavenou na vybrané objednávce.
Objednávka musí být vybrána pomocí fce *OrderSelect* před zavolání fce *OrderTakeProfit*

Parametry:

Tato funkce nemá žádné parametry a vrací dat. typ double

Příklad:

```
if(OrderSelect(12, SELECT_BY_POS)==true)
    Print("Order #",OrderTicket()," profit: ", OrderTakeProfit());
else
    Print("OrderSelect() â ____ ě î è ê _ - ",GetLastError());
```

OrderTicket:

Syntax:

```
int OrderTicket( )
```

Popis:

Funkce *OrderTicket* vrací číslo ticketu pro vybranou objednávku.

Objednávka musí být vybrána pomocí fce *OrderSelect* před zavolání fce *OrderTicket*

Parametry:

Tato funkce nemá žádné parametry a vrací dat. typ integer

Příklad:

```
if(OrderSelect(12, SELECT_BY_POS)==true)
    order=OrderTicket();
else
    Print("OrderSelect failed error code is",GetLastError());
```

Doufám, že obchodní funkce jsou teď jasnější.

Coders' Guru

20-12-2005