

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií



Diplomová práce

**Tvorba obchodní strategie v jazyku MQL (MetaQuotes
Language)**

Michal Borek

© 2015 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Michal Borek

Informatika

Název práce

Tvorba obchodní strategie v jazyku MQL (MetaQuotes Language)

Název anglicky

Creating a business strategy in the language MQL (MetaQuotes Language)

Cíle práce

Cílem práce je zpracování literární rešerše, která bude vytvořena na základě studia vhodné literatury, navržení vhodného prostředí pro subjektivní rozhodování na finančních trzích. Bude navržen proces obchodování na finančních trzích, převeden do programovacího kódu, který umožní exekovat na finančních trzích a tím zautomatizovat velkou část celého procesu obchodování. Hlavním cílem práce je vytvořit programový kód v jazyce MetaQuotes Language, který se bude v oblasti finančních trhů chovat podle investorem předem určených pravidel, popřípadě umožní zanalyzovat omezení, kvůli kterému se kód podle zvolených pravidel chovat nemůže.

Metodika

Teoretická část diplomové práce je založena na sběru, studiu a interpretaci odborných informací. V teoretické části práce bude na základě vytvořené literární rešerše zpracována analýza zvolené oblasti a vytvořeny předpoklady pro zpracování praktické části diplomové práce.

Praktická část práce poté využije informace získané z teoretické části a jejím hlavním přínosem bude analýza problémové oblasti (subjektivně zvolených pravidel) a následná syntéza výsledků z analýz do podoby kódu. Poté bude provedeno srovnávání výsledků subjektivních pravidel a programového kódu nasazených na finančním trhu.

Doporučený rozsah práce

60-80 stran

Klíčová slova

Investor, finanční trhy, Forex, CFD, obchodní strategie, MetaQuotes Language, MetaTrader

Doporučené zdroje informací

BERGERON, Bryan. Essentials of XBRL: Financial Reporting in the 21st Century. USA: Wiley. 1 edition. May 12, 2003. ISBN 978-0471220770

GOLDIE, Daniel C., MURRAY, Gordon S. Co potřebujete vědět o investování. ČR: Pragma. 2011. ISBN 978-80-7349-264-9

HARTMAN, Ondřej, TUREK, Ludvík. První kroky na Forexu. ČR: Computer Press. 2009. ISBN 978-80-251-2006-4

MQL 5 Ltd., MQL 4 Documentation [online]. Rusko: MQL 5 Ltd. [2014-06-20]. Dostupný z WWW: <<http://docs.mql4.com/>>.

Předběžný termín obhajoby

2015/16 ZS – PEF

Vedoucí práce

Ing. Edita Šilerová, Ph.D.

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 11. 3. 2015

Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 11. 3. 2015

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 29. 11. 2015

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Tvorba obchodní strategie v jazyku MQL (MetaQuotes Language)" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 30.11.2015

Poděkování

Rád bych touto cestou poděkoval Ing. Editě Šilerové, Ph.D. za čas strávený při konzultacích a za odborné rady a připomínky, které mi věnovala při psaní této diplomové práce.

Tvorba obchodní strategie v jazyku MQL (MetaQuotes Language)

Souhrn

Tato diplomová práce je zaměřena na problematiku obchodování na finančních trzích pomocí automatických obchodních strategií, které lze vytvořit v jazyku MQL. V teoretické části je popsána struktura a rozdělení finančních trhů a způsoby, kterými lze na nich obchodovat. Dále je definován pojem „obchodní strategie“ a popsán celý proces její tvorby jak v subjektivní podobě, tak v podobě kódu jazyka MQL. V praktické části této práce bude nejprve vybrán rámec, ve kterém chceme obchodní strategii vytvářet a poté navrženo několik variant možných podob strategie, z nichž variantu, která bude disponovat nejlepšími parametry, převedu do podoby automatické obchodní strategie v jazyku MQL.

Klíčová slova: Investor, finanční trhy, Forex, CFD, obchodní strategie, MetaQuotes Language, MetaTrader, technická analýza

Creating a business strategy in the language MQL (MetaQuotes Language)

Summary

This thesis is focused on the financial markets trading by automated trading strategies created in MQL language. The theoretical part describes financial markets structure and typology and it also describes the styles how to trade. This part also defines “trading strategy” concept and shows whole strategy creating process in both discretionary and MQL code form. The practical part brings the frame for trading strategy creation and the variants of the strategy are designed. The variant with best performance are then transformed into MQL automated trading strategy.

Keywords: Investor, financial markets, Forex, CFD, trading strategy, MetaQuotes Language, MetaTrader, technical analysis

Obsah

| | |
|---|----|
| 1 Úvod..... | 10 |
| 2 Cíl a metodika..... | 11 |
| 3 Finanční trhy..... | 12 |
| 3.1 Pojem finanční trh a jeho rozdělení..... | 12 |
| 3.2 Organizovaný trh..... | 14 |
| 3.2.1 Burzovní trh v ČR..... | 14 |
| 3.2.2 Světový burzovní trh..... | 15 |
| 3.3 Neorganizovaný trh..... | 16 |
| 3.3.1 Forex..... | 16 |
| 4 Subjekty na finančních trzích..... | 21 |
| 4.1 Brokeři..... | 21 |
| 4.2 Obchodníci na finančních trzích..... | 23 |
| 4.2.1 Typologie způsobu obchodování..... | 24 |
| 5 Obchodní strategie..... | 31 |
| 5.1 Obchodní strategie a její parametry..... | 31 |
| 5.1.1 Signály technické analýzy..... | 32 |
| 5.2 Metriky obchodní strategie..... | 39 |
| 5.3 Životní cyklus obchodní strategie..... | 41 |
| 5.4 Software pro tvorbu obchodní strategie..... | 43 |
| 6 MetaQuotes Language..... | 46 |
| 6.1 Historie..... | 46 |
| 6.2 Základní pojmy..... | 47 |
| 6.3 Struktura a syntaxe jazyka..... | 49 |
| 6.3.1 Proměnné v jazyku MQL 4..... | 49 |
| 6.3.2 Datové typy v jazyku MQL 4..... | 50 |
| 6.3.3 Operátory a funkce..... | 51 |
| 6.3.4 Průběh vykonávání programu..... | 56 |
| 7 Tvorba obchodní strategie..... | 58 |
| 7.1 Zvážení rizika a kapitálu..... | 58 |
| 7.2 Volba nástrojů a způsob analýzy..... | 60 |
| 7.3 Porovnávání obchodních přístupů..... | 61 |
| 7.4 Detailní testování a ladění strategie..... | 66 |
| 7.4.1 Zhodnocení výsledků strategie na historických datech..... | 73 |
| 8 Závěr..... | 79 |
| 9 Seznam použitých zdrojů..... | 81 |
| 10 Přílohy..... | 83 |

Seznam obrázků

| | |
|---|----|
| Obrázek 3.1: ASK/BID cena..... | 19 |
| Obrázek 4.1: Příklad svíčkového grafu- zvolený časový úsek: 1 hodina..... | 27 |
| Obrázek 4.2: Variace zobrazení svíčkového grafu..... | 27 |
| Obrázek 4.3: Support a resistance na trhu akcií..... | 28 |
| Obrázek 4.4: Pozdní signál k otevření obchodu při prolomení rezistence..... | 29 |
| Obrázek 5.1: Příklad Stop-lossu a Profit-Targetu, umístěného na trh. | 32 |
| Obrázek 5.1: Příklad diskreční analýzy prostého klouzavého průměru (perioda 150) na trhu EUR/USD..... | 35 |
| Obrázek 5.2: Příklad diskreční analýzy křížení klouzavých průměrů na trhu EURUSD.... | 35 |
| Obrázek 5.4: Příklad diskreční analýzy Bollinger bands na trhu..... | 37 |
| Obrázek 5.5: Příklad výpočtu indikátoru ATR na trhu USD/JPY. | 38 |
| Obrázek 5.6: Naznačení formace „Klín“ („Wedge“). | 39 |
| Obrázek 5.7: Příklad vyhodnocení metrik obchodní strategie v programu MetaTrader4 na zvoleném časovém období..... | 40 |
| Obrázek 5.8: Hlavní obrazovka aplikace MetaTrader 4..... | 44 |
| Obrázek 5.9: Hlavní obrazovka aplikace VT Trader (verze 2). | 45 |
| Obrázek 6.1: Architektura aplikace MetaTrader..... | 47 |
| Obrázek 7.1: Zakládání nového demo-úctu v aplikaci Alpari-Demo (modifikovaná aplikace MetaTrader4). | 59 |
| Obrázek 7.2: Příklad výkyvu ceny (tzv. gapu), který může nastat na nízkém (obvykle 1-5 minutovém) timeframu. | 61 |
| Obrázek 7.3: Výsledky nejméně ziskové varianty strategie při hrubém testování. | 66 |
| Obrázek 7.3: Nastavení parametrů pro testování strategie MA_krizeni v komponentě Tester. | 73 |
| Obrázek 7.4: Nastavení velikosti počátečního depositu fiktivního testovacího účtu a povolených typů objednávek. | 74 |
| Obrázek 7.5: Přehled výsledků testování strategie MA_krizeni v období od 1.1.2014 do 1.11.2015. | 75 |
| Obrázek 7.6: Filtrování potenciálních neziskových signálů pomocí diskreční analýzy trhu..... | 76 |
| Obrázek 7.7: Přehled výsledků testování upravené strategie MA_krizeni v období od 1.1.2014 do 1.11.2015..... | 78 |

Seznam tabulek

| | |
|---|----|
| Tabulka 3.1: Obchodní seance..... | 17 |
| Tabulka 4.1: Minimální, průměrný a nejvyšší spread prestižních brokerů..... | 22 |
| Tabulka 4.2: Základní makroekonomické ukazatelé fundamentální analýzy..... | 25 |
| Tabulka 6.1: Základní primitivní datové typy. | 51 |
| Tabulka 7.1: Tabulka variant obchodních přístupů..... | 61 |
| Tabulka 7.2: Výsledky variant navržené obchodní strategie..... | 66 |

1 Úvod

Možnost investovat a z investice v budoucnu profitovat lákala a fascinovala člověka od nepaměti. V posledních desetiletích (nebo možná spíš letech) však došlo v oblasti investování a obchodování na trzích zejména díky technologickému pokroku ke vzniku nových možností, o kterých si dávní investoři a obchodníci mohli pouze nechat zdát. Investování je díky tomu snazší, přesnější s mnohem atraktivnější obchodníkovi. Jelikož mě tato problematika zajímá, rozhodl jsem se vybrat toto téma a soustředit se na oblast, která zmíněné přívlastky ještě zvýrazňuje – a sice možnost automatizace obchodování.

Správně provedený proces automatizace pravidel pro obchodování umožňuje být obchodníkovi mnohem časově i prostorově flexibilnější a v mnohých případech také ziskovější. Tento proces však v sobě neskýtá pouze informačně-technologické řešení, jak by se mohlo zprvu zdát – pro správnou automatizaci procesu obchodování je také důležité mít potřebnou znalost z dalších oborů – zejména pak z problematiky finančních trhů a managementu obchodního kapitálu. O tomto a mnohém dalším potřebném pro úspěšný proces automatizace obchodování na finančních trzích pojednává tato diplomová práce.

2 Cíl a metodika

Podkladem pro splnění cílů této diplomové práce je zpracování literární rešerše, která bude vytvořena na základě studia vhodné literatury a navržení vhodného prostředí pro subjektivní rozhodování na finančních trzích. Bude navržen proces, jak vytvořit obchodní strategii na finančních trzích. Jedním z cílů předložené diplomové práce je vytvořit několik variant strategie a vybrat z ní podle zvolené metriky tu nejvhodnější. Další z cílů je převést vybranou strategii kompletně do programovacího kódu jazyka MetaQuotes Language tak, aby se na finančních trzích chovala podle investorem předem určených pravidel. Tím tedy bude možné zautomatizovat velkou část celého procesu obchodování. Posledním cílem je pokusit se vytvořit takovou strategii, která v předchozím a tomto kalendářním roce (2014 a 2015) dosahovala při testech na historických datech jakéhokoliv zisku. Tím se pokusím splnit to, že proces vytváření strategie generuje v určité míře dostatečně kvalitní strategii (neztrátovou při testech na historických datech).

Teoretická část diplomové práce je založena na sběru, studiu a interpretaci odborných informací. V teoretické části práce bude na základě vytvořené literární rešerše zpracována analýza zvolených oblastí a vytvořeny předpoklady pro zpracování praktické části diplomové práce. Hlavní zpracovávané a analyzované oblasti jsou – finanční trhy, způsob obchodování na finančních trzích, proces tvorby obchodní strategie a programovací jazyk MQL.

Praktická část práce poté využije informace získané z teoretické části a jejím hlavním přínosem bude analýza problémové oblasti (subjektivně zvolených pravidel) a následná syntéza výsledků z analýz do podoby kódu. Bude provedeno vytvoření několika variant pravidel a srovnávání jejich výsledků na finančním trhu. Na základě srovnání pak bude vybrána vhodná varianta strategie, která bude případně rozšířena tak, aby byly splněny všechny cíle předložené diplomové práce.

3 Finanční trhy

První vlastní kapitola této práce specifikuje pojem finanční trh a rozděluje finanční trhy na několik typů. Blíže se pak zabývá zejména neorganizovaným trhem a instrumenty, které je na tomto trhu možné obchodovat, protože tyto instrumenty budou využity v dalších částech této práce.

3.1 Pojem finanční trh a jeho rozdělení

Finanční trh je „systém institucí a instrumentů, zabezpečující pohyb peněz a kapitálu“ ([Revenda a kol., 2004], s. 95). Poptávka a nabídka peněz a kapitálu v tržním systému je pak tvořena subjekty, které na tomto trhu obchodují (viz Kapitola 2). Na finančních trzích se obchoduje s tzv. investičními nástroji (instrumenty). Taxonomii investičních nástrojů nejpodrobněji popisuje § 8a odst. 1 zákona č. 591/1992 Sb., o cenných papírech. Pojem investiční nástroj zahrnuje:

- investiční cenné papíry,
- podílové listy otevřených podílových fondů,
- instrumenty peněžního trhu s výjimkou zejména mezibankovních depozit a úvěrů,
- finanční termínové smlouvy (futures), včetně rovnocenných instrumentů, z nichž vyplývá
- právo na vypořádání v penězích,
- úrokové termínové smlouvy (FRA),
- úrokové a devizové swapy a swapy na akcie a akciové indexy (equity swaps), a
- nákupní a prodejní opce týkající se investičního instrumentu, včetně rovnocenných instrumentů, ze kterých vyplývá právo na vypořádání v penězích; těmi se rozumí zejména devizové a úrokové opce

Zde bych definoval pár pojmů, které v další části práce ještě použiji. Cenný papír je listina, se kterou je spjata nějaké právo. V praxi má toto právo podobu pohledávky, kterou má vlastník cenného papíru vůči emitentovi, který cenný papír vydal. Pojem deviza pak představuje bezhotovostní pohledávku na cizí měnu [Budík, 2011].

Samotné finanční trhy lze dělit podle několika taxonomií [Budík, 2011].:

- Podle teritoriálního hlediska (toto hledisko je však v dnešní době globalizace trhů spíše minoritním):
 - Národní trhy
 - Mezinárodní trhy

- Podle doby splatnosti kapitálu:
 - Peněžní trhy- probíhá zde obchodování s krátkodobým finančním kapitálem (s krátkodobými finančními instrumenty se splatností méně než jeden rok);
 - Kapitálové trhy- obchoduje se zde s dlouhodobým finančním kapitálem se splatností vyšší než je jeden rok;
 - Trh zahraničního kapitálu.
 - Devizový a valutový trh- trh spojený se zahraničním obchodem;
 - Trh drahých kovů.

- Podle emise obchodovaných instrumentů:
 - Primární trh- dochází zde k tzv. emisi instrumentu, tedy tento trh je prvotním místem, kde došlo k prodeji instrumentu;
 - Sekundární trh- obchoduje se s investičními instrumenty, které již byly v minulosti emitovány.
 - Organizovaný trh- trh, na kterém licencovaný subjekt organizuje nabídku a poptávku po investičních instrumentech v souladu s platnou legislativou a podle stanovených pravidel a předpisů;
 - Neorganizovaný trh- je volně přístupným trhem, na kterém poptávka a nabídka instrumentů není organizována žádným nadřazeným subjektem. Na tomto trhu obchodují subjekty/investoři, kteří vlastní určité instrumenty a prodávají je všem, kteří jsou ochotni akceptovat jejich cenu. Účastníci těchto trhů uzavírají obchody často prostřednictvím zprostředkovatelů online, místo klasického „parketového obchodu“. Na tomto trhu není jedno ústřední obchodní místo. Neorganizovaný trh nepodléhá tak přísným normalizovaným pravidlům jako organizovaný trh. Subjekty, které na tomto trhu působí, jsou ovšem většinou držitelé licencí, které potvrzují jejich serióznost a způsobilost k obchodování. Ostatní subjekty tak mají alespoň tuto záruku při vzájemném obchodování [Pedagogická

Pro účely této práce je důležité rozdělení podle emise instrumentů a zejména pak sekundární trh. Ten je možné rozdělit na organizovaný a neorganizovaný, o čemž pojednává podkapitola č. 3.2 a 3.3.

3.2 Organizovaný trh

Jak bylo zmíněno výše, organizovaný trh je více legislativně omezeným a regulovaným trhem. Je možné jej dále rozdělit na [Pedagogická fakulta JCU, 2009]:

- Burzovní trh- speciálním způsobem organizované shromáždění subjektů, které přímo nebo elektronicky obchodují s jasně určenými instrumenty, jasně vymezeným způsobem, vymezených burzovních pravidel a pokynů, v burzou určeném čase. Na burzovním trhu se obchoduje s nejlukrativnějšími cennými papíry;
- Mimoburzovní trh- trh, kde nabídku a poptávku po instrumentech vytváří subjekty, které sice musí být licencované, ale na obchodování samotné nejsou kladeny také přísné požadavky a pravidla jako při obchodování na burze. Je tak umožněno například obchodovat jiné cenné papíry než na burze a to za nižší ceny (a objemy).

Burzy je dále možné dělit na [Budík, 2011]:

- Peněžní – zahrnuje burzy cenných papírů, devizové burzy a burzy finančních derivátů.
- Komoditní - obchod kontraktů souvisejících se surovinami;
- Burzy služeb.

3.2.1 Burzovní trh v ČR

Hlavním českým organizátorem trhu s cennými papíry je Burza cenných papírů Praha

(BCPP) (Burza cenných papírů Praha, a.s.). Na českém trhu ještě existuje tzv. RM-SYSTÉM (RM-SYSTÉM, česká burza cenných papírů a.s.), což je burza pro malé a střední investory. RM-SYSTÉM vznikl v květnu 1993 původně jako mimoburzovní trh, ale na konci roku 2008 došlo k jeho transformaci na burzu [BCPP, 2015].

RM-SYSTÉM se od burzy cenných papírů liší zejména [RM Systém, 2008]:

- Spektrem obchodovaných instrumentů- RM- SYSTÉM umožňuje obchodovat určité akcie, které na BCPP není možné obchodovat- např. Siemens, Volkswagen nebo Intel;
- Přístupem investorů- I po změně charakteru z mimoburzovního na burzovní trh zůstává na RM-SYSTÉMU zákaznický princip namísto členského (obchodují zde kromě licencovaných obchodníků také profesionální investoři);
- Termínem vypořádání obchodů- Vypořádání a změna majitele cenného papíru probíhá na RM-SYSTÉMU okamžitě, zatímco na BCPP za tři dny;
- Podílem na českém kapitálovém trhu- Ten u RM-SYSTÉMU tvoří asi pouze 1%.

Na funkci obou burz dohlíží Česká národní banka, která spolu s ministerstvem financí vytváří legislativní rámec pro jejich fungování.

3.2.2 Světový burzovní trh

Mezi světové burzy s největším objemem obchodů patří zejména [Nývtová – Režňáková, 2007]:

- NYSE – New York Stock Exchange (USA) – největší burza na světě podle tržní hodnoty cenných papírů v USD.
- TSE – Tokyo Stock Exchange (Japonsko)
- NASDAQ – National Association of Securities Dealers Automated Quotations (USA)
- Euronext – Euronext (Evropa)
- LSE – London Stock Exchange (Velká Británie)

- TSX – Toronto Stock Exchange (Kanada)
- FWB – Frankfurt Stock Exchange (Německo)

3.3 Neorganizovaný trh

Neorganizovaný trh neboli Over-the-counter (zkratka OTC) je způsob uspořádání trhu s cennými papíry či jinými instrumenty. Na OTC trzích není obchodování řízeno dozorcím subjektem, který by měl za úkol odpovídat za vypořádání dohodnutých obchodů. Účastníci obchodů se zde domlouvají na podmínkách kontraktů přímo mezi sebou. To umožňuje obchodujícím subjektům určitou volnost, ovšem za cenu vyššího rizika, jelikož na rozdíl od organizovaného trhu, podmínky kontraktů zde nejsou standardizovány [XTB, 2007].

Protože je obtížné vytvořit výčet neorganizovaných trhů, jelikož můžou mít různou podobu, zaměřím se v této práci zejména na dva, se kterými budu pracovat v další části práce:

- Forex - trh pro obchodování s cizími měnami (forex trading), který je zároveň největším a také nejlikvidnějším finančním trhem na světě. Z pohledu finančních trhů můžeme likviditu popsat jako možnost najít v trhu protistranu pro náš obchod. Znamená to, že pokud chce například investor nakupovat a je-li vysoká likvidita, pak snadno najde subjekt prodávající daný instrument. Tento trh svým rozsahem zahrnuje většinu zemí světa [FXStreet, 2015];
- Trh CFD - CFD jsou finanční deriváty, což jsou finanční instrumenty, které umožňují investorům profitovat z nárůstu nebo poklesu cen podkladových instrumentů, aniž by tato podkladová aktiva vlastnil [Kracík, 2013].

3.3.1 Forex

Forex je mezinárodní trh s cizími měnami, jenž umožňuje obchodování s devizami. Výraz Forex je odvozen ze slova FOReign Exchange (zkratka FX). Obrovské objemy obchodů na Forexu tvoří především banky, které spekulují na pohyb cen a snaží se tak vytvořit zisk (viz Kapitola č. 2). Pouze šestinu objemu obchodů na Forexu provádí firmy a vlády, které nakupují a prodávají zboží a služby v zahraničí. Forex je nejlikvidnějším ze všech finančních, kapitálových i komoditních trhů. Průměrný denní obrat činí více než 5,3 biliónů dolarů, což je více než čím disponují New Yorkská a Londýnská burza cenných

papírů dohromady [Hartman – Turek, 2009].

Přestože obchod s měnami existoval po staletí, Forex jako takový vznikl v roce 1973, kdy prezident USA Richard Nixon ukončil politiku zlatého standardu (dolar vázaný pevně na hodnotu zlata). Tato změna vedla ke vzniku pohyblivých směnných kurzů [Hartman – Turek, 2009].

Hlavní subjekty, které na Forexu obchodují jsou [Hartman – Turek, 2009]:

- Centrální banky
- Banky
- Investiční společnosti a korporace
- Spekulanti a investoři- nárůst jejich přístupu na trh je umožněn zejména nízkým možným vstupním kapitálem a moderními komunikačními technologiemi, které usnadňují přístup k trhu
- Vládní organizace

Jedním z důležitých pojmů je tzv. obchodní seance. Obchodní seance je časový rozsah mezi otevřením a uzavřením trhu v určitém geografickém pásmu. Nejdůležitější obchodní centra se nacházejí v New Yorku, Tokiu, Sydney a Londýně (viz Tab. 3.1) (veškeré časy uvedené níže jsou v našem časovém pásmu, tedy UTC+1).

Tabulka 3.1: Obchodní seance. Zdroj: [Hartman – Turek, 2009]

| <u>Město</u> | <u>Začátek</u> | <u>Konec</u> |
|---------------------|-----------------------|---------------------|
| New York | 14.00 | 23.00 |
| Tokio | 1.00 | 10.00 |
| Sydney | 23.00 | 08.00 |
| Londýn | 9.00 | 18.00 |

Je patrné, že určité seance se kryjí. V těchto časových rozmezích tak často dochází k nárůstu likvidity. Klidnější období je charakteristické pro asijskou seanci. Měnový trh je také velmi závislý na obchodní seanci americké burzy (15-22:30). Obchodovat se začíná v

neděli ve 23.00 a končí v pátek ve 22.00 [Hartman – Turek, 2009].

Na Forexu obchodujeme měny v tzv. měnových párech, které nám udávají informaci o ceně jedné měny vůči druhé měně. Všechny měny mají svůj symbol.

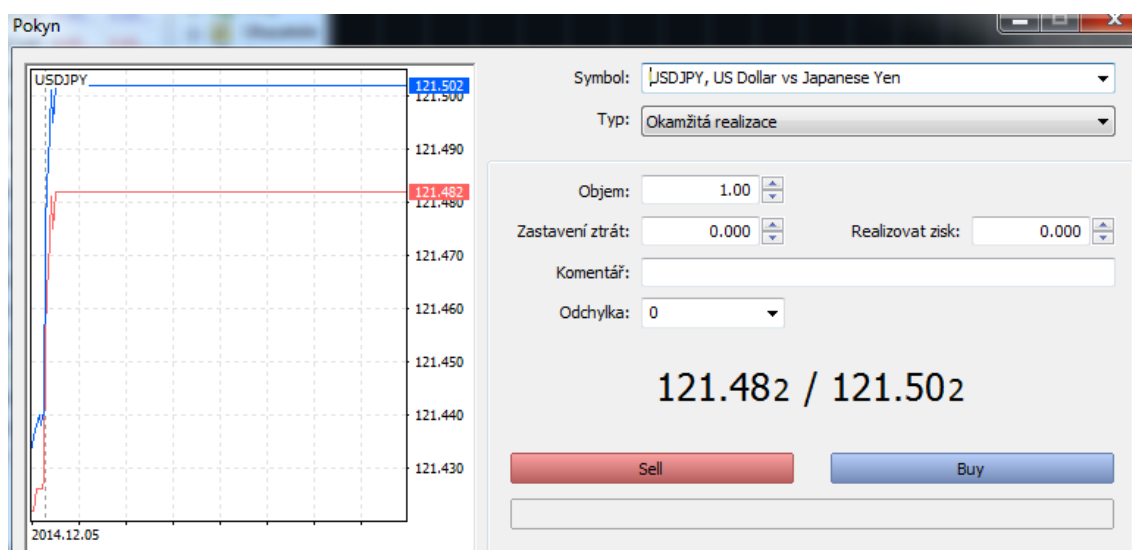
Měny, které můžeme obchodovat lze rozdělit do následujících skupin [Hartman – Turek, 2009]:

- Major (nejčastěji obchodované):
 - USD (americký dolar)
 - EUR (euro)
 - GBP (britská libra)
 - CHF (švýcarský frank)
 - JPY (japonský jen)
- Minor:
 - AUD (australský dolar)
 - DKK (dánská koruna)
 - CAD (kanadský dolar)
 - NZD (novozélandský dolar)
 - NOK (norská koruna)
- Exotické (mimo jiné sem patří i Česká koruna)

Každý pár (např. EUR/CHF) obsahuje měnu základní (EUR) a kótovací (CHF).

3.3.1.1 Nákup a prodej měnových párů

Cena nabídky či „ASK“ cena udává cenu, za kterou měnu kupujeme. Cena poptávky nebo také „BID“ cena označuje finanční hladinu, za kterou bychom měli v daný okamžik měnovou dvojici prodat. Rozdíl mezi těmito cenami se nazývá „SPREAD“. Jestli-že měnu nakupujeme, zajímá nás nabídková cena. Pokud měnu prodáváme, sledujeme poptávkovou cenu (viz Obr. 3.1)



Obrázek 3.1: ASK/BID cena. Obchodní terminál ukazuje cenu, za kterou můžeme prodávat (BID- 121,482) a nakupovat (ASK- 121,502). Zdroj: Metatrader 5

„PIP“ neboli Price Interest Point, vyjadřuje nejmenší možný pohyb měnového páru. Většina měnových dvojic se kótuje na tři až pět desetinných míst. Například měnový pár USD/JPY se kotuje na tři desetinná místa, jeden pip má tedy hodnotu 0,001 bodu. Chceme-li spočítat dolarovou hodnotu jednoho pipu na tomto měnovém páru, použijeme vzorec

$$\text{hodnota pipu} = \text{nejnižší pohyb měnového páru} / \text{kurz}.$$

Pokud bude tedy kurz USD/JPY 110,610, pak dolarová hodnota pipu bude

$$0,001/110,610 = \$ 0,0000090407.$$

Tato hodnota se zdá jako velmi nízká, je však důležité se zmínit, že měnové páry na Forexu se obchodují v tzv. „lotech“. Obchodování s loty úzce souvisí s pákovým efektem, který se při obchodování na Forexu používá. Prakticky to znamená, že zprostředkovatel půjčuje obchodníkovi kapitál, a obchodník kryje tuto půjčku pouze určitým procentem z vypůjčované částky (více viz Kapitola 4). Standardní velikost lotu je \$ 100 000 (existují však i nižší typy lotů). V předchozím příkladu je tedy velikost pipu při obchodování jednoho standardního lotu \$ 0,90407 [Hartman – Turek, 2009].

Velikost pipu při obchodování lotů je důležitá pro výpočet eventuálního zisku, či ztráty z obchodu. Uvažujme, že chceme koupit americký dolar a prodat japonský jen. BID/ASK kurzy jsou 110,595/110,600. Nakupujeme tedy za kurz 110,600, což je kurz, za který jsou ostatní obchodníci ochotni instrument prodat. O chvíli později se kurz pohne na hodnotu

110,610/110,615. Prodáme tedy za hodnotu 110,610 (což je hodnota, za kterou chtějí ostatní obchodníci nakoupit) a uzavřeme tak obchod. Rozdíl mezi nákupní a prodejní cenou je 10 pipů, což je náš zisk. Jelikož dolarová hodnota jednoho pipu je \$ 0,90407, znamená to, že jsme vydělali \$ 9,0407. Je nutné znovu poukázat na „SPREAD“, tedy rozdíl mezi cenou ASK a BID, kterému jsme jako obchodníci vystaveni, a z kterého má zisk zprostředkovatel obchodu (viz Kapitola 4) [Hartman – Turek, 2009].

4 Subjekty na finančních trzích

Druhá vlastní kapitola práce pojednává o subjektech, které se zúčastní obchodů na finančních trzích. Nejprve je popsán způsob práce tzv. brokerů- zprostředkovatelů obchodů. Hlavní přínos této kapitoly však spočívá v definici pojmu „Obchodník na finančních trzích“ a rozboru možných stylů obchodování. Kapitola vychází zejména z [Hartman – Turek, 2009].

4.1 Brokeři

Broker neboli také makléř je společnost (nebo zástupce této společnosti), která obchoduje (tedy nakupuje nebo prodává) na finančním trhu podle pravidel, které si zvolí obchodník. Typicky takový broker vydělává tím, že si za každý zprostředkovaný obchod účtuje určitý poplatek a do nákladů obchodu se obchodníkovi na Forexu promítne ve formě SPREADU. Jedním z důležitých úkolů obchodníka je tak výběr brokera, který umožňuje obchodovat s co nejnižšími spready, přitom však splňuje i další požadavky (viz Tab. 4.1). Mezi tyto požadavky patří zejména [Hartman – Turek, 2009]:

- Regulace- ačkoliv je Forex trhem neorganizovaným, brokeři samotní bývají držitelé určitých licencí, které potvrzují jejich spolehlivost a splnění určitých finančních požadavků. Pokud například obchodník obchoduje u amerického brokera, je vhodné si ověřit, zda je registrován u tzv. Commodity Futures Trading Commission, což je nezávislá vládní organizace, jež vykonává dohled nad subjekty obchodujícími na finančních trzích.
- Komunikace s obchodníkem- pravděpodobně nejdůležitější požadavek, kladený na brokera. Kvalitní broker by měl být obchodníkovi nápomocen 24 hodin denně během obchodních dnů. Broker by měl být dostupný nejen telefonicky a e-mailem, ale ideálně i on-line, formou chatu.
- Obchodní platforma a data- standardně by měla kvalitní obchodní platforma podporovat real-time data trhu a broker by měl nabízet více typů platform v případě nefunkčnosti jedné z nich. Nutnost vytváření určitých analýz z dat v dané platformě pak již do jisté míry závisí na preferencích obchodníka a jeho stylu obchodování (viz Podkapitola 4.2).
- Obchodovatelné instrumenty- broker by měl nabízet veškeré páry obsahující Major i Minor měny.
- Pákový efekt, minimální velikost účtu, minimální velikost obchodu- jak již bylo zmíněno v Kapitole 3.3.1.1, pákový efekt umožňuje obchodníkovi si „půjčit“ od

brokera vyšší částku, než má obchodník aktuálně na účtu. Za tuto půjčku ručí tzv. udržovací marží, tedy jistou výši vkladu, kterou musí mít obchodník neustále na účtu. Pokud je tedy udržovací marže například 1% (1:100) a na účtu má obchodník k dispozici 1000 USD, je možné pro účely obchodování manipulovat se 100 000 USD, tedy obchodovat v celých lotech (viz Kapitola 3.3.1.1). Na druhou stranu, účet obchodníka při vypořádání obchodů nesmí klesnout pod 1000 USD, jinak broker vyzve obchodníka, aby vložil peníze na účet nebo mu udržovací marži zablokuje. Tím broker chrání sebe před ztrátou. Výši páky si volí obchodník sám, typicky se pohybuje mezi 0,5 a 5% [Hartman – Turek, 2009].

V době psaní této práce je u některých brokerů možné otevřít účet i s několika desítky USD a provádět obchody o velikosti jednoho mikrolotu (setina standardního lotu, viz Kapitola 3.3.1.1). Ačkoliv se nízká minimální velikost účtu může zdát zprvu jako výhoda, jelikož je možné si prostředí finančních trhů vyzkoušet s malou částkou, je potřeba se zamyslet nad tím, že hlavním úkolem obchodníka je vytvořit zisk. Obchodník může velmi snadno podlehnout vidině vysokých zisků a s relativně malým účtem se snažit obchodovat vysoké objemy. Je nutné si uvědomit, že pokud obchodník obchoduje celé loty a přitom má neadekvátně nízký vklad, pak každý pohyb (pip) proti jeho směru obchodu, znamená mnohem větší ztrátu, než kdyby zvolil vyšší vklad, popřípadě snížil objem obchodu (například na mikroloty) [Hartman – Turek, 2009].

Tabulka 4.1: Minimální, průměrný a nejvyšší spread prestižních brokerů.

Zdroj: www.investing.com

| <u>Broker</u> | <u>Minimální spread</u> | <u>Průměrný spread</u> | <u>Maximální spread</u> |
|----------------------|--------------------------------|-------------------------------|--------------------------------|
| Liquid Markets | 0.9 | 0.9 | 0.9 |
| Admiral Markets | 1.0 | 1.4 | 2.2 |
| GCI | 0.0 | 1.4 | 6.0 |
| ForexClub | 0.6 | 1.0 | 9.0 |
| DF Markets | 0.8 | 1.4 | 6.0 |
| GO Markets | 1.0 | 1.6 | 2.4 |
| AlpariUK | 1.5 | 1.5 | 1.5 |
| FXPro | 1.0 | 1.6 | 2.4 |
| Ikon | 2.0 | 2.0 | 2.0 |
| FXDD | 1.8 | 2.2 | 13.3 |
| Markets.com | 3.0 | 3.0 | 3.0 |
| Avatrade | 3.0 | 3.0 | 12.1 |

| | | | |
|------------|-----|-----|------|
| FXCM | 0.3 | 2.7 | 11.0 |
| MIG Bank | 1.8 | 2.1 | 7.9 |
| SaxoBank | 2.0 | 2.3 | 6.0 |
| Exness | 0.3 | 1.0 | 7.9 |
| Forex.com | 1.5 | 1.9 | 6.4 |
| FinFX | 0.0 | 1.1 | 9.5 |
| FXOpen ECN | 0.4 | 1.3 | 8.6 |
| FxNet | 0.0 | 0.6 | 11.2 |
| Oanda | 1.0 | 1.7 | 15.0 |
| CitiFX Pro | 1.2 | 1.4 | 14.3 |
| MBTrading | 0.0 | 1.5 | 17.7 |

4.2 Obchodníci na finančních trzích

Mezi subjekty s největším podílem obchodů na finančních trzích patří [IUS, 2012]:

- Instituce sektoru bankovníctví
 - Obchodní banky
 - Nebankovní spořitelní instituce
 - Platební instituce
 - Instituce elektronických peněz
 - Pomocné instituce bankovního sektoru
- Nebankovní instituce sektoru investičních služeb
 - Obchodníci s cennými papíry a deriváty
 - Investiční společnost a investiční
 - Penzijní společnosti a penzijní fondy

V této práci se budeme zabývat zejména rolí obchodníka s deriváty (FOREX), který má na trh přístup přes brokerskou společnost (viz podkapitola 4.1)

4.2.1 Typologie způsobu obchodování

4.2.1.1 Obchodování podle doby držení obchodu

Podle doby držení obchodu můžeme v zásadě rozdělit obchodníky do třech kategorií [Nesnidal – Podhajský, 2009]:

- Krátkodobý obchodník (tzv. daytrader)- jeho cílem je nedržet obchod přes noc. Jeho obchody tedy trvají pouze několik hodin, někdy dokonce pár minut (v extrémních případech pouze pár sekund, tento způsob obchodování se nazývá scalping). Tento obchodní způsob má výhodu ve vyhnutí se držení pozice (neboli otevřeného obchodu) přes noc, což má zejména psychologickou výhodu (výsledek obchodu je mu znám před spánkem). Obchodník drží obchod pouze přes den, neustále sleduje vývoj trhu a chce mít nad obchodem v každou chvíli kontrolu.
- Střednědobý obchodník (tzv. swingový obchodník)- tento obchodník drží své obchody otevřené v řádu několika dní. Swingový obchodník nepotřebuje mít nad obchody neustálou kontrolu a vývoj obchodu vyhodnocuje pravidelně, ale během kratšího časového úseku, než daytrader.
- Dlouhodobý obchodník (investor)- jeho obchody typicky trvají několik měsíců. Obchoduje způsobem nejméně časově náročným na kontrolu obchodů, ovšem většinou mnohem náročnějším na kapitalizaci (výši) účtu. Je to z toho důvodu, že cena instrumentu většinou za dlouhé časové období mnohem více fluktuuje a obchodník tak musí být na tento fakt připraven.

4.2.1.2 Obchodování pomocí fundamentální vs. technické analýzy

Obchodníky lze také rozdělit podle toho, jakým způsobem přistupují k analýze situace na trhu pro zhodnocení, jakým způsobem budou obchodovat. Rozlišujeme dva základní způsoby rozdělení analýzy [Hartman – Turek, 2009] :

- Fundamentální analýza - jejím úkolem je nalézt vnitřní hodnotu instrumentu, což je cena, jenž odpovídá kapitalizovaným výnosům z daného instrumentu. Zjednodušeně řečeno, obchodník se snaží najít cenu, jakou by mohl instrument v budoucnu mít a porovnává jej se současnou reálnou cenou. Pokud vidí v tomto

rozdílu potenciál, pak instrument zobchoduje. Fundamentální analýza není exaktní věda, vychází totiž z chování lidí, které neobsahuje vždy zákonité vazby. Konkrétní způsob analýzy pak velmi závisí na typu instrumentu, který obchodník obchoduje. Například zemědělské komodity jsou velmi ovlivněny nabídkovou stranou. Obchodník se tak snaží odhadnout, jaká bude sklizeň, zda na ni nebudou mít vliv přírodní jevy a podobně. Měny (FOREX) jsou zase ovlivněny ekonomickou a politickou situací příslušných zemí. Podstata spočívá v tom, že čím lépe si vede ekonomika dané země, tím více spatřují okolní země její měnu jako výhodnou investici, více měnu nakupují a tím roste její kurz. Důležité je pak tuto informaci zhodnotit v kontextu s chováním ostatních ekonomik. Obchodníci na měnových trzích, kteří používají fundamentální analýzu sledují tzv. makroekonomické ukazatele, což jsou „statistické údaje vyjadřující aktuální stav hospodářství určitého státu v závislosti na konkrétní oblasti dané ekonomiky (průmysl, trh práce, obchodní bilance atd.). Jsou zveřejňovány pravidelně v určitou dobu vládními agenturami a soukromým sektorem“ [Markets.com, 2015].

Popis nejdůležitějších makroekonomických ukazatelů obsahuje Tabulka 4.2.

Tabulka 4.2: Základní makroekonomické ukazatele fundamentální analýzy. Zdroj: <http://www.markets.com/cz/education/fundamental-analysis/main-economic-indicators.html>

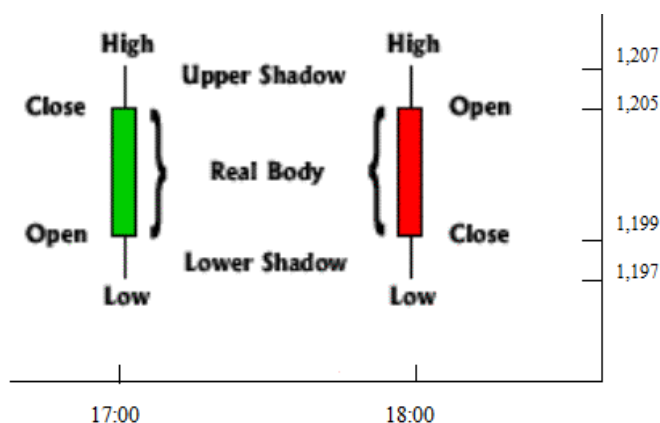
| |
|--|
| Vyhlašování úrokových sazeb |
| Nejdůležitější nástroj pro ovlivňování kurzu měny. Jsou ovlivňovány rozhodnutím centrální banky. Úrokové sazby v podstatě určují, kam se budou přesouvat investice. Rozdíly v úrokových sazbách jednotlivých států mají vždy relativní dopad na určitou měnu vůči měně jiné. |
| Hrubý domácí produkt |
| HDP je základním měřítkem ekonomiky země. Představuje celkovou tržní hodnotu veškerého zboží a služeb vyprodukovaných ve zvolené zemi a roce. Někteří obchodníci sledují také předběžné kalkulace HDP, které jsou zveřejněné před samotným konečným HDP. Tyto ukazatele se nazývají „advance report“ a „preliminary report“. Rozdíl mezi těmito dvěma reporty mohou způsobit velké pohyby kurzů párů, obsahujících danou měnu. |
| Ukazatele zaměstnanosti |
| Ukazatele zaměstnanosti poukazují na celkové zdraví ekonomiky. Pro zhodnocení stavu ekonomiky je zapotřebí vědět, kolik pracovních míst se vytváří či ruší, procentuální zastoupení aktivně pracujících a počet nových žádostí o příspěvky v nezaměstnanosti. |
| Maloobchodní prodej |
| Ukazatel maloobchodního prodeje je vyhlášen každý měsíc a pro obchodníka s měnovými páry je podstatný v tom, že hodnotí celkovou míru spotřebitelských výdajů a výsledky maloobchodníků. Používá se k předvídání výsledků důležitějších ukazatelů a ke zhodnocení nejbližšího směřování dané ekonomiky. |

| |
|---|
| Platební bilance |
| Platební bilance reprezentuje poměr mezi platbami přijatými ze zahraničí a platbami odchozími do zahraničí. Pokud příchozí platby překračují platby odchozí, je platební bilance kladná, v opačném případě záporná. Přebytek může předvídat možný růst dané měny. |
| Daňová a měnová politika |
| Stabilizaci ekonomiky (např. plná zaměstnanost, řízení inflace a vyrovnaná platební bilance) se vlády pokoušejí dosáhnout pomocí kontroly daňové a měnové politiky. Daňová politika se zaměřuje na problematiku daní a výdajů, měnová politika souvisí s nabídkou úvěrů, peněz a jiných finančních aktiv. |

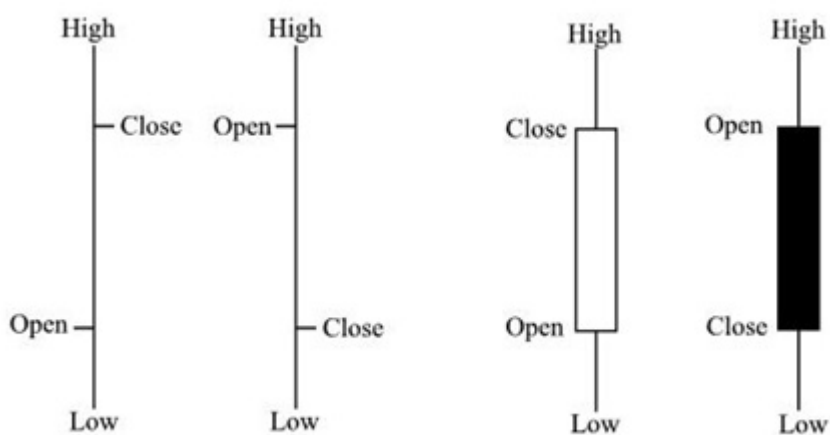
- Technická analýza – je založena na předpokladu, že veškeré informace potřebné pro úspěšné obchodování, jsou vyjádřeny v současné tržní ceně. Není tedy nutné studovat jednotlivé makroekonomické faktory, jako v případě zkoumání fundamentálních vlivů. Technická analýza vychází zejména ze dvou otázek:
 - „Jaká je současná cena instrumentu?“
 - „Jaká je historie vývoje ceny tohoto instrumentu?“

Technická analýza považuje historický vývoj ceny za významný faktor, jelikož je postavena na domněnce, že historický vývoj ceny se pravidelně opakuje a trhy se tak pohybují podle vypočitatelných vzorců. Technická analýza tedy zkoumá tyto vzorce a snaží se v nich nalézt signály, kdy je pro obchodníka nejvhodnější obchodovat. Předpokládá se totiž, že cena se pohybuje v tzv. trendech – buď stoupá vzhůru, klesá dolů, nebo se pohybuje směrem do strany. Pokud se podaří obchodníkovi objevit trend, předpokládá se, že tento pohyb bude dále pokračovat [Hartman – Turek, 2009].

Jelikož praktická část této práce je založena právě na využití technické analýzy, je vhodné se zmínit o jejích silných a slabých stránkách. Zřejmě nejdůležitější silnou stránkou je možnost snadné a zřetelné interpretace technické analýzy do grafů. Mezi obchodníky je nejvíce rozšířené využití tzv. svíčkových grafů. Svíčkový graf zobrazuje 4 důležité cenové úrovně pro zvolený časový úsek (svíčku) – otevírací cenu (Open), nejnižší cenu (Low), nejvyšší cenu (High) a uzavírací cenu (Close) (Obrázek 4.1 se zvoleným časovým úsekem – tzv. timeframeem – 1 hodina a variace grafů – Obrázek 4.2) [Hartman – Turek, 2009].



Obrázek 4.1: Příklad svíčkového grafu- zvolený časový úsek: 1 hodina. Graf zobrazuje následující cenovou akci- v 17:00 byla cena instrumentu EUR/USD na úrovni 1,199 (Open). V průběhu 17:00- 17:59 cena klesla nejnižší na úroveň 1,197(Low) a vzrostla nejvýše na úroveň 1,207 (High). V 17:59 byla cena instrumentu 1,205 (Close). Cena v 18:00 byla taktéž 1,205, mezi 18:00 a 18:59 cena kolísala mezi úrovněmi 1,197 a 1,207 a v čase 18:59 byl instrument přesně na cenové úrovni 1,199. Svíčka, která má Close výše než Open, bývá vyplněna zelenou barvou popř. je nevyplněna barvou; naopak svíčka, která má Close níže než Open, bývá vyplněna červenou barvou, popř. je vyplněna barvou. Zdroj: Autor



Obrázek 4.2: Variace zobrazení svíčkového grafu. Nalevo je zobrazení hodnot Open a Close pomocí odrážek, napravo pomocí výplně těla svíčky. Zdroj: <http://www.financial-spread-betting.com/course/candle-stick-charting.html>

Síla svíčkového grafu spočívá v možnosti efektivně a jednoduše zobrazit chování poptávkové a nabídkové strany trhu. High dané svíčky (časového úseku) ukazuje sílu poptávky (kupujících), Low svíčky pak sílu nabídky (prodávajících). U první svíce na

Obrázku 4.1 například můžeme soudit, že poptávka byla v průběhu časového rozmezí 17:00-17:59 silná, nabídka ustupovala a nechala tedy cenu významně vyrůst. Close úroveň pak reprezentuje cenu, na které se prodávající a kupující na konci této hodiny shodli [Hartman – Turek, 2009].

Další výhodou grafické reprezentace je možnost snadného určení úrovní tzv. supportu a rezistence. Tyto úrovně ohraničují určité rozpětí, ve kterém se cena dlouhodobě pohybuje. Průlom horní (rezistence) nebo spodní hranice (supportu), pak znamená změnu v rozložení sil na trhu a pro obchodníka možný signál pro vstup na trh – viz Obrázek 4.3 (o strategiích a signálech pro vstup do obchodu více v Kapitole 5) [Hartman – Turek, 2009].



Obrázek 4.3: Support a rezistence na trhu akcií. Tyto úrovně určují oblast, ve které se cena dlouhodobě (několik měsíců) pohybovala. V tomto případě souvisel průlom rezistence s masivním nárůstem poptávky a růstu ceny instrumentu – obchodník však může podobnou situaci interpretovat jinak – záleží na jeho preferencích. Zdroj: <http://www.swing-trade-stocks.com/support-and-resistance.html>

S poslední větou však souvisí jedna z hlavních slabých stránek technické analýzy – nejednoznačná interpretace jejích výsledků. Stejný vzorec na grafu je mnohdy možné interpretovat mnoha způsoby. Technickou analýzu tedy nelze brát jako exaktní disciplínu, ale obchodník by si z ní měl vzít pouze prvky, které jsou pro něj pochopitelné, použitelné a zejména takové, se kterými je schopný dosahovat požadovaných výsledků [Hartman – Turek, 2009].

Dále je technická analýza kritizována názorem, že její výsledky poskytují zpožděný pohled na trh. To je do jisté míry pravda, jelikož jakmile vznikne určitý vzorec pro možný vstup obchodníka na trh, je obvykle velká část pohybu ceny instrumentu již dokončena – viz Obrázek 4.4



Obrázek 4.4: Pozdní signál k otevření obchodu při prolomení rezistence. Na obrázku je patrná rezistence na úrovni 11500, která byla prolomena 17. dubna – pro obchodníka využívající technickou analýzu to může znamenat poměrně jednoduchý signál pro vstup do obchodu. Je však důležité si uvědomit, jak důležitá část pohybu ceny instrumentu obchodníkovi již unikla, pokud by vstoupil na trh již od počátku cenového pohybu (začátek cenového pohybu je na úrovni 9500). Zdroj: <http://www.tradergods.com/know/TA/article/KA00000005.aspx>

Je potřeba ale brát v potaz, že příliš brzký vstup do obchodu obvykle znamená vyšší riziko neúspěchu. Obchodník si tedy musí vybrat mezi potenciálně ziskovějším obchodem, který ale s vyšší pravděpodobností může být ztrátový, nebo mezi obchodem, kde již podstatná část pohybu ceny byla dokončena, ovšem správnost volby vstupu na trh je již podpořena předchozí cenovou akcí. Pro obchodníka je tedy důležité si pravidla pro vstup statisticky vyhodnotit a zvolit, která z možností je pro něj výhodnější [Hartman – Turek, 2009].

4.2.1.3 Mechanické vs. diskreční obchodování

Rozdíl mezi mechanickým a diskrečním obchodováním spočívá ve způsobu obchodování podle obchodníkem předem určených pravidel. V případě mechanického obchodování má obchodník k dispozici výčet předem jasně určených pravidel, pomocí kterých vstupuje a vystupuje z obchodu. Tato pravidla vytvářejí ve své podstatě algoritmus a tak je možné je převést do podoby kódu v programovacím jazyce. Pro obchodníka to znamená možnost

vytvoření teoreticky automatického programu, který se na trhu chová přesně tak, jak jej je obchodník hodlá (a je schopný) vytvořit [Financnik.cz, 2005].

Diskréční obchodování je sice také postaveno na jasných pravidlech, nicméně obchodník si nechává jakýsi „prostor“ pro zohlednění jistých (často nepříliš pravděpodobných) vlivů, které mohou na trhu nastat. Tento faktor vychází z velké části ze zkušeností obchodníka. Obchodník může při tomto způsobu obchodování využít tak svůj „cit“ pro trh a zahrnout do obchodování adaptabilní složku, kterou lze přímo do mechanického obchodování těžko vložit [Financnik.cz, 2005].

Nevýhodou mechanického obchodování je nutnost neustále revidovat pravidla a přizpůsobovat je aktuálnímu chování trhu, což je jeden z faktorů, který zkušený diskréční obchodník dokáže. Naproti tomu zásadní slabou stránkou diskréčního obchodování je vliv emocí a psychologie, který je mnohdy na výsledky obchodníka (mnohdy i zkušeného) patrný. Jelikož mechanické obchodování je řízeno striktně danými pravidly (které většinou na trhu automaticky obchoduje aplikace), nepodléhá psychologickým vlivům – na rozdíl od diskréčního obchodování, kdy například negativní emoce po sérii několika ztrátových obchodů mohou způsobit, že obchodník přestane dodržovat (jinak fungující) pravidla pro obchodování [Financnik.cz, 2005].

5 Obchodní strategie

Tato kapitola pojednává o obchodních strategiích – vysvětluje pojem obchodní strategie, definuje základní parametry, kterými se obchodní strategie vyznačují. Kapitola se zaměřuje na popis strategií s využitím technické analýzy, která bude aplikována ve zbytku této práce. V této kapitole jsou také popsány metriky, podle kterých lze strategie porovnávat, z nichž některé budou použity později v praktické části práce. Závěr kapitoly je pak určený pro představení fází životního cyklu obchodních strategií.

5.1 Obchodní strategie a její parametry

Obchodní strategie je přesně stanovený plán, podle něhož se obchodník řídí na trhu. Tento plán obsahuje přinejmenším tyto základní části [Financnik.cz, 2005]:

- Cíl obchodování – obchodník si volí cíl (obvykle zisk), kterého chce dosáhnout v rámci určitého období
- Technické parametry obchodování – velikost účtu pro obchodování, výběr brokera
 - Minimální velikost účtu pro obchodování na finančních trzích, která je obvykle doporučována je 10 000 USD [Financnik.cz, 2005].
- Způsob otevírání a uzavírání obchodu – signál pro vstup a výstup – viz podkapitola 5.1.1
- Nastavení úrovní profit targetu a stop-lossu (viz Obrázek 5.1)
 - Profit-target je cena, na níž chceme uzavřít naši otevřenou pozici (neboli obchod) se ziskem (než se taková pozice může obrátit do ztráty) [Financnik.cz, 2005]
 - Stop-loss je předem definovaná krajní cena, při které dobrovolně inkasujeme malou ztrátu dříve, než se taková rozroste do mnohem větší ztráty [Financnik.cz, 2005].



Obrázek 5.1: Příklad Stop-lossu a Profit-Targetu, umístěného na trh. Pokud nakoupíme instrument na úrovni 460 pipů, poté stop-lossem jasně definujeme, že nejhůře na úrovni 440 pipů instrument prodáme – tím definujeme, že v nejhorsím případě inkasujeme ztrátu 20 pipů. Profit target na úrovni 470 pipů znamená, že pokud cena vzroste na tuto úroveň dříve než by klesla na úroveň 440 pipů (Stop-loss), pak obchod skončí v zisku 10 pipů.
Zdroj: Autor

- Definice časového rámce, ve kterém obchodování probíhá – týká se zejména krátkodobého obchodování, kdy obchodník jasně určuje, ve které hodiny během dne bude vstupovat do obchodů
- Pravidla money-managementu a risk-managementu [Financnik.cz, 2005]:
 - Money-management znamená ve zkratce řízení toku peněz (příjmů a výdajů) a s tím spojené řízení obchodů; v praxi to znamená především volba velikosti objemu obchodů v závislosti na aktuální velikosti obchodníkovy účtu
 - Risk-management se soustředí na minimalizaci ztrát, zejména pak těch, které by mohly mít na obchodování extrémní dopad (což se dá prakticky velmi efektivně omezovat Stop-lossem)
- Další pravidla neuvedená výše – např. maximální počet obchodů během dne (nebo jiného sledovaného období); maximální počet ztrát během dne (nebo jiného sledovaného období), po němž přestaneme do konce sledovaného období obchodovat

5.1.1 Signály technické analýzy

Technická analýza obsahuje řadu nástrojů, které nám mohou pomoci v rozhodování, zda otevřít či uzavřít obchod. Tyto nástroje lze rozdělit do dvou hlavních skupin [Hartman – Turek, 2009]:

- Indikátory
- Cenové formace (neboli tzv. patterny)

Indikátor je „matematickým výpočtem, jenž lze aplikovat na cenu nebo objem daného aktiva. Jeho výsledkem je hodnota, která se používá k odhadu budoucích změn ceny“ ([Hartman – Turek, 2009], s. 51). Indikátory lze pak rozdělit do několika podskupin v závislosti na tom, na jakém typu dat jsou aplikovány a jakým způsobem vypočítávají výslednou hodnotu. Základní podskupiny indikátorů jsou [Hartman – Turek, 2009]:

- Monetární indikátory – jako jediné souvisejí s fundamentální analýzou. Soustředí se na ekonomická data, zejména pak úrokové míry, inflace, výše korporátního dluhu
- Trendové indikátory – používají se k určení dlouhodobých směrových pohybů (slovo trend vyjadřuje tendenci pokračování cenového pohybu v jistém směru – nahoru nebo dolů, pokud není přítomný trend, trh se pak nachází v tzv. chopu – pohybuje se směrem do strany)
 - Nejpoužívanější trendové indikátory: klouzavé průměry, MACD
- Oscilátory – na rozdíl od trendových indikátorů, nezkontrolují, zda pohyb ceny bude pokračovat v trendu, ale naopak, zda-li se trend otočí.
 - Nejpoužívanější oscilátory: RSI, Stochastic
- Indikátory objemu – měří tržní sílu. Jako vstupní data používají tržní objem obchodů, nebo počet otevřených obchodů na daném trhu.
 - Nejpoužívanější indikátory objemu: On balance volume, MFI (index peněžního toku)
- Indikátory volatility – měří volatilitu, což je velikost cenových pohybů bez ohledu na jejich směr.
 - Nejpoužívanější indikátory volatility: Bollinger bands, Envelopes, ATR

Pro ukázkou fungování indikátorů za účelem nalezení vstupního signálu budou využity dva typy indikátorů: trendový indikátor klouzavý průměr a indikátor volatility Bollinger bands. Dále je v této práci prezentován indikátor ATR, který bude v praktické části práce použit pro nastavení Stop-lossu a Profit-Targetu vytvářené strategie.

Klouzavý průměr je indikátor, který reprezentuje průměrnou cenu aktiva za zvolené časové období. Výpočet klouzavého průměru je tak vlastně matematická analýza průměrné hodnoty aktiva za časový úsek. Pro účely obchodování na finančních trzích se používá

nejčastěji 5 typů klouzavých průměrů: prostý, exponenciální, triangulární, variabilní a vážený [Hartman – Turek, 2009]. V rámci této práce budeme později (zejména v Kapitole 7) pracovat se 2 typy [Financnik.cz, 2005]:

- Prostý klouzavý průměr, jehož matematické vyjádření zní:

$$SMA = \frac{p_1 + p_2 + \dots + p_n}{n}$$

kde n je perioda (neboli počet svíček, ze které indikátor počítáme) a p_n je cena n -té svíčky na grafu.

- Exponenciální klouzavý průměr:

$$EMA = (p_n * Exp) + (p_{(n-1)} * (1 - Exp)) + \dots$$

kde n je perioda (neboli počet svíček, z jejichž cen indikátor počítáme), p_n je cena n -té svíčky na grafu a $Exp = 2/(n+1)$.

Klouzavý průměr je možné počítat pro jakoukoliv cenovou hodnotu svíčky (Open, Low, High, Close), nejčastěji se však používá pro hodnoty Close. Je to jeden z nejpoužívanějších indikátorů technické analýzy vůbec.

Ačkoliv existuje mnoho způsobů, jak analyzovat chování klouzavých průměrů aplikovaných na instrument trhu pro účel vstupního signálu, nejvíce se používají dva způsoby – křížení ceny (svíček) s klouzavým průměrem nebo křížení samotných klouzavých průměrů. Na Obrázku 5.2 je patrné křížení ceny s klouzavým průměrem ve směru shora dolů. Taková situace bývá obvykle považována za vstupní prodejní signál (tzv. obchod Short). Opačně, křížení ceny s klouzavým průměrem ve směru zdola nahoru by znamenalo vstupní nákupní signál (tzv. obchod Long). Volba periody klouzavého průměru pak samozřejmě záleží na preferenci obchodníka a na porovnání výsledků strategie s různými nastaveními [Hartman – Turek, 2009].

Druhým, mezi obchodníky často používaným typem vstupního signálu, je křížení samotných klouzavých průměrů (viz Obrázek 5.3). V tomto případě se obvykle analyzuje chování dvou nebo tří klouzavých průměrů, přičemž jeden z nich bývá exponenciální, s nízkou periodou (5-20) a citlivěji reaguje na změnu ceny samotného instrumentu (proto se mu také říká „rychlý“). Druhý klouzavý průměr (popřípadě také třetí) je prostý, s vysokou periodou (obvykle 50 a více) a je více rezistentní při změně ceny instrumentu

(říká se mu „pomalý“, je-li mezi klouzavým průměrem s nejnižší a nejvyšší periodou další klouzavý průměr s periodou, nacházející se mezi dvěma krajními, pak se označuje jako „středně rychlý“). Vstupní nákupní signál poté nastává, pokud „rychlý“ klouzavý průměr překříží pomalý klouzavý průměr ve směru zdola nahoru (popřípadě pokud rychlý i středně rychlý klouzavý průměr překříží pomalý klouzavý průměr zdola nahoru). Opačně, vstupní prodejní signál nastává, pokud „rychlý“ klouzavý průměr překříží pomalý klouzavý průměr ve směru shora dolů (popřípadě pokud rychlý i středně rychlý klouzavý průměr překříží pomalý klouzavý průměr shora dolů) [Hartman – Turek, 2009].



Obrázek 5.8: Příklad diskreční analýzy prostého klouzavého průměru (perioda 150) na trhu EUR/USD. Na příkladu je zobrazen prodejní vstupní signál, při němž cena instrumentu EUR/USD (reprezentována zeleným svíčkovým grafem) překřížila zvolený klouzavý průměr shora dolů. Zdroj: Metatrader 4



Obrázek 5.9: Příklad diskreční analýzy křížení klouzavých průměrů na trhu EURUSD. Na příkladu jsou zobrazeny tři klouzavé průměry – červený „rychlý“ exponenciální klouzavý průměr (s periodou 10), žlutý „středně rychlý“ prostý klouzavý průměr (s periodou 50) a modrý „pomalý“ prostý klouzavý průměr (s periodou 150). Vstupní nákupní signál nastal ve chvíli, kdy červený i žlutý klouzavý průměr již překročil modrý klouzavý průměr. Zdroj:

Metatrader 4

Indikátor Bollinger bands se na cenovém grafu zobrazuje jako tři křivky – prostřední křivka je normální klouzavý průměr z n posledních uzavíracích cen (Close) časových úseků (svíček) [Hartman – Turek, 2009]:

$$\text{prostredniKrivka} = \frac{\sum_{j=1}^n \text{Close}}{n}$$

Horní křivka vychází z prostřední křivky (klouzavého průměru), jen je posunuta o k standardních odchylek nahoru:

$$\text{horniKrivka} = \text{prostredniKrivka} + \left(k + \sqrt{\frac{\sum_{j=1}^n (\text{Close} - \text{prostredniKrivka})^2}{n}} \right)$$

Spodní křivka je analogicky klouzavý průměr posunutý dolů o k standardních odchylek:

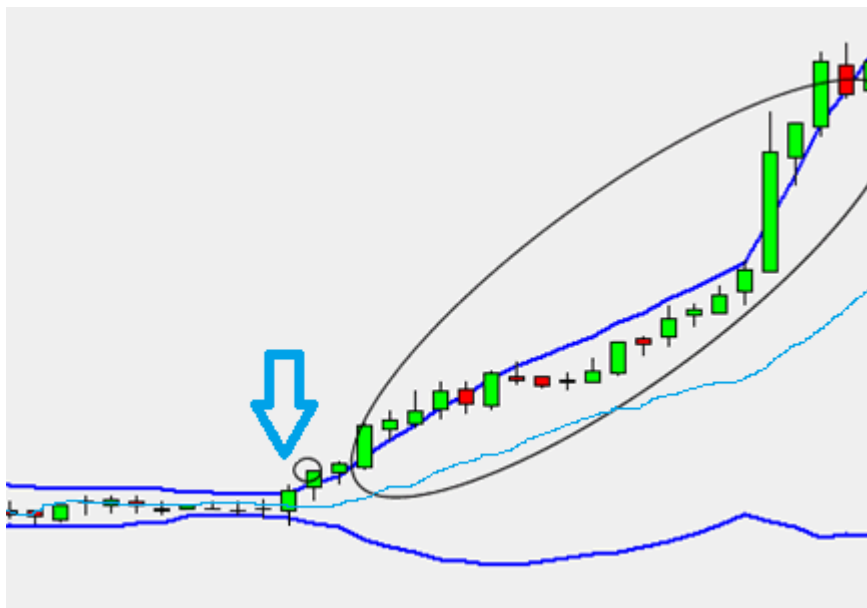
$$\text{spodniKrivka} = \text{prostredniKrivka} - \left(k + \sqrt{\frac{\sum_{j=1}^n (\text{Close} - \text{prostredniKrivka})^2}{n}} \right)$$

Způsobů jak využít tento indikátor pro vstupní i výstupní signály je mnoho – jedno z obecně používaných pravidel zní:

„Po zúžení Bollinger bands se mají tendenci objevovat prudké cenové změny“ [Hartman – Turek, 2009].

Platnost tohoto pravidla lze poukázat na Obrázku 5.4. Pro obchodníka (a zejména obchodujícího mechanicky) je však potřeba toto pravidlo ještě detailněji specifikovat, aby

bylo možné jej obchodovat (viz např. pravidlo právě na Obrázku 5.4).



Obrázek 5.4: Příklad diskréční analýzy Bollinger bands na trhu. Je patrné, že v době zúžení všech 3 křivek Bollinger bands, dochází k minimálnímu cenovému pohybu, ale trh se „připravuje“ na velký cenový pohyb. Lze tak navrhnout vstupní signál – pokud bude kanál Bollinger bands natolik úzký, že se obě krajní křivky dotknou svíčky uvnitř, lze očekávat, že přijde významný pohyb (což se na tomto příkladu vyplnilo). Směr pohybu, pro který by se obchodovalo, by pak mohl vycházet z toho, jaká křivka se stále dotýká další svíce – v tomto případě horní (v černém kroužku), proto by se v tomto případě instrument nakupoval.
Zdroj: Autor

Indikátor ATR (Average True Range) patří stejně jako Bollinger Bands mezi indikátory volatility. Určuje hodnotu průměrného cenového rozpětí. To znamená, že čím nižší volatilita trhu nastává, tím nižší hodnota ATR vzniká, a naopak. Hodnota ATR pro konkrétní svíčku (časový úsek) se pak konkrétně počítá jako nejvyšší z těchto tří hodnot [Etrading, 2015]:

- High uvažované svíčky (časového úseku) – Low uvažované svíčky (časového úseku)
- ABS (High uvažované svíčky – Close předchozí svíčky)
- ABS (Low uvažované svíčky – Close předchozí svíčky)

Nad takto vzniklými hodnotami ATR pro jednotlivé svíčky se pak obvykle vytváří klouzavý průměr se zvolenou periodou (pro n hodnot) (Obrázek 5.5). Jelikož hodnota ATR reflektuje aktuální situaci (rozpětí ceny) na trhu, je tato hodnota ideální jako nástroj pro

určení Stop-lossu a Profit-Targetu. Oproti pevně volené hodnotě má takto určený Stop-loss (nebo Profit-target) tu výhodu, že lépe odpovídá chování kurzu a není příliš vysoký (v případě Profit-targetu) nebo nízký (v případě ochranného Stop-lossu). Stop-loss nebo Profit-target určený tímto způsobem se pak počítá jako hodnota příslušného prvku klouzavého průměru ATR (adaptabilní složka) vynásobená určitou konstantou, zvolenou obchodníkem tak, aby průměrný předpokládaný Stop-loss odpovídal zvolenému Risk-managementu (většinou obchodník omezuje maximální ztrátu do 5% velikosti účtu) [Nesnidal – Podhajsky, 2009]. Takto spočítaná hodnota se pak:

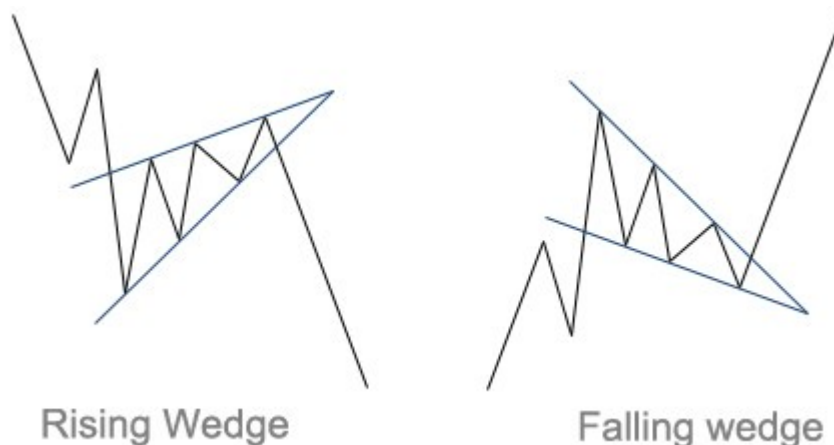
- přičítá k nákupní vstupní ceně – v případě určení úrovně Profit-targetu
- odečítá od nákupní vstupní ceny – v případě určení úrovně Stop-lossu
- odečítá od prodejní vstupní ceny – v případě určení úrovně Profit-targetu
- přičítá k prodejní vstupní ceně – v případě určení úrovně Stop-lossu



Obrázek 5.5: Příklad výpočtu indikátoru ATR na trhu USD/JPY. V oblasti 1 je patrné, že jednotlivé svíčky (timeframe 1 minuta), mají velmi malé cenové rozpětí – respektive rozdíl mezi hodnotou High a Low je v rámci těchto svíček menší oproti svíčkám v oblasti 2. Tato skutečnost se projevuje na hodnotách indikátoru ATR – zatímco v oblasti se hodnoty ATR pohybují okolo hodnoty 0,003 – 0,004, v oblasti se hodnota zvýšila až na hladinu 0,008. Zdroj: Metatrader 4

Další skupinou nástrojů technické analýzy jsou cenové formace. Cenové formace vyžadují mnohem více diskrečního přístupu, protože na rozdíl od indikátorů nevycházejí tolik z exaktních výpočtů. Jelikož cenových formací existuje velké množství, soustředil bych se v rámci této práce pouze na jednu z nich a tou je formace „Klín“ („Wedge“) (viz Obrázek 5.6). Formace „Klín“ vychází z podobné myšlenky jako Bollinger bands – pokud se pohyb ceny instrumentu zužuje do určitého kanálu, lze očekávat, že po prolomení tohoto kanálu, dojde k výraznějšímu pohybu ceny. Směr tohoto pohybu pak závisí na tom,

v jakém směru byl vytvořený kanál prolomen. V čem je oproti použití klouzavých průměrů nebo Bollinger bands rozdíl, je způsob jakým tuto formaci rozpoznat – zatímco u indikátorů je průlom křivky jasně definovatelný, v rámci formace „Klín“ je tento průlom značně subjektivní a tudíž také diskreční. Obchodník by měl zvážit například kolik by formace měla obsahovat svíček a zda analyzovat formaci „Klín“ na základě celých cenových svíček nebo například pouze podle uzavíracích cen [Hartman – Turek, 2009].



Obrázek 5.6: Naznačení formace „Klín“ („Wedge“). Tato formace se vytváří, pokud se cenový pohyb instrumentu odehrává v rámci určitého kanálu, přičemž tento kanál se zužuje. Po prolomení tohoto kanálu pak obchodujeme ve směru pohybu prolomení. Zdroj: <http://www.marketgeeks.com/stock-chart-analysis/>

5.2 Metriky obchodní strategie

Metriky obchodní strategie poskytují informaci o výkonnosti strategie ve sledovaném časovém období. Existuje obrovské množství metrik, pro účely této práce bych však chtěl detailněji popsat pouze následující [MQL4, 2015] (Obrázek 5.7):

- Bars in test – počet svíček obsažený v testu obchodní strategie
- Modeling quality – vyjadřuje míru v jaké granularitě dat se strategie vytváří/testuje. Čím nižší časový úsek je použit, tím je % Modeling quality vyšší
- Initial deposit – počáteční peněžní vklad obchodníka
- Total Net Profit, Gross Profit, Gross Loss – Celkový zisk z obchodů (neboli tzv. Gross Profit) – Celková ztráta z obchodů (neboli tzv. Gross Loss)
- Profit Factor – poměr mezi Celkovým ziskem z obchodů a Celkovou ztrátou z obchodů (hodnota menší než 1 představuje ztrátovou strategii)

- Maximal Drawdown – nejhlubší propad účtu do chvíle kdy dojde k novému nejvyššímu stavu účtu (vyjádřený v obchodované měně)
- Relative Drawdown - nejhlubší propad účtu do chvíle kdy dojde k novému nejvyššímu stavu účtu (vyjádřený v procentech Initial deposit)
- Total Trades – celkový počet obchodů strategie (podle zvolených pravidel) v testovaném časovém období
- Short/Long Positions (Won %) – počet Short/Long obchodů strategie v testovaném časovém období (v závorce procento ziskových Short/Long obchodů)
- Profit/Loss Trades % - procento Ziskových/Ztrátových obchodů ze všech obchodů strategie v testovaném časovém období
- Largest Profit/Loss Trade – největší ziskový/ztrátový obchod vyjádřený v obchodované měně
- Average Profit/Loss Trade – průměrný ziskový/ztrátový obchod vyjádřený v obchodované měně
- Maximum consecutive wins/losses – největší počet po sobě jdoucích ziskových/ztrátových obchodů

| | | | | | |
|---------------------|---------|-----------------------------|-----------------|------------------------------|----------------|
| Bars in test | 5034 | Ticks modelled | 3458232 | Modelling quality | 90.00% |
| Mismatched chart... | 0 | | | | |
| Initial deposit | 2000.00 | | | | |
| Total net profit | 310.31 | Gross profit | 912.56 | Gross loss | -602.26 |
| Profit factor | 1.52 | Expected payoff | 8.17 | | |
| Absolute drawdo... | 867.93 | Maximal drawdown | 887.99 (43.9... | Relative drawdown | 43.96% (887... |
| Total trades | 38 | Short positions (won %) | 19 (63.16%) | Long positions (won %) | 19 (21.05%) |
| | | Profit trades (% of total) | 16 (42.11%) | Loss trades (% of total) | 22 (57.89%) |
| | Largest | profit trade | 501.32 | loss trade | -249.62 |
| | Average | profit trade | 57.04 | loss trade | -27.38 |
| | Maximum | consecutive wins (profit... | 3 (634.40) | consecutive losses (loss ... | 6 (-112.54) |
| | Maximal | consecutive profit (cou... | 634.40 (3) | consecutive loss (count ... | -319.63 (3) |
| | Average | consecutive wins | 2 | consecutive losses | 2 |

Obrázek 5.7: Příklad vyhodnocení metrik obchodní strategie v programu MetaTrader4 na zvoleném časovém období (viz Podkapitola 5.4 a Kapitola 6). Zdroj: Metatrader 4

5.3 Životní cyklus obchodní strategie

Tato kapitola má za cíl vystihnout, jakým způsobem vzniká (popřípadě i zaniká) obchodní strategie. Je důležité podotknout, že snahou je vytvořit strategii, která má šanci na reálném trhu fungovat (ideálně být zisková) a proto by měla být postavena na určitých pravidlech – viz také Podkapitola 5.1. Vývoj strategie by pak měl ideálně projít těmito jednotlivými kroky (o což se pokusím v praktické části této práce) [Financnik.cz, 2005]:

- 1) Zvážení rizika a kapitálu – Stejně tak jako u jakékoliv formy podnikání je pro obchodníka důležité zhodnotit, jaký má kapitál k dispozici a jakou částku je tak ochoten riskovat. Podle toho se řídí celý další proces vytváření strategie. Jelikož v rámci této práci budu prezentovat vytvářené strategie na demo účtu, není zde kapitál omezením. Při reálném obchodování by se ale obchodník měl řídit přinejmenším základním pravidlem o omezení maximální ztráty na jeden obchod. Obvykle si profesionální obchodníci jako maximální přípustnou ztrátu volí 1-5% z celého kapitálu dostupného na účtu (a podle této částky si pak nastavují stop-loss na příslušném trhu). Toto pravidlo jim tak umožňuje rozdělit riziko na více obchodů a nezkrachovat po několika málo nevydařených obchodech [Financnik.cz, 2005].
- 2) Volba nástrojů a způsobu obchodování – V této fázi by měl získat obchodník hrubou představu, jakým způsobem by chtěl obchodovat. Měl by být rozhodnutý, zda chce obchodovat diskrétně/mechanicky, v jakém časovém rámci a zda pomocí fundamentální nebo technické analýzy. Těmito atributům by měl přizpůsobit své další hledání vhodné obchodní strategie a vybírat nástroje, které mu v dalších fázích pomohou v obchodování [Financnik.cz, 2005]
- 3) Nalézání a porovnávání obchodních přístupů + hrubé testování – Obchodník už pracuje pod určitým rámcem (fáze 2), má nápady, jaké by mohl používat obchodní přístupy a porovnává je mezi sebou – v této fázi zejména pravidla, za jakých podmínek vstupovat do obchodů a kdy z nich vystupovat. Cílem je soustředit se zejména na hrubý test funkčnosti zvolených pravidel na širším spektru trhů – tedy na tzv. test robustnosti strategie. Pokud totiž pravidla fungují na větším počtu rozmanitých trhů, dá se předpokládat, že strategie může uspět i při možných změnách podmínek na trzích v budoucnu. Obchodník testuje pravidla (prozatím jen vstupní a výstupní podmínky) na historických datech, které jsou poskytovány brokerem – jedná se o tzv. backtesting. V této fázi je rozumné otestovat vstupní pravidla na několika desítkách obchodů na každém ze zahrnutých trhů (zpětně). Po této fázi by obchodník měl mít odpovědi na to, jaký trh a časový rámeček je pro jeho strategii nejvhodnější a jaká pravidla vykazují ze všech kombinací z backtestů nejlepší výsledky [Financnik.cz, 2005].

- 4) Detailní testování + ladění strategie – Tato fáze slouží již k detailnímu testování strategie na určeném trhu a timeframeu a doladění pravidel- zejména z pohledu money-managementu. Smyslem je otestovat strategii na masivním vzorku historických obchodů – minimálně 100 obchodech (ideálně pokud je to možné, tak více). Jelikož hledání nejvhodnějších parametrů může být velice náročné (časově i výpočetně), má v této fázi výhodu spíše obchodník, který má v plánu obchodovat mechanicky – jelikož pravidla pro vstupy a výstupy lze napsat v programovém kódu a poté spustit jako program, který v příslušné aplikaci zobrazí výsledky strategie na historických datech (nejpoužívanější aplikace jsou popsány v Podkapitole 5.4) [Financnik.cz, 2005].
- 5) Pilotní nasazení na demo-úctu popřípadě malém reálném účtu – Testování na historických datech je důležitá věc, obchodník však potřebuje vědět, že strategie je schopná fungovat i v reálném a „živém“ prostředí trhů. Z toho důvodu by nasazení strategie „naživo“ mělo vždy předcházet odzkoušení na tzv. demo-úctu – tedy účtu, na kterém se pracuje s fiktivními penězi. Demo účet má tu výhodu, že ve většině případů dokonale simuluje reálné chování trhu a obchodník se tak může dost spolehlivě připravit na skutečné fungování trhu. Navíc je v dnešní době dostupný pro většinu typů trhů, se kterými investoři běžně pracují – včetně Forex trhu, na kterém se bude vytvářet strategie v této práci. Obchodník v době obchodování na demo účtu zkusí strategii již v podobě, se kterou plánuje obchodovat. Soustředí se zejména na to, jestli strategie, poskytuje při simulaci na demo-úctu přibližně stejnou obchodní statistiku, jako při detailním testování a zda jsou pravidla navržená v dřívějších fázích vůbec realizovatelná na prostředí pohybujících se trhů. Obchodování na demo-úctu by mělo probíhat alespoň několik měsíců (ideálně alespoň 3). Tato doba je smysluplná pro to, aby si mohl být obchodník jistý, že strategie může na trhu opravdu fungovat stejně, jako ji navrhl v předchozích fázích vývoje. Tato fáze je zejména důležitá pro diskrétní obchodování, jelikož to je právě velmi závislé na živém chování trhu. Z toho důvodu by měl obchodník používající diskrétní analýzu také zvážit, zda mezi obchodování na demo-úctu a reálném účtu s předpokládaným kapitálem, nezkusit také obchodovat na reálném účtu s menším kapitálem. Obchodník má tak možnost vyzkoušet si psychologický vliv obchodování s reálnými finančními prostředky, který má na výsledky mnohdy obrovský vliv [Financnik.cz, 2005].
- 6) Nasazení strategie do prostředí reálného trhu – Teprve až po řádném otestování na historických datech a simulaci na demo účtu by strategie měla být nasazena do prostředí reálného trhu a dosahovat tak cílů, které si obchodník stanovil (zpravidla zisku). Úkolem obchodníka je také v tomto období výsledky strategie monitorovat a statisticky vyhodnocovat. Pokud se metriky strategie po nasazení výrazně liší, je nutné uvažovat o vypnutí strategie [Financnik.cz, 2005].
- 7) Vypnutí strategie – K vypnutí strategie dochází, pokud strategie přestane plnit cíl, kvůli kterému ji obchodník vytvářel. Obchodník by měl mít předem stanovené

pravidlo, při kterém dojde k vypnutí strategie – typicky tímto pravidlem bývá nedosažení zisku v jednom nebo více časových obdobích po sobě nebo například překročení určitého násobku maximálního drawdownu zjištěného v rámci detailního testování [Financnik.cz, 2005].

5.4 Software pro tvorbu obchodní strategie

V rámci této práce se budu zabývat pouze softwarem, který je určený pro analýzu a tvorbu strategií na trhu FOREX. Existuje nespočet aplikací, které jsou sice v plné míře použitelné pro analýzu FOREX trhu a jednodušeji či složitěji použitelné pro podporu tvorby a obchodování strategie, ale primárně jsou určené pro použití na akciových či derivátových trzích. Proto se budu soustředit spíše na aplikace, které jsou určeny výhradně pro analýzu a obchodování na trhu FOREX.

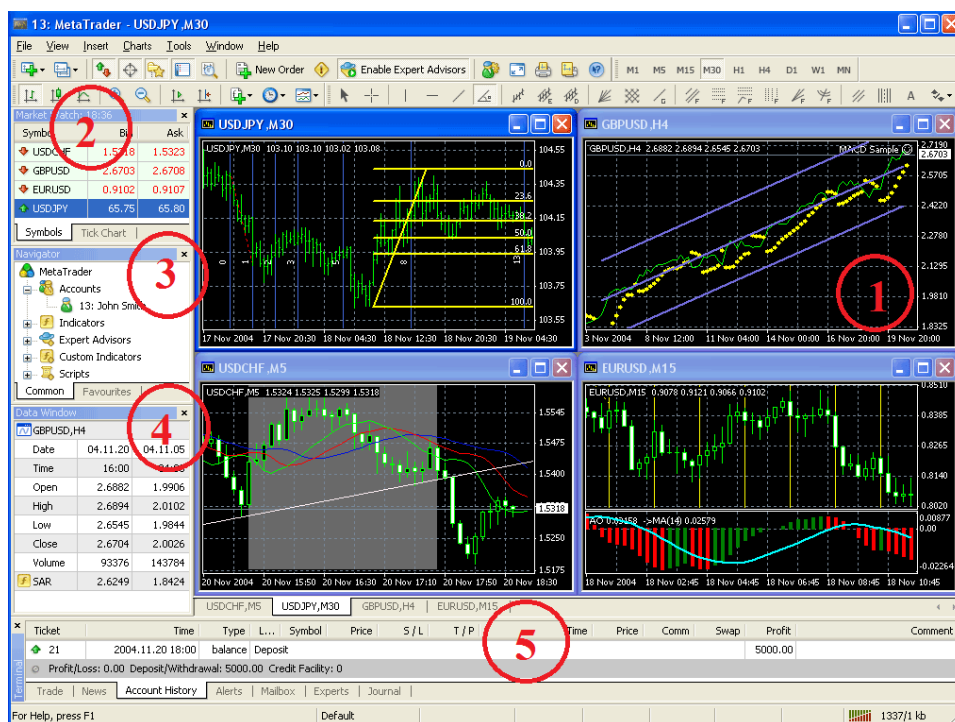
Pro obchodování na trhu FOREX jsou globálně nejvíce využívány zejména dvě aplikace: MetaTrader a VT (Visual Trading) Trader [Hartman – Turek, 2009]. Pokud blíže srovnáme tyto dvě aplikace, zjistíme, že mají mnoho společných prvků, které jsou pro obchodování podstatné (viz Obrázek 5.8 a Obrázek 5.9):

- Obchodování přímo z cenového grafu – možnost využití technické analýzy v plné míře a real-time. Obě aplikace poskytují v reálném čase informace o cenách instrumentů a přenášejí je na grafy technické analýzy. Na grafu jsou zobrazeny také informace o ceně, na které byl obchod otevřen, úrovně stop-lossů a a profit targetů
- Možnost uživatelského nastavení interface – uživatel může libovolně nastavovat pozici na obrazovce cenovým grafům, panelu s informací o otevřených obchodech a cenách instrumentů, odebírat tyto panely, přidávat je
- Předdefinované technické indikátory – obě aplikace defaultně obsahují desítky až stovky technických indikátorů, které může obchodník využít pro identifikaci vstupních a výstupních signálů své strategie
- Možnost tvorby nových technických indikátorů – obchodník může vytvořit nový technický indikátor založený na předdefinovaných indikátorech nebo použít svůj vlastní výpočet pro vznik úplně nového originálního indikátoru
- Nástroje pro risk-management – obě aplikace umožňují zadávat na trh ochranný Stop-Loss, a dokonce i nastavovat na cenových jakých úrovních se má případně posunout
- Uživatelské výstrahy (Custom alerts) – program, který si vytvoří uživatel, aby ho upozornil, že na trhu se aktuálně odehrává jistá pro něj zajímavá cenová akce (například rozdíl mezi otevírací a uzavírací cenou je větší než obchodník nastavil do tohoto programu). Obchodník tedy v případě této výstrahy ihned neobchoduje,

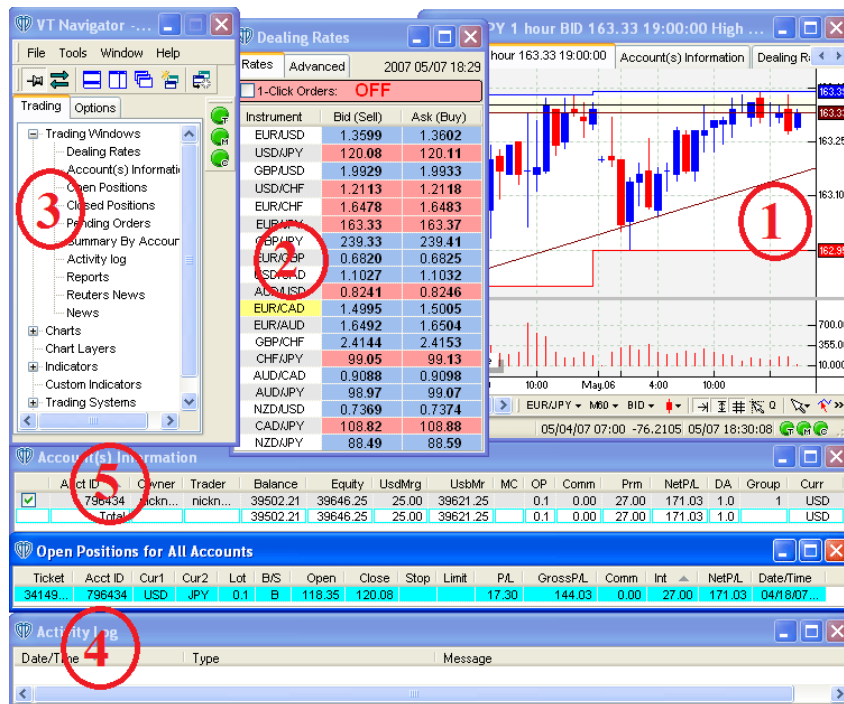
ale může se lépe a dříve soustředit na situaci na trhu a tak lépe reagovat.

- Podpora automatických obchodní strategie – nástroje pro mechanické obchodníky, kteří mohou vytvořit na základě zvolených pravidel program, který aplikace MetaTrader nebo VT Trader poté automaticky obchoduje.

Ze srovnání obou aplikací z mého pohledu lépe vyjde MetaTrader. Důvodem však není to, že by aplikace obsahovala lepší nástroje, ale fakt, že MetaTrader je mnohem více globálně využívaná platforma. S tím souvisí také lepší podpora aplikace a větší výběr brokerů, se kterými lze přes tuto platformu obchodovat. Nutno dodat, že obě aplikace jsou zdarma pro využití k technické analýze a u vybraných brokerů také pro obchodování (včetně historických dat pro účel backtestů, které jsou poskytovány skrz aplikaci brokerem).



Obrázek 5.8: Hlavní obrazovka aplikace MetaTrader 4. V pravé části obrazovky (oblast 1) jsou zobrazeny obchodníkem vybrané grafy technické analýzy. Panel Market Watch (oblast 2) obsahuje pohled na aktuální cenu brokerem zprostředkovaných instrumentů. Panel Navigator (oblast 3) je pohled na strom souborů aplikace MetaTrader 4, který obsahuje také uživatelem vytvořené programy – viz kapitola 6. Data Window panel (oblast 5) poskytuje informace o požadovaném časovém úseku (svíci na grafu). Terminal (oblast 5) se pak zaměřuje zejména na správu obchodníkovy účtu (vklady a výběry) a zobrazuje aktuálně otevřené i historické obchody. Zdroj: Metatrader 4



Obrázek 5.9: Hlavní obrazovka aplikace VT Trader (verze 2). Je patrné, že většina klíčových panelů je velice podobná programu MetaTrader 4 – graf technické analýzy (oblast 1), seznam poskytovaných instrumentů a cen (oblast 2), adresářový strom včetně uživatelových programů (oblast 3), Activity Log (oblast 4) – zobrazuje základní aktivity, které se od přihlášení udály na obchodníkově účtu a panel s informacemi o účtu (oblast 5) – zobrazuje stav účtu a otevřené obchody. Zdroj: VT Trader 2

6 MetaQuotes Language

Tato kapitola se zabývá jazykem MetaQuotes Language (MQL). V úvodu je popsána historie tohoto jazyka, základní vlastnosti jazyka, v dalších částech pak podrobněji některé nejdůležitější funkce a proměnné tohoto jazyka. Závěr kapitoly obsahuje praktickou ukázkou, jakých typů programových výstupů je možné za pomoci tohoto jazyka dosáhnout. Kapitola slouží především jako stěžejní teoretický základ pro následující praktickou část, jejímž cílem (respektive jedním z nich) je vytvořit funkční obchodní strategii právě v tomto jazyku.

6.1 Historie

Jazyk MetaQuotes Language je hlavním produktem společnosti MetaQuotes Software Corporation. Tato společnost byla založena v Rusku v roce 2000. Prvním velkým produktem, který firma uvedla na trh, byl program FX Charts, který umožňoval graficky analyzovat situaci na Forexu a posléze ji obchodovat. FX Charts byl ve světě tradingu jakousi revolucí, protože se jednalo o jeden z prvních produktů pro grafickou analýzu Forexu, určených pro malé obchodníky. O rok později společnost vydala vylepšenou verzi původního produktu- tento projekt nazvala MetaQuotes. MetaQuotes se odlišoval od FX Charts zejména rozšířeným spektrem trhů pro analýzu a obchodování (přidává se CFD) a novým prvkem- jazykem Meta Quotes Language. Tento jazyk již od počátku umožňoval tvorbu jednoduchých programů a strategií v prostředí Meta Quotes. V roce 2002 vychází druhá verze tohoto jazyka. Díky tomu, že je jazyk ve formě Open-source (tedy volně dostupný), zapojuje se do jeho vývoje čím dál více developerů a postupně vzniká čím dál obsáhlejší API. K tomu vzniká třetí verze MetaQuotes, která umožňuje obchodovat také futures trhy [Wikipedia.com, 2015].

V červenci 2005 přichází na trh aplikace MetaTrader 4. Od předchozích produktů se odlišovala vyšší stabilitou na straně serveru a celkově zvýšené spolupráci mezi klientem a serverem. Současně vzniká verze 4 jazyka MQL, která sice není zpětně kompatibilní s předchozími verzemi, ale zato výrazně rozšiřuje klientské API a nově dovoluje podporu DLL interface. Ke konci roku 2005 také vzniká mobilní klientská aplikace [Wikipedia.com, 2015].

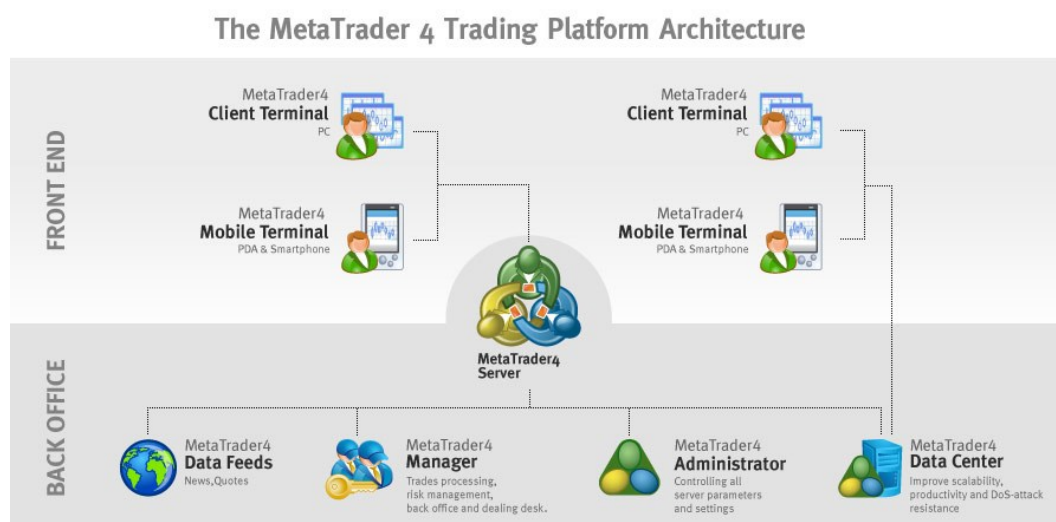
V červnu 2010 vzniká poslední pátá verze aplikace MetaTrader, spolu s jazykem MQL verze 5. Ačkoliv je samotná aplikace rozšířena o řadu funkcí a dovoluje obchodovat také akciové a derivátové trhy, jazyk MQL 5 není zpětně kompatibilní s verzí MQL 4. Tento problém je zásadní, protože před vznikem páté verze MetaTrader 5 a MQL se během

pěti let stala předchozí verze platformy velmi populární a nedostatečná kompatibilita způsobila, že v březnu 2011 na novou verzi platformy přešlo pouze 50 z celkových 450 brokerských společností nabízejících účet fungující na aplikaci MetaTrader [Wikipedia.com, 2015].

Právě nepříliš velká rozšířenost jazyka MQL 5 je důvodem, proč v této práci bude použita široce rozšířená verze MQL 4. K tomuto rozhodnutí přispěl také fakt, že verze MQL 4 byla na začátku roku 2015 (od buildu 600) rozšířena o většinu funkcí MQL 5 – včetně možností objektově orientovaného programování, která byla specifická pro verzi MQL 5 [MQL4, 2015].

6.2 Základní pojmy

MetaTrader 4 je aplikace určená k obchodování na finančních trzích, přičemž k usnadnění obchodování obsahuje nástroje technické analýzy. Funguje na principu architektury klient-server (viz Obrázek 6.1). Obchodník zadá příkaz k zobchodování (diskréční způsob obchodování) nebo vytvoří program v MetaQuotes Language, který zobchoduje situaci na trhu podle jím zvolených pravidel (mechanický způsob obchodování). Požadavek na zobchodování je předán klientskému terminálu, který jej odešle na server. Ten buď přijme nebo odmítne objednávku a odešle informaci o stavu objednávky zpět na klientský terminál, který zobrazí informaci obchodníkovi nebo předá data do programu [MQL4, 2015].



Obrázek 6.1: Architektura aplikace MetaTrader. Zdroj: <http://www.metaquotes.net/en/metatrader4>

MetaQuotes Language 4 je od buildu 600 vysokoúrovňovým objektově orientovaným programovacím jazykem, strukturou založeným na jazyku C++. Je určený pro tvorbu následujících typů programů [MQL4, 2015]:

- Automatický obchodní systém (neboli Expert Advisor (EA)) – je to automatický mechanický obchodní systém, který čeká na událost (signál) na grafu, po níž začne vykonávat zvolená pravidla – takovou událostí může být například změna cenové akce na grafu, časová událost, či jakékoliv jiné tvůrcem naprogramované pravidlo. Cílem praktické části práce je vytvořit právě takový Automatický obchodní systém.
- Technický indikátor (Custom Indicator) – prvek technické analýzy, který mechanicky pracuje se zvoleným grafem a interpretuje z něj obchodníkem požadované informace, na rozdíl od automatického obchodního systému není možné automatizovaně obchodovat jen pomocí indikátoru, ovšem automatické obchodní systémy typicky technické indikátory používají pro nalezení signálu pro vstup do obchodu
- Skript (Script) – slouží pro jednorázové spuštění zvolené akce – na rozdíl od EA, které jsou spouštěny obchodním terminálem při každé změně ceny
- Knihovna (Library) – jak platí ve světě programování, knihovna je určena pro uchování a distribuci často používaných bloků kódu

Pokud porovnáme jazyk MQL 4 s jeho vzorem C++, existují zde určité rozdíly a to zejména, že MQL 4 [MQL4, 2015]:

- Neobsahuje `goto` příkaz
- Nepodporuje deklaraci anonymního výčtového typu
- Nepodporuje vícenásobnou dědičnost

Aplikace MetaTrader obsahuje pro účely tvorby kódu v MQL vestavěné vývojové prostředí (IDE) – MetaEditor. Toto IDE obsahuje pro usnadnění práce programátora několik užitečných nástrojů [MQL4, 2015]:

- MQL Wizard – komponenta, která umožňuje i obchodníkovi, který nezná konstrukci jazyka MQL, vytvořit spustitelný program (obchodní systém). Obchodník musí zvolit pouze pravidla pro vstup a výstup z obchodu – zbytek kódu je tímto nástrojem vygenerován.
- Překladač – překládá MQL do nízko-úrovňového jazyka a umožňuje tak program spustit

- Debugger – nástroj, který se umožňuje vnořit do kódu a nalézt v něm chybu

Pro maximální podporu mechanického způsobu práce s obchodní strategií obsahuje aplikace Metatrader také komponentu Tester, která umožňuje otestovat strategii vytvořenou v jazyku MQL4 na historických datech zvoleného instrumentu. Výsledný report testování pak obsahuje základní přehled výkonnosti strategie na zvoleném historickém období (podle Obrázku 5.7), výsledky jednotlivých obchodů nasimulovaných podle pravidel strategie na historických datech a graf vývoje stavu testovacího účtu po exekuci nasimulovaných obchodů v čase a deník, který obsahuje technické podrobnosti o exekuci jednotlivých obchodů [MQL4, 2015].

6.3 Struktura a syntaxe jazyka

6.3.1 Proměnné v jazyku MQL 4

Proměnná je v každém programovacím jazykem nevyhnutelným nástrojem pro uložení mezi-výpočtů. Na rozdíl od konstanty se uchovávaná hodnota může měnit, proměnná tedy často bývá pouze dočasným úložištěm informace.

6.3.1.1 Uživatelské proměnné

Jazyk MQL umožňuje uživateli deklarovat a používat vlastní proměnné. Podle rozsahu použití, pak můžeme tyto proměnné rozdělit dále do dvou skupin [MQL4, 2015].

- Lokální proměnné – deklaruje se uvnitř funkce a lze ji tedy použít pouze uvnitř této funkce. Jakmile je běh funkce ukončen, tak proměnná zaniká
- Globální proměnné – proměnná je deklarována nad všemi funkcemi a je tedy dostupná v rámci celého programu. Ukončením programu proměnná zaniká.

Lokální proměnnou je navíc možné rozšířit pomocí klíčového slova `static` na statickou proměnnou – proměnná se pak použije v těle funkce, s tím rozdílem, že statická proměnná nezanikne po tom, co je funkce ukončena. Proměnná se tak inicializuje při prvním volání funkce a poté zůstává alokovaná v paměti i při dalším volání funkce [MQL4, 2015].

Dále je možné využít tzv. externí proměnnou (klíčové slovo `extern`), což je globální proměnná, do níž se ukládají vstupní nastavení pro daný program. Hodnota externí

proměnné se nastavuje v dialogovém okně platformy Metatrader při spuštění programu [MQL4, 2015].

6.3.1.2 Základní předdefinované proměnné

Předdefinované proměnné představují jeden ze základních prvků jazyka MQL 4 - umožňují interpretovat aktuální stav trhu. Jsou to read-only (pouze ke čtení, ne k ukládání) hodnoty, které uživateli významným způsobem urychlují práci. Proměnné se pak vztahují vždy k aktuálnímu měnovému páru, pro který se program používá. Proměnné lze rozdělit do dvou skupin podle datového typu, ve kterém se zobrazují [MQL4, 2015]:

- Proměnné primitivního datového typu – základní proměnné:
 - Ask – poptávková cena
 - Bid – nabídková cena
 - Bars – počet svíček ve zvoleném období
 - Point – hodnota bodu ve zvolené měně (podle zvoleného instrumentu)
 - Digits – počet desetinných míst zvoleného instrumentu

- Proměnné abstraktního datového typu Array (pole) – v těchto proměnných je uchována informace o svíčkách grafu (nejaktuálnější svíčka na grafu, tedy ta, která je napravo, je indexována číslem 0, každá starší na grafu má pak index o jednotku vyšší). Uchovávají se tyto hodnoty:
 - Time – čas
 - Open – otevírací cena svíčky
 - Close – uzavírací cena svíčky
 - High – nejvyšší cena svíčky
 - Low – nejnižší cena svíčky
 - Volume – počet změn cenové úrovně během trvání vykreslování dané svíčky

6.3.2 Datové typy v jazyku MQL 4

Do proměnné lze uložit jakoukoliv hodnotu, která odpovídá oboru hodnot daného datového typu. Nejpoužívanější primitivní datové typy jazyka MQL 4 jsou popsány v Tabulce 6.1 [MQL4, 2015]:

Tabulka 6.1: Základní primitivní datové typy. Zdroj: <http://docs.mql4.com>

| Datový typ | Obor hodnot | Velikost | Rozsah hodnot |
|------------|-----------------|----------|---------------------------------|
| bool | logická hodnota | 4 bajty | - |
| int | celé číslo | 4 bajty | -2 147 483 648 až 2 147 483 648 |
| double | reálné číslo | 8 bajtů | -1.7e – 308 až 1 .7e + 308 |
| string | textový řetězec | 8 bajtů | - |
| color | barva | 4 bajty | 0 až 255 |
| datetime | datum | 4 bajty | 1.1 1970 až 31.12.2037 |

Hodnoty typu `color` lze reprezentovat třemi způsoby – číselně, slovně nebo literálem. Při vyjádření literálem je hodnota složena ze tří částí, kde každá část vyjadřuje číselně intenzitu jedné z barev modelu RGB. Hodnota se uvádí prefixem `C`, za níž následuje trojice čísel uzavřená v jednoduchých uvozovkách. Číselný způsob reprezentace je podobný, ale barva je kódována jedním číslem decimálně nebo hexadecimálně (např. `0xRRGGBB`, kde jednotlivá písmena představují číslice, vyjadřující intenzitu daných barev). Slovně lze pak barvu vyjádřit přímo jejím názvem [MQL4, 2015].

Proměnná typu `date` je složena ze šesti částí – rok, měsíc, den, hodinu, minuty a sekundy. Na počátku se uvádí prefix `D`, za kterým je obsažen časový údaj uzavřený v jednoduchých uvozovkách [MQL4, 2015].

6.3.3 Operátory a funkce

Operátory jazyka MQL vychází z jazyka C++ a mezi operátory těchto dvou jazyků se nevyskytují žádné pro tuto práci významné rozdíly.

Dá se tvrdit, že funkce jsou základním stavebním prvkem programovacího jazyka. Každá funkce je jednoznačně určena svou definicí a deklarací. Funkce má dva základní významy:

- Umožňuje rozdělit funkčnost celého programu do logických bloků
- Umožňuje parametrizovat algoritmus

6.3.3.1 Uživatelské funkce

Každá uživatelská funkce se skládá z hlavičky (deklarace) a těla (definice). Návrátový typ je určen v deklaraci (pokud funkce nemá za úkol navracet hodnotu, poté je návratový typ `void`). Deklarace dále obsahuje unikátní název funkce a seznam parametrů. Definice

funkce je poté obsažena v složených závorkách a zakončena příkazem `return` (pokud se nejedná o návratový typ `void`) [MQL4, 2015].

6.3.3.2 „Special functions“

Jedná se o zvláštní skupinu funkcí, specifickou pro MQL 4. Speciální funkce jsou volány klientským terminálem – nezahrnují se tedy explicitně do kódu. Jedná se o tyto 3 funkce [MQL4, 2015]:

- `int init()` – je vykonaná pouze jedenkrát při inicializaci programu
- `int start()` – je vykonaná n-krát při každém ticku (neboli aktualizaci cenové úrovně daného instrumentu)
- `int deinit()` – je vykonaná pouze jedenkrát při deinicializaci programu

6.3.3.3 Předdefinované funkce

Podobně jako předdefinované proměnné, také předdefinované funkce napomáhají ke snazšímu vývoji programu. Nejčastěji používané funkce pak jsou atomizované a standardizované [MQL4, 2015].

Funkce pro přístup k datům na grafu

Předdefinované funkce vychází z předdefinovaných proměnných. Ty jsou vázány k příslušnému měnovému páru. Funkce pro přístup k datům na trh tedy umožňují získat data z těchto proměnných pro příslušný pár a časovou periodu. Patří mezi ně například:

- `iBars()`
- `iClose()`
- `iHigh()`
- `iHighest()` – vrací nejvyšší hodnotu ceny v rámci zvolených svíček
- `iLow()`
- `iLowest()` – vrací nejnižší hodnotu ceny v rámci zvolených svíček
- `iOpen()`
- `iTime()`
- `iVolume()`

MQL4 umožňuje získat přístup nejen k těmto základním datům, vázajícím se k ceně instrumentu, ale také k datům defaultních indikátorů – tedy také ke klouzavým průměrům,

Bollinger bands a ATR, které byly představeny v Kapitole 5. Funkce pro tyto příslušné indikátory mají následující podobu a parametry [MQL4, 2015]:

- `double iMA (string symbol, int timeframe, int ma_period, int ma_shift, int ma_method, int applied_price, int shift)`
 - parametr „symbol“ představuje instrument, na kterém chceme indikátor aplikovat; pro použití stejného instrumentu jako ve zbytku strategie, v níž má být indikátor použit se jako parametr dává výstup funkce `Symbol()`
 - parametr „timeframe“ vyjadřuje timeframe- velikost časových úseků, pro které se má indikátor zobrazovat; hodnota 0 znamená použití stejného timeframe jako ve zbytku strategie
 - parametr „ma_period“ znamená periodu n , se kterou se má klouzavý průměr aplikovat
 - „ma_shift“ umožňuje posun křivky indikátoru na grafu (nebude v této práci využito)
 - parametr „ma_method“ vyjadřuje typ klouzavého průměru; 1- prostý klouzavý průměr, 0 – exponenciální klouzavý průměr
 - „applied_price“ představuje cenovou úroveň svíček, pro něž se indikátor počítá (nejčastěji to bývá hodnota `PRICE_CLOSE`)
 - parametr „shift“ vyjadřuje pozici konkrétního prvku indikátoru, který má být vybrán; hodnota 0 znamená poslední (nejnovější) prvek indikátoru, 1 je druhý nejnovější (druhý zprava) prvek na grafu, atd.

- `double iBands(string symbol, int timeframe, int period, double deviation, int band_shift, int applied_price, int mode, int shift)`
 - parametr „symbol“ se stejným významem jako u předchozí funkce
 - parametr „timeframe“ se stejným významem jako u předchozí funkce
 - „period“ znamená periodu n , se kterou se má Bollinger bands aplikovat
 - parametr „deviation“ znamená směrodatnou odchylku horní a spodní křivky od prostřední křivky (klouzavého průměru)
 - „band_shift“ umožňuje posun křivky indikátoru na grafu (nebude v této práci využito)
 - parametr „applied_price“ představuje cenovou úroveň svíček, pro něž se indikátor počítá (nejčastěji to bývá hodnota `PRICE_CLOSE`)
 - int „mode“ vyjadřuje výběr jedné ze tří křivek indikátoru: 0 - `MODE_MAIN`, 1 - `MODE_UPPER`, 2 - `MODE_LOWER`
 - parametr „shift“ vyjadřuje pozici konkrétního prvku indikátoru, který má být vybrán; hodnota 0 znamená poslední (nejnovější) prvek indikátoru, 1 je druhý nejnovější (druhý zprava) prvek na grafu, atd.

- `double iATR(string symbol, int timeframe, int period, int shift)`
 - parametr „symbol“ se stejným významem jako u předchozí funkce
 - parametr „timeframe“ se stejným významem jako u předchozí funkce
 - „period“ znamená periodu n , se kterou se má ATR aplikovat
 - parametr „shift“ se stejným významem jako u předchozí funkce

Funkce pro řízení průběhu programu

Mezi tyto funkce můžeme zařadit zejména funkce pro podporu debugingu nebo vytváření logů [MQL4, 2015]:

- `Print(argument, next argument)` - slouží pro výpis zvolených argumentů do logu aplikace
- `Comment()` - slouží pro výpis přímo do příslušného grafu
- `Alert()` - slouží pro přímou interakci s obchodníkem pomocí dialogového okna

Funkce pro práci s okny

- `WindowFind()` – každému zobrazenému grafu v platformě je přiřazené okno – tato funkce slouží k zprehlednění práce s mnoha okny, kdy se jako parametr zadává jméno okna a funkce nalezne jeho číselný index
- `WindowRedraw()` – používá se, pokud je potřeba překreslit graf v okně (z důvodu změny vlastností objektů na grafu)
- `WindowScreenShot()` – uloží snímek obrazovky aktuálního grafu do souboru
- `RefreshRates()` – obnoví hodnoty předdefinovaných proměnných [MQL4, 2015].

Funkce pro obchodování

Tyto funkce jsou určeny výhradně pro podporu automatického obchodování. Mezi nejdůležitější patří [MQL4, 2015]:

- `int OrderSend(string symbol, int cmd, double volume, double price, int slippage, double stoploss, double takeprofit, string comment, int magic, datetime expiration, color arrow_color)` – odesílá objednávku z klientského

terminálu na server, kde čeká na vyplnění při splnění pravidel definovaných obchodníkem

- parametr „symbol“ určuje zobchodovaný symbol
 - „cmd“ vyjadřuje typ obchodu – v této práci OP_BUY či OP_SELL
 - „volume“ představuje počet zobchodovaných lotů
 - „price“ představuje cenu při níž chceme nakupovat/prodávat
 - „slippage“ je maximální hodnota spreadu, kterou chceme akceptovat
 - „stoploss“ je cenová úroveň Stop-lossu
 - „profittarget“ je cenová úroveň Profit-targetu
 - „comment“ představuje poznámku k obchodu
 - „magic“ je identifikátor obchodního příkazu
 - „expiration“ doba, po níž objednávka vyprší (platí pro typy objednávky, které nejsou užívány v rámci této práce)
 - „arrow_color“ vyjadřuje barvu označení obchodu na grafu
- `bool OrderClose(int ticket, double lots, double price, int slippage, color arrow_color)` – odesílá požadavek na uzavření otevřeného obchodu
 - „ticket“ představuje unikátní číslo již realizovaného obchodu na obchodním serveru
 - „lots“ vyjadřuje počet lotů k uzavření
 - „price“ je cena, při níž chceme obchod uzavřít
 - „slippage“ je maximální hodnota spreadu, kterou chceme akceptovat
 - „arrow_color“ vyjadřuje barvu označení obchodu na grafu
- `bool OrderModify(int ticket, double price, double stoploss, double takeprofit, datetime expiration, color arrow_color)` – odesílá požadavek na úpravu čekající objednávky
 - „ticket“ představuje unikátní číslo již realizovaného obchodu na obchodním serveru
 - „price“ upravená vstupní cena (platí pro typy obchodů, které nejsou v této práci využívány)
 - „stoploss“ je upravená cenová úroveň Stop-lossu
 - „profittarget“ je upravená cenová úroveň Profit-targetu
 - „expiration“ doba, po níž objednávka vyprší (platí pro typy objednávky, které nejsou užívány v rámci této práce)
 - „arrow_color“ vyjadřuje barvu označení obchodu na grafu

- `bool OrderSelect(int index, int select, int pool)` – namapuje existující objednávku do cache (podle zvoleného parametru), kde s ní lze dále pracovat (volat její funkce)
 - „index“ je identifikátor objednávky, určený podle hodnoty druhého parametru této funkce
 - „select“ vyjadřuje z jaké množiny se má vybírat hodnota parametru „index“ – `SELECT_BY_POS` vyjadřuje výběr obchodu podle hodnoty indexu ze seznamu obchodů uskutečněných obchodníkem; hodnota `SELECT_BY_TICKET` vyjadřuje výběr podle čísla (indexu) ticketu, který se uchovává na obchodním serveru
 - „pool“ – tento parametr se používá, pokud druhý má druhý parametr hodnotu `SELECT_BY_POS`. Může nabývat dvou hodnot – `MODE_TRADES` představuje výběr obchodu podle indexu z aktuálně otevřených obchodů; `MODE_HISTORY` představuje výběr obchodu podle indexu z již uzavřených nebo zrušených obchodů

- `OrderLots()` – poskytuje informaci o počtu otevřených lotů objednávky
- `OrderDelete()` – odesílá požadavek na odstranění čekající objednávky
- `OrderOpenPrice()` – vrátí informaci o úrovni ceny při otevření obchodu
- `OrderOpenTime()` – vrátí informaci o čase otevření obchodu
- `OrderClosePrice()` – vrátí informaci o úrovni ceny při uzavření (ukončení) obchodu
- `OrderCloseTime()` – vrátí informaci o čase ukončení obchodu

Zpracování chyb

MQL4 také podporuje debugging a kontrolu chyb. V zásobníku lze nalézt pouze kód poslední chyby, který lze zobrazit do logu terminálu pomocí funkce `GetLastError()`. K tomuto kódu pak lze dohledat popis konkrétní chyby v knihovně `stderror.mqh` [MQL4, 2015].

6.3.4 Průběh vykonávání programu

Program vytvořený v jazyku MQL4 je zpravidla rozdělený do těchto částí [MQL4, 2015]:

- Hlavička programu (obsahuje odkaz na zahrnuté knihovny, deklarace globálních proměnných a jejich inicializace)
- Část se speciálními funkcemi (`start()`, `init()`, `deinit()`), které volají uživatelské funkce

- Blok s uživatelskými funkcemi

Po spuštění programu v konkrétním grafu je běh kroků následující [MQL4, 2015]:

1. Spustí se kód hlavičky
2. Spustí se funkce `init()` – tato funkce je volána terminálem pouze jednou, musí proto obsahovat kód nutný pro start programu – např. inicializační výpočty
3. Vykona se funkce `start()` – ta povětšinou obsahuje kód pro vykonání nějaké činnosti (typicky manipulace s objednávkou) a může být závislá na hodnotách předdefinovaných proměnných. Po ukončení této funkce je řízení navraceno zpět terminálu, jakmile však dojde k vykreslení nového ticku, řízení programu je opět předáno z terminálu na funkci `start()`.
4. Po dokončení běhu programu je terminálem zavolána funkce `deinit()`, které je odevzdáno řízení běhu programu – tato funkce se musí volat kvůli validnímu ukončení celého programu (může také obsahovat funkce pro uzavření a odstranění objednávky)
5. Pokud dojde k volání uživatelské funkce, poté se řízení předá z volající funkce volané uživatelské funkci a po skončení uživatelské funkce je řízení programu opět předáno funkci, která uživatelskou funkci volala

7 Tvorba obchodní strategie

Tato kapitola je součástí praktické části této práce a obsahuje analýzu problémové oblasti a návrh řešení. Kapitola se zabývá tvorbou obchodní strategie, přičemž dodržuje strukturu životního cyklu obchodní strategie definovanou v kapitole číslo 3. Proces vytváření strategie pokrývá tento definovaný cyklus až do bodu 4, tedy strategie bude detailně otestována. Obchodní strategie bude vytvářena s pomocí diskréční i mechanické analýzy trhu. Diskréční analýza má za úkol pomoci při hrubém testování strategie a návrhu používaných indikátorů. Ve fázi tvorby detailních pravidel a detailního testování výkonnosti strategie bude však použit mechanický způsob práce se strategií. Cílem je, aby vzniklá strategie byla plně mechanicky obchodovatelná na zvoleném trhu.

7.1 Zvážení rizika a kapitálu

Jelikož celý životní cyklus strategie bude probíhat na demo účtu, na kterém lze pracovat neomezeně s fiktivním kapitálem, není potřeba kapitálu problémem, který by bylo v rámci této práce potřeba nadměrně řešit. Přesto se však v rámci tvorby strategie pokusím dodržovat základní pravidla money-managementu a risk-managementu definované v předchozích kapitolách této práce.

Hlavní otázkou k řešení v této podkapitole tak bude především výběr brokera, jelikož je to jeden z faktorů, který má na řízení kapitálu přímý dopad. Z Tabulky 4.1 jsem vybral 3 brokery s nejnižšími maximálními spready, u nichž budu srovnávat další parametry. Těmito brokery jsou:

- Liquid Markets
- Alpari
- Ikon

V dalším kroku bylo potřeba ověřit, zda tito brokeri nabízejí obchodní platformu MetaTrader4, s níž chceme při tvorbě, testování a zkušebním obchodování strategie pracovat, což se potvrdilo.

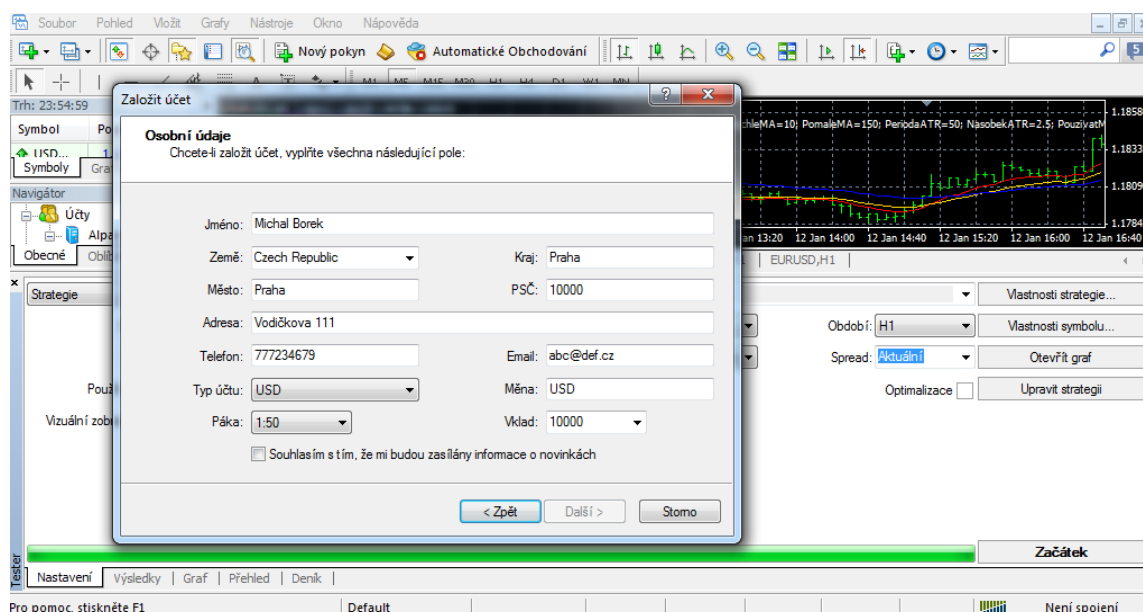
Velmi důležitá je otázka regulace brokera orgány, které drží dohled nad finančním trhem. Zatímco brokerská společnost Ikon sídlí v Hong-Kongu a její evropská společnost

IKON Capital Limited je regulována regulačním úřadem Financial Conduct Authority (FCA) a společnost Alpari organizací Center for Regulating OTC Financial Instruments and Technologies (CRFIN), která byla vytvořena samotnými Forex brokery, pro jejich vlastní organizaci a regulaci, společnost Liquid Markets měla v době psaní této práce s regulací problémy, jelikož nemohla dostávat svým závazkům a poskytovat služby po aktu Švýcarské národní banky, která v lednu 2015 zrušila minimální hladinu kurzu EUR/CHF. Z tohoto důvodu bych společnost Liquid Markets rozhodně jako brokera pro obchodování strategie (v době psaní této práce) nevolil.

Jak bylo zmíněno v teoretické části práce, velmi důležitým bodem v hodnocení brokera je míra a způsob podpory zákazníka. Z tohoto srovnání vychází výrazně lépe společnost Alpari, jelikož poskytuje plnohodnotný kvalitní servis zákaznické podpory – nepřetržitou telefonickou podporu, možnost kontaktování e-mailem a v době pracovních dnů také on-line chat podporu (v anglickém jazyce). Naproti tomu společnost Ikon nenabízí chat podporu v anglickém jazyce (nabízí pouze v čínštině), což vnímám jako obrovský nedostatek v případě, že bych se rozhodl s tímto brokerem obchodovat na reálném účtu.

Jelikož možnosti pákového efektu a obchodovatelné instrumenty jsou u obou společností prakticky totožné (obě společnosti nabízí veškeré Major i Minor instrumenty) a také vzhledem k tomu, že společnost Alpari na FOREX trhu figuruje téměř 17 let, rozhodl jsem se pro účel této práce vybrat právě ji.

Pro účely této práce byl tedy založen fiktivní demo-účet u společnosti Alpari o velikosti 10 000 USD, což je minimální částka doporučovaná v praxi pro seriózní obchodování (viz Podkapitola 5.1). Pákový efekt byl zvolen na 2% (viz Obrázek 7.1).



Obrázek 7.1: Zakládání nového demo-úctu v aplikaci Alpari-Demo (modifikovaná aplikace)

MetaTrader4). Je možné zvolit velikost finanční páky a fiktivního vkladu. Zdroj: Metatrader 4

7.2 Volba nástrojů a způsob analýzy

Tato podkapitola slouží pro určení jakéhosi rámce, pod kterým se budou nalézat konkrétní pravidla pro obchodování. Jedná se především o určení způsobu analýzy trhu a obchodování a velikosti časového úseku (timeframe), se kterým chceme pracovat.

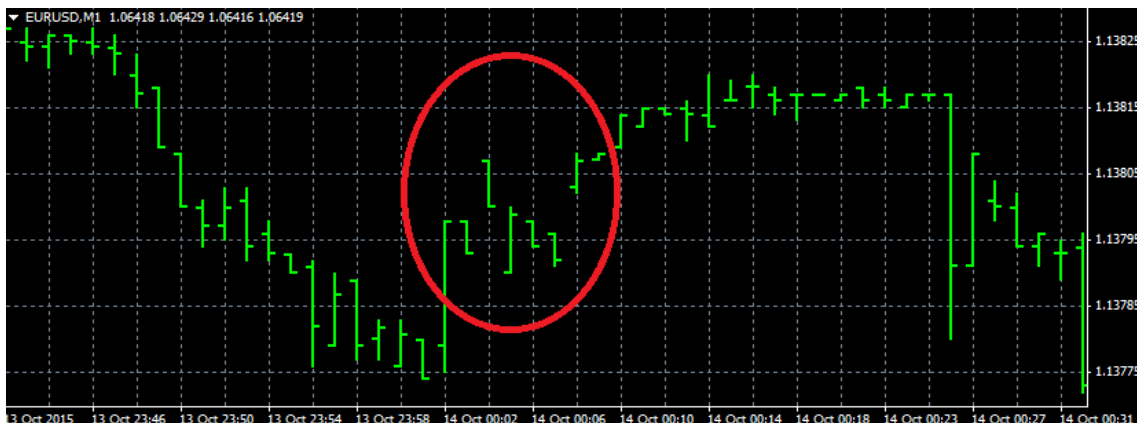
Při volbě analýzy trhu bychom měli rozlišit, zda využijeme fundamentální nebo technickou analýzu. Jelikož cílem této práce je vytvořit strategii v jazyku MQL, tedy pro aplikaci MetaTrader, která ve své podstatě nepodporuje fundamentální analýzu, je zřejmé, že se při hledání obchodní strategie budu soustředit plně na technickou analýzu.

Co se týká volby způsobu obchodování, je nutné zvolit, zda obchodovat (nebo také testovat na historických datech) strategii diskřétně nebo mechanicky. Cílem práce je vytvořit strategii, kterou bude možné obchodovat a testovat mechanicky, tedy bude možné ji převést do programu (počítačového kódu). To však neznamená, že nebude použit diskřétní způsob práce se strategií. Diskřétní analýzu trhu bude potřeba použít pro vyfiltrování vhodného souboru pravidel strategie, který má dále smysl převádět do mechanické podoby k testování a obchodování.

Co se týká volby timeframe, je nutné si uvědomit, že je vhodné zvolit takový timeframe, aby pro daný rozsah historických dat, který máme od brokera Alpari k dispozici (v době psaní této práce byla k dispozici data od 1.1.2014), bylo možné vytvořit seriózní vzorek dat (tedy v podstatě svíček) pro otestování. Proto je nesmyslné pro daný rozsah dat například volit měsíční či týdenní timeframe, pro který bychom měli k dispozici velmi malý vzorek dat (23 (měsíčních), respektive 96 (týdenních) svíček). Na druhou stranu je nebezpečné se soustředit také na příliš nízký timeframe – především minutový. Takový timeframe může obsahovat velké množství zkreslených dat, z důvodu krátkodobých výkyvů na finančních trzích, které broker sice zobrazí, ale při reálném obchodování nejsou obchodovatelné (viz Obrázek 7.2). Proto při sestavování pravidel strategie volím mezi timeframy:

- 15 min
- 30 min
- 1 h

Pro tyto timeframy je v daném časovém období poskytnut seriózní vzorek dat a případné výkyvy kurzů na finančních trzích se na výsledky testování a případného obchodování nepromítnou takovou mírou, jako u nižších timeframů.



Obrázek 7.2: Příklad výkyvu ceny (tzv. gapu), který může nastat na nízkém (obvykle 1-5 minutovém) timeframu. Cenový pohyb může být natolik rychlý, že na grafu vzniknou prázdné mezery mezi časovými úseky, popřípadě nemusí být ani pohyb ceny zobrazen brokerem validně. Zdroj: Metatrader 4

Celý proces tvorby obchodní strategie pak bude probíhat v aplikaci MetaTrader 4, jelikož tato aplikace plně podporuje jazyk MQL4 a cílem této práce je vytvořit strategii právě v tomto jazyce.

7.3 Porovnávání obchodních přístupů

Hlavním úkolem této fáze je porovnat mezi sebou různé obchodní přístupy – zejména prvky technické analýzy, vybrat z nich pro obchodníka ten, který je pro něj nejvhodnější tak, aby v další fázi bylo možné mít k dispozici kompletní a jednoznačná pravidla a detailně otestovat je na vybraném trhu se zvoleným nastavením.

V této fázi tedy byly porovnány instrumenty s časovými rámci podle Tabulky 7.1:

Tabulka 7.1: Tabulka variant obchodních přístupů. Zdroj: Autor

| | | |
|-----------------------------|-----------------------------|-----------------------------|
| EUR/USD, timeframe 15 minut | EUR/USD, timeframe 30 minut | EUR/USD, timeframe 1 hodina |
| GBP/USD, timeframe 15 minut | GBP/USD, timeframe 30 minut | GBP/USD, timeframe 1 hodina |

| | | |
|-----------------------------|-----------------------------|-----------------------------|
| EUR/GBP, timeframe 15 minut | EUR/GBP, timeframe 30 minut | EUR/GBP, timeframe 1 hodina |
|-----------------------------|-----------------------------|-----------------------------|

Na každou kombinaci trhu a timeframe z této tabulky budou aplikovány dva indikátory technické analýzy, které byly popsány v podkapitole 5.1.1, tedy:

- Křížení dvou klouzavých průměrů
 - perioda n rychlého klouzavého průměru bude zvolena 10
 - perioda n pomalého klouzavého průměru bude zvolena 150
- Zúžení Bollinger bands (tedy protnutí všech křivek Bollinger bands celou jednou svíčkou a pokračování cenové akce ve směru jedné z křivek podle Obrázku 5.4)
 - perioda indikátoru n bude zvolena 50
 - standardní odchylka pro posun horní a spodní křivky k bude zvolena podle defaultního nastavení indikátoru, tedy 2

Stop-Loss a také Profit-Target budou zvoleny v této fázi fixně, tedy bez použití adaptabilního indikátoru ATR. Obě úrovně budou nastaveny v závislosti na timeframu trhu, na němž se aplikují a sice:

- timeframe 15 minut – Stop-Loss a Profit-Target: 300 pips
- timeframe 30 minut – Stop-Loss a Profit-Target: 600 pips
- timeframe 1 hodina – Stop-Loss a Profit-Target: 1000 pips

Velikost těchto hodnot byla zvolena diskrečně a odpovídá přibližně trojnásobku průměrného rozpětí svíčky na daných trzích a zvoleném timeframe za období říjen 2015. Smyslem je porovnat zejména chování indikátorů pro vstupní signál na určených kombinacích instrumentu a timeframe, nikoliv úrovní stop-lossu a profit-targetů, které budou detailně specifikovány v další fázi životního cyklu strategie.

Jelikož pravidla hrubého návrhu variant strategie byla velmi striktně určena, bylo možné vytvořit jednoduchý program v jazyku MQL4 a vytvořit tak mechanickou podobu variant strategie. To mělo obrovský přínos nejen pro rychlost testování kombinací na historických datech, ale také na přesnost testů.

Základní nákupní vstupní podmínka (celý kód je zobrazen v příloze A) při zúžení Bollinger bands, převedená do kódu programovacího jazyka MQL4 má podobu:

```
if(
    VsechnyObchody_buy == 0 && iLow (Symbol(), 0, 2) < spodni && iHigh
(Symbol(), 0, 2) > horni && iHigh (Symbol(), 0, 1) > aktualni_horni &&
iLow (Symbol(), 0, 2) > aktualni_spodni)
{
    int ticket = OrderSend(Symbol(), OP_BUY, 0.1, Ask, SkluzVPlneni,
0, 0, "", 12345, 0, Blue);
    OrderSelect(ticket, SELECT_BY_TICKET, MODE_TRADES);
    OrderModify(ticket, OrderOpenPrice(), Ask - 0.00300 , 0 , 0,
Green);
    OrderModify(ticket, OrderOpenPrice(), OrderStopLoss (), Ask +
0.00300 , 0, Green);
}
```

Instrument se tedy nakoupí, pokud není otevřen žádný nákupní příkaz, Low a High svíčky dva časové úseky nazpět od současné překrývají celý kanál Bollinger bands a svíčka, která je jeden časový úsek zpět se dotýká už pouze horní křivky Bollinger bands. Jsou-li tedy splněny tyto podmínky, nákup instrumentu se uskuteční na úrovni Open aktuálního časového úseku. Objednávce je také okamžitě nastavený Stop-Loss a Profit-Target – v tomto případě (jelikož prezentovaný kód je určený pro 15 minutový timeframe) na úrovni 300 pipů od aktuální vstupní ceny.

Stejně tak instrument se prodá na úrovni Open aktuálního časového úseku, pokud není otevřen žádný prodejní příkaz, Low a High svíčky dva časové úseky nazpět od současné překrývají celý kanál Bollinger bands a svíčka, která je jeden časový úsek zpět oproti aktuální, se dotýká už pouze spodní křivky Bollinger bands. Pro nastavení Stop-Lossu a Profit-Targetu platí stejná pravidla jako v případě nákupního signálu (pouze v opačných směrech).

```
if(
    VsechnyObchody_sell == 0 && iLow (Symbol(), 0, 2) < spodni && iHigh
(Symbol(), 0, 2) > horni && iHigh (Symbol(), 0, 1) < aktualni_horni &&
iLow (Symbol(), 0, 2) < aktualni_spodni)
{
```

```

int ticketb = OrderSend(Symbol(), OP_SELL, 0.1, Bid, SkluzVPlneni,
0, 0, "", 12345, 0, Red);

OrderSelect(ticketb, SELECT_BY_TICKET, MODE_TRADES);

OrderModify(ticketb, OrderOpenPrice(), Bid + 0.00300 , 0 , 0,
Green);

OrderModify(ticketb, OrderOpenPrice(), OrderStopLoss (), Bid -
0.00300 , 0, Green);
}

```

Nákupní vstupní podmínka (celý kód je opět zobrazen v příloze B) při překřížení klouzavých průměrů, v mechanické podobě kódu programovacího jazyka MQL4 vypadá následovně:

```

if(
    VsechnyObchody_buy == 0 && fast_ma > slow_ma && fast_ma1 <
slow_ma1)
{

    int ticket = OrderSend(Symbol(), OP_BUY, 0.1, Ask,
    SkluzVPlneni, 0, 0, "", 12345, 0, Blue);

    OrderSelect(ticket, SELECT_BY_TICKET, MODE_TRADES);

    OrderModify(ticket, OrderOpenPrice(), Ask - 0.00300 , 0 , 0,
Green);

    OrderModify(ticket, OrderOpenPrice(), OrderStopLoss (), Ask +
0.00300 , 0, Green);
}

```

Instrument se tedy nakoupí, pokud není otevřen žádný nákupní příkaz, dva časové úseky nazpět oproti aktuální svíčce se rychlý klouzavý průměr na grafu nachází ještě pod pomalým klouzavým průměrem a jeden časový úsek nazpět oproti aktuální svíčce se už rychlý klouzavý průměr nachází nad pomalým klouzavým průměrem. Jsou-li tedy splněny tyto podmínky, nákup instrumentu se uskuteční na úrovni Open aktuálního časového úseku. Objednávce je také okamžitě nastavený Stop-Loss a Profit-Target – v tomto případě (jelikož příklad kódu je určený pro 15 minutový timeframe) na úrovni 300 pipů od aktuální vstupní ceny.

Stejně tak instrument se prodá na úrovni Open aktuálního časového úseku, pokud není otevřen žádný prodejní příkaz, dva časové úseky nazpět oproti aktuální svíčce se rychlý klouzavý průměr na grafu nachází ještě nad pomalým klouzavým průměrem a jeden časový úsek oproti aktuální svíčce nazpět se už rychlý klouzavý průměr nachází pod pomalým klouzavým průměrem. Pro nastavení Stop-Lossu a Profit-Targetu platí stejná pravidla jako v případě nákupního signálu (pouze v opačných směrech).

```
if (
    VsechnyObchody_sell == 0 && fast_ma < slow_ma && fast_ma1 >
    slow_ma1)
{
    int ticketb = OrderSend(Symbol(), OP_SELL, 0.1, Bid,
        SkluzVPlneni, 0, 0, "", 12345, 0, Red);
    OrderSelect(ticketb, SELECT_BY_TICKET, MODE_TRADES);
    OrderModify(ticketb, OrderOpenPrice(), Bid + 0.00300 , 0 , 0,
        Green);
    OrderModify(ticketb, OrderOpenPrice(), OrderStopLoss (), Bid -
        0.00300 , 0, Green);
}
```

Po vytvoření kódu obou variant strategie pro všech 9 variant kombinací trhu a timeframe bylo otestováno celkem 18 variant možné strategie (Tabulka 7.2) na historickém období 1.1.2015 - 1.11.2015.

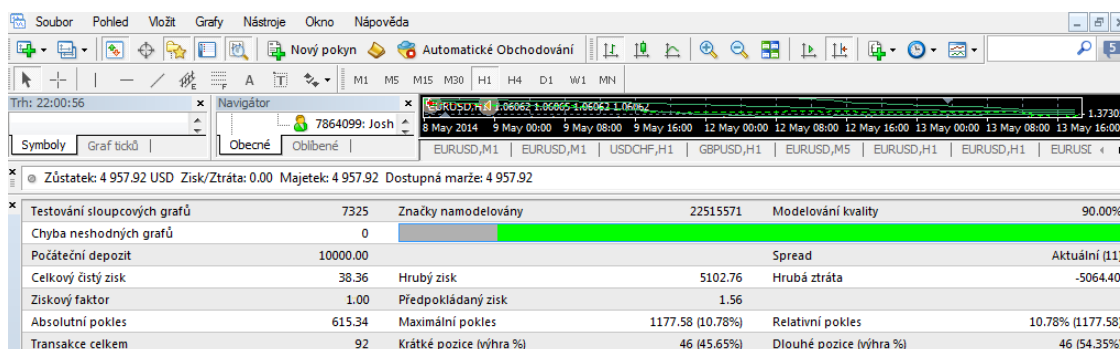
Z výsledků je patrné, že největší potenciál pro detailní otestování má varianta aplikace křížení klouzavých průměrů na trhu EUR/USD na hodinovém timeframe, která jako jediná dosáhla na zvoleném období zisku (Obrázek 7.3). Je nutné podotknout, že nejenom že balance zisků a ztrát byla ve výsledku kladná, ale do výsledků variant strategie jsou také započítány poplatky (spread) brokera (v průměru 3-4 pips). To znamená, že pokud by se tyto poplatky nebraly v potaz, mohl by být zisk strategie na takto krátké historii ještě vyšší. V této práci se však snažím zohledňovat veškeré faktory, které mají na vývoj a management strategie vliv při reálném obchodování. Tato varianta strategie poskytla na testovaném období vzorek celkem 92 obchodů, což je dostatečné množství pro fázi hrubého testování strategie.

Pro fázi detailního testování tedy bude využita varianta křížení rychlého a pomalého klouzavého průměru na trhu EUR/USD a 1 hodinovém timeframe.

Tabulka 7.2: Výsledky variant navržené obchodní strategie (při testu od 1.1.2015 – 1.11.2015). Zdroj: Autor

| indikator/instrument | 15 min, klouzavý p. | 15 min, Bollinger b. | 30 min, klouzavý p. |
|----------------------|---------------------|----------------------|---------------------|
| EUR/USD | 9742 USD | 9689 USD | 9825 USD |
| GBP/USD | 9696 USD | 9578 USD | 9805 USD |
| EUR/GBP | 9688 USD | 9595 USD | 9788 USD |

| indikator/instrument | 30 min, Bollinger b. | 1 h, klouzavý p. | 1 h, Bollinger b. |
|----------------------|----------------------|------------------|-------------------|
| EUR/USD | 9804 USD | 10038 USD | 9980 USD |
| GBP/USD | 9784 USD | 9952 USD | 9942 USD |
| EUR/GBP | 9780 USD | 9947 USD | 9957 USD |



Obrázek 7.3: Výsledky nejziskovější varianty strategie při hrubém testování. Zdroj: Metatrader 4

7.4 Detailní testování a ladění strategie

Aby bylo možné variantu strategie, která byla vybraná v předchozí fázi, detailně otestovat a případně vyladit její parametry, bylo nutné pravidla převést do podoby kódu jazyka MQL4.

V první řadě bylo potřeba nadefinovat konstanty – `Nazev` se používá v programu EA pouze jako label, `Posun` označuje, že se do této proměnné ukládá hodnota svíčky, která je druhá nejnovější, na grafu druhá zprava (označená jako `iBar [1]`, aktuální svíčka je pak označena `iBar [0]`).

```
#define Nazev          "Krizeni MA"

#define Posun          1
```

Externí proměnné slouží k uchování hodnot, které může uživatel před spuštěním EA manuálně přenastavit. V první řadě jsou definovány proměnné pro ohraničení časového úseku během dne, kdy je povoleno strategii obchodovat. Proměnné RychleMA a PomaleMA označují periody pro výpočet klouzavých průměrů určených pro vstupní signály strategie. Podobně PeriodaATR a NasobekATR slouží k výpočtu adaptivního Stop-lossu. Proměnná PouzivatMM je typu boolean a vyjadřuje, zda uživatel chce používat při obchodování Money management – počet obchodovaných lotů se tak řídí velikostí riskovaného procenta účtu (proměnná RiskovaneProcentoUctu) a proměnnou VelikostLotu. SkluzVPlneni vyjadřuje nejvyšší povolený skluz v plnění (vyjádřený v pipech) a MagicNumber slouží jako identifikátor pro obchody vytvořené touto strategií.

```
extern int    ZacatekHodina      = 00;
extern int    ZacatekMinuta     = 30;
extern int    KonecHodina       = 23;
extern int    KonecMinuta       = 30;

extern int    RychleMA          = 10;
extern int    PomaleMA          = 150;

extern int    PeriodaATR        = 50;
extern double NasobekATR        = 2.5;

extern bool   PouzivatMM        = true;
extern double RiskovaneProcentoUctu = 2;
extern double VelikostLotu      = 0.1;

extern int    SkluzVPlneni      = 4;
extern int    MagicNumber       = 12345;
```

Proměnné níže jsou datového typu Pole a slouží k uchování dat o posledních otevřených objednávkách, se kterými se pracuje v dalších uživatelských funkcích.

```
int    TicketPosledniObjednavky[2];

datetime CasPosledniObjednavky[2];

double LotyPosledniObjednavky[2];
```

Funkce `init()` a `deinit()` obsahují pouze inicializační příkazy některých proměnných a příkaz `return (0)`, který v podstatě signalizuje, že běh programu je úspěšný (předává řízení programu Terminálu). Důležitější je proto v této části práce se zaměřit na obsah funkce `start()` a důležité části mnou vytvořených uživatelských funkcí. Veškeré funkce nemohou být v hlavní části práce zobrazeny, ovšem jsou zobrazené v příloze C.

Nejprve je potřeba nadefinovat podmínku, že nelze se vstupními signály pracovat pokud aktuální čas není v nastaveném časovém rozmezí (viz nastavení externích proměnných):

```
int hour = TimeHour(TimeCurrent());
int minute = TimeMinute(TimeCurrent());

if(!((ZacatekHodina == 0 && KonecHodina == 0) ||
      (hour >= ZacatekHodina && hour <= KonecHodina)) ||

(ZacatekMinuta > 0 && hour == ZacatekHodina && minute < ZacatekMinuta) ||

(KonecMinuta > 0 && hour == KonecHodina && minute > KonecMinuta))

{ return(0); }
```

Dále je potřeba určit, že lze obchodovat jen, pokud poslední svíčka na grafu je dokončená (již je vykreslené `Close`) a obchod se tak otevře již na nové svíčce. Tuto informaci uchovává proměnná `svickaUkoncena`, která jí získává funkcí `JeSvickaUkoncena()`. První parametr této mnou vytvořené uživatelské funkce označuje zvolený timeframe – v případě hodnoty 0 je zvolený aktuální timeframe grafu, pro který se strategie používá. Druhý parametr typu boolean určuje, zda se má updatovat uchovávaná hodnota poslední svíčky na grafu (`iBar [0]`).

```
bool svickaUkoncena = JeSvickaUkoncena(0, true);

if(!svickaUkoncena) return(0);
```

Dále je potřeba uchovat hodnoty rychlého a pomalého klouzavého průměru a to jak pro předposlední svíčku na grafu – parametr `Posun` (tj. `iBar [1]`), tak předchozí – parametr `Posun+1` (tj. `iBar[2]`):

```
double fast_ma = iMA(Symbol(), 0, RychleMA, 0, 0 , PRICE_CLOSE, Posun);
```

```

double slow_ma = iMA(Symbol(), 0, PomaleMA, 0, 1, PRICE_CLOSE, Posun);

double fast_ma1 = iMA(Symbol(), 0, RychleMA, 0, 0, PRICE_CLOSE,
Posun+1);

double slow_ma1 = iMA(Symbol(), 0, PomaleMA, 0, 1, PRICE_CLOSE,
Posun+1);

```

Strategie si musí pamatovat, kolik je nákupních a prodejních objednávek je aktuálně otevřeno. Tato informace se ukládá do lokálních proměnných VsechnyObchody_buy a VsechnyObchody_sell, která ji získává z uživatelské funkce GetVsechnyObchody. První parametr vyjadřuje hodnotu Výčtového typu a označuje, zda chceme vrátit nákupní nebo prodejní objednávky. Druhý parametr je pak identifikátorem obchodů, které jsou generované právě touto strategií.

```

int VsechnyObchody_buy = GetVsechnyObchody(OP_BUY, MagicNumber);

int VsechnyObchody_sell = GetVsechnyObchody(OP_SELL, MagicNumber);

```

Pokud již není otevřen obchod stejného typu (Nákup/prodej), jako který se předpokládá po vzniku signálu a nastane vstupní signál – tedy překřížení rychlého klouzavého průměru přes pomalý klouzavý průměr na svíčce, která je již uzavřená (tedy iBar [1]) a zároveň klouzavé průměry nebyly překříženy již o jednu svíčku dříve (tedy na úrovni iBar [2]), poté se zavře případná otevřená objednávka ve směru opačném, než je vstupní objednávka, která se hned poté otevře. Po dokončení celé funkce start() následuje předání kontroly Terminálu (funkce return (0)).

```

if(VsechnyObchody_buy == 0 && fast_ma > slow_ma && fast_ma1 < slow_ma1)
{
    ZavritObjednavku(OP_SELL);
    OtevritObjednavku(OP_BUY, GetVelikostLotu());
}

if(VsechnyObchody_sell == 0 && fast_ma < slow_ma && fast_ma1 > slow_ma1)
{
    ZavritObjednavku(OP_BUY);
    OtevritObjednavku(OP_SELL, GetVelikostLotu());
}
return (0);

```

Mimo funkci start() bych se ještě rád zmínil o několika uživatelských funkcích, které jsou v programu implementovány, a které jsou v rámci funkce start() volány.

První z nich je funkce OtevritObjednavku (), která jako parametr obsahuje Typ obchodu, tedy nákupní nebo prodejní objednávku a velikost Lotu, tedy objem, ve kterém chce obchodník obchodovat. Tato velikost je však řízena automatizovaně pomocí funkce GetVelikostLotu(), která přepočítává počet lotů, se kterými se bude obchodovat podle zvoleného risku na obchod (proměnná RiskNaObchod) a aktuální výše účtu. Tím je aplikován Money-management.

```
void OtevritObjednavku(int Typ, double Lot)
{
    int err;
    color barva;
    double cena, loc_stoploss = 0;
    double loc_profittarget = 0;
    string akce;
    RefreshRates(); //obnovi hodnoty na cenovem grafu

    int Magic = MagicNumber;

    // nastavení grafu a cen pro Nakup/Prodej
    if(Typ == OP_BUY){ cena = Ask; barva = Blue; akce = "Buy"; }
    if(Typ == OP_SELL){ cena = Bid; barva = Red; akce = "Sell"; }

    // Zkontrolovat, zda je pouzivane jeste jine EA
    while (IsTradeContextBusy()) Sleep(1000); //pokud ano, pozdrz
    exekuci programu o 1000ms, neplati pro StrategyTester
    int loc_datetime = TimeCurrent();

    // Odeslat objednávku
    int loc_ticket = OrderSend(Symbol(), Typ, Lot, cena, SkluzVPlneni,
0, 0, "", Magic, 0, barva);
    // instrument, typ objednávky, loty, cena, maximalni povoleny
    skluz, stop loss level, profit target level, komentar, identifikator
    objednávky, doba expirace objednávky pokud se nestaci vyplnit, barva
    sipky na grafu

    // dalsi pokus, pokud se predchozi objednávka nevyplni
    if (loc_ticket == -1)
    {
        while(loc_ticket == -1 && TimeCurrent() - loc_datetime < 60 &&
```

```

!IsTesting()) //casovy limit je 60 ms od posledniho updatu promenne
loc_datetime, plati pouzve pro realne obchodovani a ne testovani
{
    err = GetLastError();
    if (err == 148) return; // pokud je pocet aktivnich i
cekajicich objednavek vyssi nez limit brokera, pak cekej
    Sleep(1000);
    while (IsTradeContextBusy()) Sleep(1000);

// znovu zkontrolovat, zda se nepouziva jine EA
    RefreshRates();
    loc_ticket = OrderSend(Symbol(), Typ, Lot, cena,
SkluzVPlneni, 0, 0, "", Magic, 0, barva); // dalsi pokus o odeslani
objednavky
}

    if (loc_ticket == -1)
        Print(Nazev +" (OrderSend Error) "+
ErrorDescription(GetLastError())); //jakmile casovy limit vyprsi,
zobraz chybovou hlasku
}

    if (loc_ticket != -1) //pokud se objednavka zdarila, uchovej data
o ni
    {
        CasPosledniObjednavky[Typ] = iTime(Symbol(), PERIOD_H1, 0);
        TicketPosledniObjednavky[Typ] = loc_ticket;
        LotyPosledniObjednavky[Typ] = Lot;

// Nastav Stop-loss
if(OrderSelect(loc_ticket, SELECT_BY_TICKET, MODE_TRADES))
{
    loc_profittarget =
NormalizovatTypDouble(GetProfitTarget(Typ));
    loc_stoploss = NormalizovatTypDouble(GetStopLoss(Typ));
    if(!OrderModify(loc_ticket, OrderOpenPrice(), loc_stoploss,
0 , 0, Green)) //uprava posledni objednavky

        Print(Nazev +" (OrderModify Error) "+
ErrorDescription(GetLastError())); //pokud se nepovede, zobraz
chybovou hlasku

        if(!OrderModify(loc_ticket, OrderOpenPrice(), OrderStopLoss
(), loc_profittarget , 0, Green)) //uprava posledni objednavky
            Print(Nazev +" (OrderModify Error) "+
ErrorDescription(GetLastError()));
}
}

```

```

    }
}

```

Risk-management je pak řízen funkcí GetRozsahSLPT, která pomocí indikátoru ATR určí, jaké má trh aktuálně rozpětí (viz Podkapitola 5.1.1) a na základě této informace a hodnot externích proměnných nasobekATR a periodaATR určí velikost Stop-lossu a Profit-Targetu, které se nastavují v dalších funkcích.

```

double GetRozsahSLPT(int Typ)

{
    if(Typ == OP_BUY) return(iATR(Symbol(), 0, PeriodaATR,
Posun)*NasobekATR);
    if(Typ == OP_SELL) return(iATR(Symbol(), 0, PeriodaATR,
Posun)*NasobekATR);
}

```

Funkce ZavritObjednavku() vybere všechny objednávky, které odpovídají aktuálně obchodovanému (nebo testovanému) instrumentu, strategii a zvolenému typu obchodu (nákup nebo prodej) a pomocí funkce OrderClose() je uzavře, popřípadě zobrazí chybovou hlášku.

```

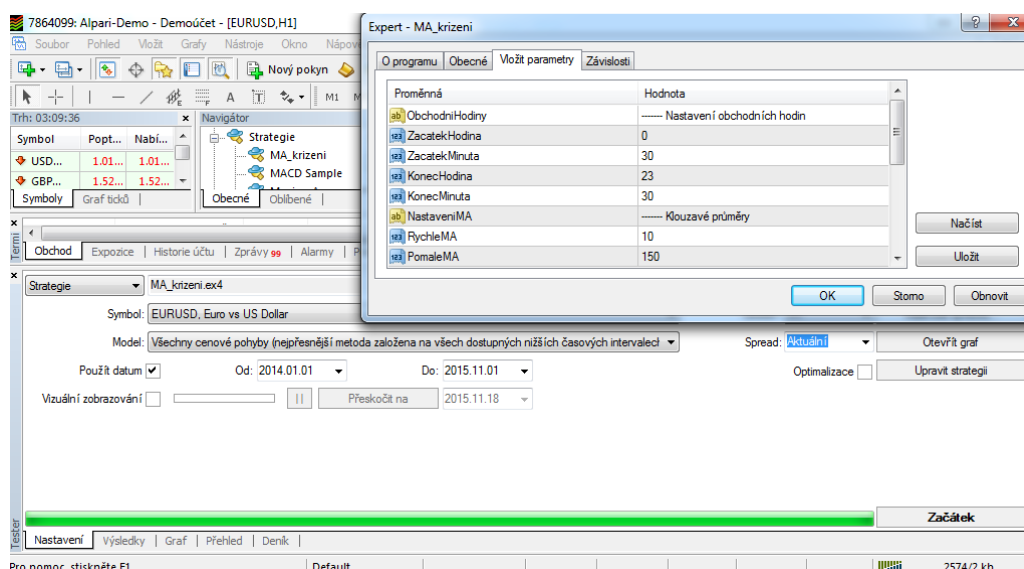
void ZavritObjednavku(int Typ)
{
    int TypObjednavky;
    int Magic = MagicNumber;
    for(int i = OrdersTotal()- 1; i >= 0; i--)
    {
        OrderSelect(i, SELECT_BY_POS, MODE_TRADES);
        TypObjednavky = OrderType();
        if(OrderSymbol() == Symbol() && OrderMagicNumber() == Magic
&& Typ == TypObjednavky)
        {
            if(!OrderClose(OrderTicket(), OrderLots(),
OrderClosePrice(), SkluzVPlneni, Purple))
//ticket uzavrene objednavky, pocet lotu, uzaviraci cena, maximalni
povolena slippage v pips, barva sipky na grafu
        }

        Print(Nazev +" (OrderClose Error) "+
ErrorDescription(GetLastError()));
    }
}

```

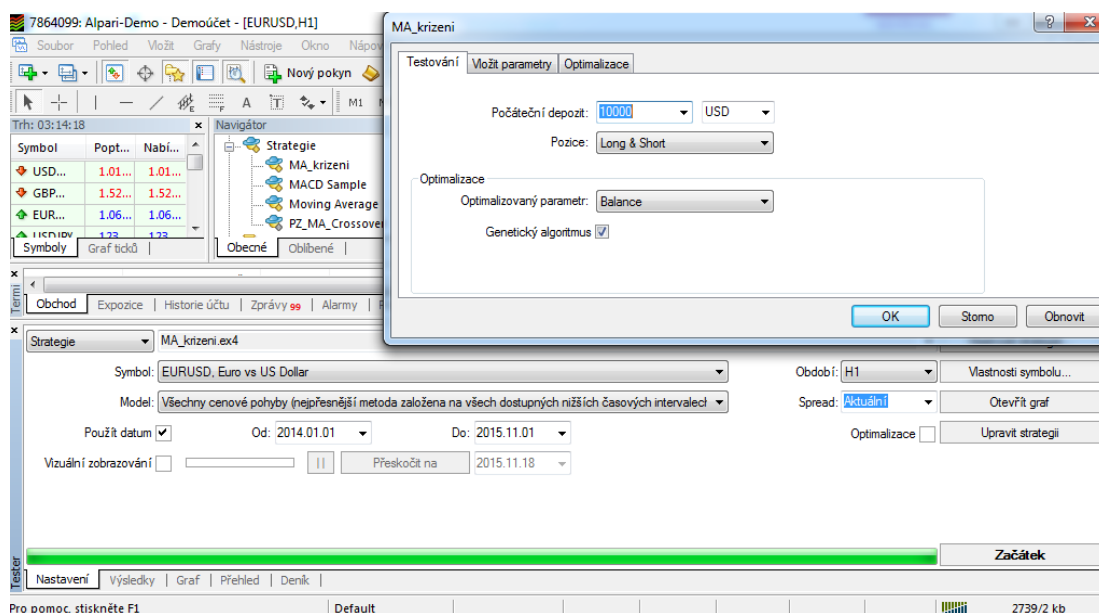

7.4.1 Zhodnocení výsledků strategie na historických datech

Po naprogramování strategie je možné ji velmi přesně a spolehlivě otestovat (což je právě jeden z nejsilnějších prvků mechanického obchodování). MetaTrader4 pro tento proces obsahuje komponentu Tester. Vybral jsem tedy parametry s nimiž byla strategie otestována - instrument EURUSD, který hrubá analýza strategie označila jako nejzajímavější a časový rámec H1 (svíčky se tedy vytvářejí po jedné hodině). Test proběhl na historických datech v období od 1.1. 2014 do 1.11.2015 (Obrázek 7.4).



Obrázek 7.10: Nastavení parametrů pro testování strategie MA_krizeni v komponentě Tester. Mimo jiné je možné upravit externí proměnné, pokud si nepřejeme zvolit defaultní hodnoty. Zdroj: Metatrader 4

Jak v případě nasazení strategie do prostředí reálného (nebo demo) účtu, tak v případě testování na historických datech, je možné volit manuálně hodnoty externích proměnných, aniž by bylo potřeba je upravovat přímo v kódu (kde jsou uloženy pouze defaultní hodnoty). Pro účely testování je navíc možné zvolit, na jak velkém počátečním kapitálu (v tomto případě bylo zvoleno 10 000 USD) chceme strategii testovat, či zda chceme uvažovat Nákupní (Long) i Prodejní (Short) objednávky (viz Obrázek 7.5). Tester dokonce umožňuje i optimalizovat hodnoty proměnných, tak aby bylo možné nalézt jejich nejvhodnější hodnotu pro obchodování na reálném trhu. Tato problematika však spadá mimo rozsah této práce.



Obrázek 7.11: Nastavení velikosti počátečního depositu fiktivního testovacího účtu a povolených typů objednávek. Zdroj: Metatrader 4

Souhrnný výsledek fungování strategie na sérii historických dat ve zvoleném období je po exekuci testu zobrazen na kartě Přehled komponenty Tester (Obrázek 7.6). Mimo tuto kartu je k dispozici také karta Výsledky (obsahuje výpis všech obchodů vytvořených v rámci exekuce testu na historických datech) a Graf (zobrazuje vývoj velikosti účtu v rámci sledovaného období v případě spuštění strategie na historických datech). Data z těchto karet jsou však již velmi obsáhlá (desítky stránek podrobných statistik).

| 7864099: Alpari-Demo - Demoúčet - [EURUSD,H1] | | | | |
|---|---|------------------------------------|------------------|---|
| Soubor Pohled Vložit Grafy Nástroje Okno Nápověda | | | | |
| Trh: 03:17:21 x Navigátor x EURUSD,M1 EURUSD,M1 USDCHF,H1 GBPUSD,H1 EURUSD,M5 EURUSD,H1 EURUSD,H1 | | | | |
| Obchod Expozice Historie účtu Zprávy 99 Alarmy Poštovní schránka 8 Obchod 57 Signály Knihožna Strategie Deník | | | | |
| Testování sloupčových grafů | 12358 | Značky namodelovány | 28804734 | Modelování kvality 90.00% |
| Chyba neshodných grafů | 0 | | | |
| Počáteční depozit | 10000.00 | | | Spread Aktuální (13) |
| Celkový čistý zisk | -118.15 | Hrubý zisk | 7531.12 | Hrubá ztráta -7649.28 |
| Ziskový faktor | 0.98 | Předpokládaný zisk | -0.72 | |
| Absolutní pokles | 579.08 | Maximální pokles | 1191.02 (10.87%) | Relativní pokles 10.87% (1191.02) |
| Transakce celkem | 163 | Krátké pozice (výhra %) | 82 (52.44%) | Dlouhé pozice (výhra %) 81 (48.15%) |
| | | Ziskové obchody (% z celkové výše) | 82 (50.31%) | Ztrátové obchody (% z celkové výše) 81 (49.69%) |
| | Největší ziskový obchod | | 197.98 | ztrátový obchod -265.40 |
| | Průměrný ziskový obchod | | 91.84 | ztrátový obchod -94.44 |
| | Maximálně návazné výhry (finanční zisk) | 5 (267.68) | | návazné prohry (finanční ztráta) 6 (-1059.22) |
| | Maximální návazný zisk (počet zisků) | 505.75 (4) | | návazná ztráta (počet ztrát) -1059.22 (6) |
| | Průměrný návazný výhry | 2 | | návazné prohry 2 |

Obrázek 7.12: Přehled výsledků testování strategie MA_krizeni v období od 1.1.2014 do 1.11.2015. Zdroj: Metatrader 4

Z výsledků je patrné, že se povedlo nasimulovat 163 historických obchodů, což splňuje podmínku, která říká, že detailní testování strategie by mělo zahrnovat minimálně 100 nasimulovaných obchodů v historii. Strategie je ve zvoleném období mírně ztrátová (-118,15 USD, což představuje přibližně 1,18% účtu), to vyjadřuje i Profit Factor, který je nižší než 1 (0,98). Velmi vyrovnaný je počet Long i Short obchodů (82 ku 81), což je logické, protože po uzavření obchodu jedním směrem se musí podle pravidel otevřít obchod druhým směrem. Dá se ovšem tvrdit, že strategie je v delším časovém horizontu (více než rok a půl) nezisková nebo spíše lehce ztrátová.

I když hlavním cílem této práce není vytvořit ziskovou strategii, nýbrž projít procesem transformace diskrečních pravidel do mechanicky obchodovatelné strategie, pokusíme se vylepšit výkonnost strategie na historických datech a tím i získat větší pravděpodobnost fungování strategie v budoucnu. K tomu se znovu pokusíme využít diskreční pohled na historická data trhu. Respektive pokusíme se tzv. „odfiltrovat“ signály, které by potenciálně nemusely poukazovat na ziskové obchody. Jelikož klouzavé průměry jsou trendovým indikátorem, měly by umožnit identifikovat situaci, kdy dochází k masivnímu nárůstu nebo poklesu ceny. Překřížení rychlého a pomalého klouzavého průmětu však samo o sobě na tuto situaci poukazovat nemusí (viz chování trhu mezi oblastmi 1 a 2 na Obrázku 7.7 níže). Pokusil jsem se tedy přidat další pravidlo – cenová akce svíček musí podpořit chování klouzavých průměrů. Při předpokládaném nákupním signálu by tedy rychlý klouzavý průměr měl překřížit pomalý klouzavý průměr zespodu nahoru a zároveň poslední uzavřená svíčka by se neměla jakkoliv dotýkat rychlého klouzavého průměru – měla by být celá nad ním. Opačně při prodejním signálu by rychlý

klouzavý průměr měl překřížit pomalý klouzavý průměr ze shora dolů a zároveň poslední uzavřená svíčka by se neměla jakkoliv dotýkat rychlého klouzavého průměru – měla by být celá pod ním.



Obrázek 7.13: Filtrování potenciálních neziskových signálů pomocí diskrétní analýzy trhu. Obchodům č. 1 (nákupní objednávka) a č. 2 (prodejní objednávka) předcházelo sice signál překřížení klouzavých průměrů, ovšem při tomto signálu se svíčka dotýkala klouzavého průměru. Oba obchody byly ztrátové. Naproti tomu obchodu č. 3 předcházelo jak překřížení klouzavých průměrů, tak svíčka, která se tělem nedotýkala klouzavého průměru – byla celá nad ním. Obchod skončil ziskem. Podobné poznatky z diskrétní analýzy trhu mohou být cenným pomocníkem při tvorbě strategií. Zdroj: Metatrader 4

Toto pravidlo lze pak jednoduše převést do podoby programového kódu mechanicky obchodovatelné strategie. Stačí vyjádřit, že Low poslední uzavřené svíčky musí být výše, než rychlý klouzavý průměr v tomto časovém úseku pro nákup instrumentu (pokud je Low výše než klouzavý průměr, pak logicky celá svíčka je výše než klouzavý průměr) a High poslední uzavřené svíčky musí být níže, než rychlý klouzavý průměr v tomto časovém úseku (pokud je High níže než klouzavý průměr, pak logicky celá svíčka je níže než klouzavý průměr).

```
If (VsechnyObchody_buy == 0 && fast_ma > slow_ma && fast_ma1 < slow_ma1
    && iLow (Symbol(), 0, 1) > fast_ma)
{
    ZavritObjednavku(OP_SELL);
    OtevritObjednavku(OP_BUY, GetVelikostLotu());
}
```

```

if(VsechnyObchody_sell == 0 && fast_ma < slow_ma && fast_ma1 > slow_ma1
    && iHigh(Symbol(), 0, 1) < fast_ma)
{
    ZavritObjednavku(OP_BUY);
    OtevritObjednavku(OP_SELL, GetVelikostLotu());
}

```

Strategii znovu zkusíme detailně otestovat pomocí komponenty Tester v programu MetaTrader 4 a to na stejném instrumentu (EURUSD) a historickém časovém rozpětí – 1.1.2014 – 1.11.2015. Timeframe ponecháváme také 1 hodina, stejně jako velikost testovacího účtu (10 000 USD) a zahrnutí jak nákupních, tak prodejních příkazů.

Pokud se podíváme na výsledky této upravené strategie (Obrázek 7.8), je patrné, že došlo k několika výrazným změnám. První z nich je, že oproti původní podobě došlo k úbytku 49 obchodů na konečných 114. To nadále splňuje podmínku o vhodném vzorku počtu obchodů při detailním testování. Co je však ještě důležitější – strategie vykazovala na konci testovaného období zisk necelých 89 USD. Ačkoliv zisk rozhodně není výrazný, došlo ke zlepšení výsledků strategie na historických datech a upravená strategie tak vykazuje o více než 200 USD vyšší zisk než strategie v původní podobě. Snížení počtu obchodů ve sledovaném období tak v podstatě znamenalo jakési odfiltrování potenciálních obchodů, které s vysokou pravděpodobností nepřinesou zisk. Tuto informaci lze také interpretovat z metriky „Ziskové obchody (% z celkové výše)“, na níž došlo k vylepšení z přibližně 50% na téměř 60%. Test mechanické podoby strategie tedy potvrdil domněnku objevenou při diskreční analýze historického vývoje trhu – zahrnutí chování ceny (svíček) ve vztahu ke klouzavým průměrům má na výsledky strategie pozitivní vliv.

7864099: Alpari-Demo - Demoučet - [EURUSD,H1]

Soubor Pohled Vložit Grafy Nástroje Okno Nápověda

Nový pokyn Automatické Obchodování

Trh: 03:53:43 Navigátor EURUSD,M1 EURUSD,M1 USDCHF,H1 GBPUSD,H1 EURUSD,M5 EURUSD,H1 EURUSD,H1

Zůstatek: 4 957.92 USD Zisk/Ztráta: 0.00 Majetek: 4 957.92 Dostupná marže: 4 957.92

| | | | | | |
|-----------------------------|-----------|------------------------------------|------------------|-------------------------------------|------------------|
| Testování sloupcových grafů | 12358 | Značky namodelovány | 28804734 | Modelování kvality | 90.00% |
| Chyba neshodných grafů | 0 | | | | |
| Počáteční depozit | 10000.00 | | | Spread | Aktuální (I0) |
| Celkový čistý zisk | 88.77 | Hrubý zisk | 6478.16 | Hrubá ztráta | -6389.39 |
| Ziskový faktor | 1.01 | Předpokládaný zisk | 0.78 | | |
| Absolutní pokles | 215.97 | Maximální pokles | 1229.58 (11.16%) | Relativní pokles | 11.16% (1229.58) |
| Transakce celkem | 114 | Krátké pozice (výhra %) | 60 (56.67%) | Dlouhé pozice (výhra %) | 54 (62.96%) |
| | | Ziskové obchody (% z celkové výše) | 68 (59.65%) | Ztrátové obchody (% z celkové výše) | 46 (40.35%) |
| | Největší | ziskový obchod | 207.80 | ztrátový obchod | -366.47 |
| | Průměrný | ziskový obchod | 95.27 | ztrátový obchod | -138.90 |
| | Maximálně | návazné výhry (finanční zisk) | 7 (374.86) | návazné prohry (finanční ztráta) | 6 (-1046.21) |
| | Maximální | návazný zisk (počet zisků) | 631.21 (5) | návazná ztráta (počet ztrát) | -1046.21 (6) |
| | Průměrný | návazné výhry | 2 | návazné prohry | 2 |

Nastavení Výsledky Graf Přehled Deník

Report testeru Default 3831/2 kb

Obrázek 7.14: Přehled výsledků testování upravené strategie MA_krizeni v období od 1.1.2014 do 1.11.2015. Zdroj: Metatrader 4

8 Závěr

Nakonec bych rád shrnul výsledky dosažené v předložené diplomové práci. Tato práce měla v první řadě za úkol shrnout tu nejdůležitější problematiku obchodování na finančních trzích a programovacího jazyka MQL4 tak, aby bylo možné vytvořit mechanickou strategii pro účel testování (a případně také pozdějšího obchodování) na trhu Forex.

Nejprve byla popsána struktura a rozdělení finančních trhů se zaměřením na trh Forex, pro který byla strategie později vytvářena. Součástí této části diplomové práce bylo také představení základních principů obchodování na tomto trhu. Dále bylo potřeba definovat hlavní subjekty finančních trhů – obchodníky a brokery. Obchodníci byli typologicky rozděleni podle způsobu analýzy a obchodování, přičemž v této práci jsem se soustředil především na obchodníky využívající technickou analýzu. V následující kapitole jsem popsal parametry a metriky obchodní strategie. Byly vybrány indikátory technické analýzy, které byly použity pro vytváření obchodní strategie v praktické části této práce, přičemž jsem také zahrnul ukázkou diskreční analýzy využití těchto indikátorů. Kapitola poté popisuje jednotlivé fáze životního cyklu vývoje obchodní strategie, z níž první čtyři jsou součástí této diplomové práce. Na závěr této kapitoly srovnávám nejpoužívanější aplikace pro obchodování na trhu Forex pomocí technické analýzy.

Diplomová práce se dále zabývá popisem historie programovacího jazyka MQL4 a jeho základních vlastností. Je popsána struktura a základní typy výstupů jazyka. Dále jsou podrobně popsány základní předdefinované proměnné a funkce, z nichž některé byly použity v praktické části práce.

Praktická část práce se soustředí na tvorbu mechanické podoby obchodní strategie. Proces tvorby obchodní strategie je v souladu s nadefinovanými fázemi životního cyklu obchodní strategie. Nejprve je vybrán vhodný broker a definovány požadavky na velikost kapitálu, které by bylo vhodné dodržet v případě reálného obchodování. Dále byl zvolen rámec (především timeframe), ve kterém chceme pravidla strategie dále hledat. Diskreční analýzou indikátorů (vycházející z teoretické části práce) na zvolených trzích jsem dospěl k několika variantám strategie. Porovnal jsem výsledky jednotlivých variant strategie a vybral takovou variantu, která poskytovala při hrubém testování nejlepší výsledky. Tím jsem splnil jeden z cílů předložené diplomové práce. Zvolenou variantu strategie jsem dále doplnil o pravidla risk-managementu a money-managementu a kompletně převedl do formy mechanické strategie. Tím jsem tedy splnil další cíl této diplomové práce. Poslední z cílů práce, a sice dosažení zisku při testování na tomto a předchozím kalendářním roce, byl poněkud těžší úkol. Nakonec se mi však podařilo strategii doplnit díky dodatečné diskreční analýze o filtr, se kterým se mi podařilo dosáhnout na zmíněném období zisku. Splnil jsem tak i třetí cíl diplomové práce.

Jsem přesvědčen, že kromě splněných cílů tato práce obsahuje také další přínosy. Jedním z nich je možnost využití vytvořeného systému pro další fáze životního cyklu obchodní strategie – tedy především pro obchodování. Na druhou stranu, je potřeba přiznat, že výsledek strategie na historických datech není natolik přesvědčivý, aby bylo možné očekávat při reálném obchodování vysoký profit. O vývoji obchodní strategie je však nutné uvažovat, jako o rozvoji reálného podnikání – není to krátkodobá aktivita, nýbrž dlouhodobý proces, kterému je potřeba mnohé obětovat. Jinými slovy – dalo by se říci, že je to „běh na dlouhou trať“. Jsem však přesvědčen, že tato diplomová práce může sloužit jako zdroj inspirace pro pokračování v tvorbě strategií pro obchodování na finančních trzích v podobném směru.

9 Seznam použitých zdrojů

Tištěné zdroje:

[Budík, 2011] BUDÍK, J.. Finanční investování. 1.vyd. Praha: Vysoká škola finanční a správní, 2011. 196s. ISBN 978-80-7408-047-0

[Hartman – Turek, 2009] HARTMAN, O. – TUREK, L.. První kroky na FOREXU. 1.vyd. Brno: Computer Press, 2009. 128s. ISBN 978-80-251-2006-4

[Nesnidal – Podhajský, 2009] NESNÍDAL, T. – PODHAJSKÝ, P.. Kompletní průvodce intradenního finančníka. 1.vyd. Praha: Centrum finančního vzdělávání, 2009. 338s. ISBN 978-80-903874-5-4

[Nývltová – Režňáková, 2007] NÝVLTOVÁ, R. – REŽŇÁKOVÁ, M.. Mezinárodní kapitálové trhy. 1.vyd. Praha: Grada, 2007. 222s. ISBN 8024719221

[Revenda a kol., 2004] REVENDA, Z. a kol.. Peněžní ekonomie a bankovníctví. 5.vyd. Praha: Management Press, 2011. 423s. ISBN 9788072612406

Elektronické zdroje:

[BCPP, 2015] Burza cenných papírů Praha [online]. C2015 [cit. 3.3.2015]. Dostupné z WWW: <https://www.pse.cz/>

[Etrading, 2015] eTrading.sk. ATR – Average True Range [online]. C2015 [cit. 3.9.2015]. Dostupné z WWW: <http://www.etrading.sk/cz/technicka-analyza/116-indikatory/671-atr-average-true-range>

[Financnik.cz, 2005] Financnik.cz [online]. C2005 [cit. 2.11.2015]. Dostupné z WWW: <http://www.financnik.cz/>

[FXStreet, 2015] FXStreet.cz. Co je Forex? [online]. C2015 [cit. 15.4.2015]. Dostupné z WWW: <http://www.fxstreet.cz/co-je-forex.html>

[IUS, 2012] IUS WIKI. Subjekty kapitálového trhu [online]. C2012 [cit. 1.5.2015]. Dostupné z WWW: http://www.ius-wiki.eu/tnh/pfuk/tnh/zkouska/otazka-79#subjekty_kapitaloveho_trhu

[Kracík, 2013] KRACÍK, L.. Co je investiční instrument Contract For Difference? [online]. C2013 [cit. 2.4.2015]. Dostupné z WWW: <http://trhy.mesec.cz/clanky/co-je-investicni-instrument-contract-for-difference/>

[Markets.com, 2015] Markets.com. Hlavní makroekonomické ukazatele [online]. C2015 [cit. 3.4.2015]. Dostupné z WWW: <http://www.markets.com/cz/education/fundamental-analysis/main-economic-indicators.html>

[MQL4, 2015] MQL4 Reference [online]. C2015 [cit. 3.11.2015]. Dostupné z WWW: <http://docs.mql4.com/>

[Pedagogická fakulta JCU, 2009] Jihočeská univerzita v Českých Budějovicích, Pedagogická fakulta, katedra matematiky. Úvod do financí [online]. C2009 [cit. 1.3.2015]. Dostupné z WWW: <http://www.pf.jcu.cz/stru/katedry/m/uf/financnitry.htm>

[RM Systém, 2008] RM System Czech Stock Exchange [online]. C2008 [cit. 3.3.2015]. Dostupné z WWW: <http://www.rmsystem.cz/>

[Wikipedia.com, 2015] Wikipedia. MetaQuotes Software [online]. C2015 [cit. 1.3.2015]. Dostupné z WWW: https://en.wikipedia.org/wiki/MetaQuotes_Software

[XTB, 2007] XTB – vst-iHned. Forex a OTC – aneb co to vlastně je...? [online]. C2007 [cit. 1.4.2015]. Dostupné z WWW: <http://byznys.ihned.cz/c1-21610490-forex-a-otc-aneb-co-to-vlastne-je>

10 Přílohy

Příloha A – varianta strategie Bollinger Bands

```
#property copyright "bol"

//---- Import knihovny
#import "stdlib.ex4"
string ErrorDescription(int e);
#import

//---- Konstanty
#define Nazev          "Uzeni Bollinger bands"
#define Posun          1

//---- Externí proměnné
extern string ObchodniHodiny          = "----- Nastavení obchodních hodin";
extern int    ZacatekHodina           = 00;
extern int    ZacatekMinuta           = 30;
extern int    KonecHodina             = 23;
extern int    KonecMinuta             = 30;

extern string NastaveniEA             = "----- Ostatní nastavení EA";
extern int    SkluzVPlneni            = 4;
extern int    MagicNumber             = 12345;

//---- Data posledních objednávek
int    TicketPosledniObjednavky[2]; // velikost pole je 2, je to pole
datetime CasPosledniObjednavky[2];
double LotyPosledniObjednavky[2];
```

```

//+-----+
//| Custom EA initialization function
//+-----+
int init()
{
    // Definice velikosti pipu a
    CasPosledniObjednavky[OP_BUY] = 0; //to je order type to v hranatych
    CasPosledniObjednavky[OP_SELL] = 0;

    Comment("MA Krizeni");

    return(0); // značí že program proběhl v pořádku
}

//+-----+
//| Custom EA deinit function
//+-----+
int deinit()
{
    return(0); // značí že program proběhl v pořádku
}

//+-----+
//| Custom EA start function
//+-----+
int start()
{

    int hour = TimeHour(TimeCurrent());
    int minute = TimeMinute(TimeCurrent());

    if(!((ZacatekHodina == 0 && KonecHodina == 0) ||
        (hour >= ZacatekHodina && hour <= KonecHodina)) ||

```

```

        (ZacatekMinuta > 0 && hour == ZacatekHodina && minute < ZacatekMinuta) ||
        (KonecMinuta > 0 && hour == KonecHodina && minute > KonecMinuta)) {
return(0); }

// Defince hodnot klouzavych prumeru
double horni = iBands(Symbol(), 0, 50, 2, 0, PRICE_CLOSE, 1, 2); // aktualni
hodnoty MA
double spodni = iBands(Symbol(), 0, 50, 2, 0, PRICE_CLOSE, 2, 2);
double aktualni_horni = iBands(Symbol(), 0, 50, 2, 0, PRICE_CLOSE, 1, 1);
double aktualni_spodni = iBands(Symbol(), 0, 50, 2, 0, PRICE_CLOSE, 2, 1);

// Pocet otevrenych objednavek
int VsechnyObchody_buy = GetVsechnyObchody(OP_BUY, MagicNumber); //MagicNumber
- identifikator objednavek vytvorených touto strategií
int VsechnyObchody_sell = GetVsechnyObchody(OP_SELL, MagicNumber);

// Nákup
if(
VsechnyObchody_buy == 0 &&
iLow (Symbol(), 0, 2) < spodni && iHigh (Symbol(), 0, 2) > horni
&& iHigh (Symbol(), 0, 1) > aktualni_horni && iLow (Symbol(), 0, 2) >
aktualni_spodni)
{
//ZavritObjednavku(OP_SELL);
int ticket = OrderSend(Symbol(), OP_BUY, 0.1, Ask, SkluzVPlneni, 0, 0, "",
12345, 0, Blue);
OrderSelect(ticket, SELECT_BY_TICKET, MODE_TRADES);
OrderModify(ticket, OrderOpenPrice(), Ask - 0.00300 , 0 , 0, Green);
OrderModify(ticket, OrderOpenPrice(), OrderStopLoss (), Ask + 0.00300 , 0,
Green);
}

// Prodej
if(
VsechnyObchody_sell == 0 &&

```

```

    iLow (Symbol(), 0, 2) < spodni && iHigh (Symbol(), 0, 2) > horni
    && iHigh (Symbol(), 0, 1) < aktualni_horni && iLow (Symbol(), 0, 2) <
aktualni_spodni)
    {
        //ZavritObjednavku(OP_BUY);
        int ticketb = OrderSend(Symbol(), OP_SELL, 0.1, Bid, SkluzVPlneni, 0, 0,
"", 12345, 0, Red);
        OrderSelect(ticketb, SELECT_BY_TICKET, MODE_TRADES);
        OrderModify(ticketb, OrderOpenPrice(), Bid + 0.00300 , 0 , 0, Green);
        OrderModify(ticketb, OrderOpenPrice(), OrderStopLoss (), Bid - 0.00300 , 0,
Green);
    }
    return (0); // program prošel úspěšně
}

//+-----+
//| Uživatelske funkce
//+-----+

/*
* Vrací počet vseh otevrenych objednavek daneho typu (Nakupni/Prodejni)
* @param int Typ
* @param int Magic
* @return int
*/
int GetVsechnyObchody(int Typ, int Magic)
{
    int pocitadlo = 0;
    for (int i = OrdersTotal() - 1; i >= 0; i--) //cyklus
    {
        if (!OrderSelect(i, SELECT_BY_POS, MODE_TRADES)) // i vybira pozici,
SELECT_BY_POS znaci razeni podle seznamu objednavek nikoliv ticketu, MODE_TRADES
jsou aktivni objednavky
        {
            Print(Nazev +" (OrderSelect Error) "+ ErrorDescription(GetLastError()));
        } else if(OrderSymbol() == Symbol() && (OrderType() == Typ || Typ ==
EMPTY_VALUE) && OrderMagicNumber() == Magic) { // parametr Magic identifikuje
vybrany order
            pocitadlo++;
        }
    }
}

```

```
    return(pocitadlo);  
}
```

Příloha B – varianta strategie Klouzavé průměry

```
#property copyright "ma_kriz"  
  
//---- Import knihovny  
#import "stdlib.ex4"  
string ErrorDescription(int e);  
#import  
  
//---- Konstanty  
#define Nazev          "Krizeni MA"  
#define Posun          1  
  
//---- Externí proměnné  
extern string ObchodniHodiny      = "----- Nastavení obchodních hodin";  
extern int    ZacatekHodina       = 00;  
extern int    ZacatekMinuta       = 30;  
extern int    KonecHodina        = 23;  
extern int    KonecMinuta        = 30;  
  
extern string NastaveniEA        = "----- Ostatní nastavení EA";  
extern int    SkluzVPlneni       = 4;  
extern int    MagicNumber        = 12345;
```

```

//---- Data posledních objednávek
int      TicketPosledniObjednavky[2]; // velikost pole je 2, je to pole
datetime CasPosledniObjednavky[2];
double   LotyPosledniObjednavky[2];

//+-----+
//| Custom EA initialization function
//+-----+
int init()
{
    // Definice velikosti pipu a
    CasPosledniObjednavky[OP_BUY] = 0; //to je order type to v hranatych
    CasPosledniObjednavky[OP_SELL] = 0;

    Comment("MA Krizeni");

    return(0); // značí že program proběhl v pořádku
}

//+-----+
//| Custom EA deinit function
//+-----+
int deinit()
{
    return(0); // značí že program proběhl v pořádku
}

//+-----+
//| Custom EA start function
//+-----+
int start()
{

    int hour = TimeHour(TimeCurrent());

```



```

int minute = TimeMinute(TimeCurrent());

if(!((ZacatekHodina == 0 && KonecHodina == 0) ||
      (hour >= ZacatekHodina && hour <= KonecHodina) ||
      (ZacatekMinuta > 0 && hour == ZacatekHodina && minute < ZacatekMinuta) ||
      (KonecMinuta > 0 && hour == KonecHodina && minute > KonecMinuta)) {
return(0); }

// Defince hodnot klouzavych prumeru
double fast_ma = iMA(Symbol(), 0, 10, 0, 0 , PRICE_CLOSE, Posun);
double slow_ma = iMA(Symbol(), 0, 150, 0, 1, PRICE_CLOSE, Posun);
double fast_ma1 = iMA(Symbol(), 0, 10, 0, 0, PRICE_CLOSE, Posun+1);
double slow_ma1 = iMA(Symbol(), 0, 150, 0, 1, PRICE_CLOSE, Posun+1);

// Pocet otevrenych objednavek
int VsechnyObchody_buy = GetVsechnyObchody(OP_BUY, MagicNumber); //MagicNumber
- identifikator objednavek vytvorených touto strategií
int VsechnyObchody_sell = GetVsechnyObchody(OP_SELL, MagicNumber);

// Nákup
if(
VsechnyObchody_buy == 0 && fast_ma > slow_ma && fast_ma1 < slow_ma1)
{

//ZavritObjednavku(OP_SELL);
int ticket = OrderSend(Symbol(), OP_BUY, 0.1, Ask, SkluzVPlneni, 0, 0, "",
12345, 0, Blue);
OrderSelect(ticket, SELECT_BY_TICKET, MODE_TRADES);
OrderModify(ticket, OrderOpenPrice(), Ask - 0.00300 , 0 , 0, Green);
OrderModify(ticket, OrderOpenPrice(), OrderStopLoss (), Ask + 0.00300 , 0,
Green);
}

```

```

// Prodej
if(
VsechnyObchody_sell == 0 && fast_ma < slow_ma && fast_ma1 > slow_ma1)
{

    //ZavritObjednavku(OP_BUY);
    int ticketb = OrderSend(Symbol(), OP_SELL, 0.1, Bid, SkluzVPlneni, 0, 0,
"", 12345, 0, Red);
    OrderSelect(ticketb, SELECT_BY_TICKET, MODE_TRADES);
    OrderModify(ticketb, OrderOpenPrice(), Bid + 0.00300 , 0 , 0, Green);
    OrderModify(ticketb, OrderOpenPrice(), OrderStopLoss (), Bid - 0.00300 , 0,
Green);
}
return (0); // program prošel úspěšně
}

//+-----+
//| Uživatelske funkce
//+-----+

/*
* Vrací pocet vseh otevrenych objednavek daneho typu (Nakupni/Prodejni)
* @param int Typ
* @param int Magic
* @return int
*/
int GetVsechnyObchody(int Typ, int Magic)
{
    int pocitadlo = 0;
    for (int i = OrdersTotal() - 1; i >= 0; i--) //cyklus
    {
        if (!OrderSelect(i, SELECT_BY_POS, MODE_TRADES)) // i vybira pozici,
SELECT_BY_POS znaci razeni podle seznamu objednavek nikoliv ticketu, MODE_TRADES
jsou aktivni objednavky
        {
            Print(Nazev +" (OrderSelect Error) "+ ErrorDescription(GetLastError()));
        } else if(OrderSymbol() == Symbol() && (OrderType() == Typ || Typ ==
EMPTY_VALUE) && OrderMagicNumber() == Magic) { // parametr Magic identifikuje
vybrany order
            pocitadlo++;

```

```
    }  
  }  
  return(pocitadlo);  
}
```

Příloha C – finální podoba strategie MA_krizeni

```
//+-----+
//| MA krizeni
//+-----+
#property copyright "MA krizeni"

//---- Import knihovny
#import "stdlib.ex4"
string ErrorDescription(int e);
#import

//---- Konstanty
#define Nazev          "Krizeni MA"
#define Posun          1

//---- Externí proměnné
extern string ObchodniHodiny      = "----- Nastavení obchodních hodin";
extern int    ZacatekHodina       = 00;
extern int    ZacatekMinuta       = 30;
extern int    KonecHodina         = 23;
extern int    KonecMinuta        = 30;

extern string NastaveniMA         = "----- Klouzavé průměry";
extern int    RychleMA            = 10;
extern int    PomaleMA           = 150;

extern string NastaveniSL         = "----- Nastavení Stop-lossu";
extern int    PeriodaATR          = 50;
extern double NasobekATR         = 2.5;

extern string NastaveniMoneyManagement = "----- Money management";
extern bool   PouzivatMM          = true;
extern double RiskovaneProcentoUctu = 2;
extern double VelikostLotu        = 0.1;
```

```

extern string NastaveniEA                = "----- Ostatní nastavení EA";
extern int    SkluzVPlneni                = 4;
extern int    MagicNumber                 = 12345;

//---- Data posledních objednávek
int    TicketPosledniObjednavky[2]; // velikost pole je 2, je to pole
datetime CasPosledniObjednavky[2];
double LotyPosledniObjednavky[2];

//+-----+
//| Custom EA initialization function
//+-----+
int init()
{
    // Definice velikosti pipu a
    CasPosledniObjednavky[OP_BUY] = 0; //to je order type to v hranatých
    CasPosledniObjednavky[OP_SELL] = 0;

    Comment("MA Krizeni");

    return(0); // značí že program proběhl v pořádku
}

//+-----+
//| Custom EA deinit function
//+-----+
int deinit()
{
    return(0); // značí že program proběhl v pořádku
}

//+-----+
//| Custom EA start function
//+-----+

```

```

int start()
{

    // Nastaveni lokalnich casovych promennych
    int hour = TimeHour(TimeCurrent());
    int minute = TimeMinute(TimeCurrent());

    // Pravidla pro obchodovani budou fungovat jen ve vymezenych hodinach
    if(!((ZacatekHodina == 0 && KonecHodina == 0) ||
        (hour >= ZacatekHodina && hour <= KonecHodina) ||
        (ZacatekMinuta > 0 && hour == ZacatekHodina && minute < ZacatekMinuta) ||
        (KonecMinuta > 0 && hour == KonecHodina && minute > KonecMinuta)) {
return(0); }

    // Nastaveni promenne, podle toho zda je posledni svicka ukoncena
    bool svickaUkoncena = JeSvickaUkoncena(0, true);

    //--
    //-- Vstupni pravidla
    //--

    // Kontrola, zda je svicka ukoncena
    if(!svickaUkoncena) return(0);

    // Defince hodnot klouzavych prumeru
    double fast_ma = iMA(Symbol(), 0, RychleMA, 0, 0 , PRICE_CLOSE, Posun); //
aktualni hodnoty MA
    double slow_ma = iMA(Symbol(), 0, PomaleMA, 0, 1, PRICE_CLOSE, Posun);
    double fast_ma1 = iMA(Symbol(), 0, RychleMA, 0, 0, PRICE_CLOSE, Posun+1); //
hodnoty MA o jeden casovy ramec zpet
    double slow_ma1 = iMA(Symbol(), 0, PomaleMA, 0, 1, PRICE_CLOSE, Posun+1);

    // Pocet otevrenych objednavek
    int VsechnyObchody_buy = GetVsechnyObchody(OP_BUY, MagicNumber); //MagicNumber
- identifikator objednavek vytvorených touto strategií
    int VsechnyObchody_sell = GetVsechnyObchody(OP_SELL, MagicNumber);

    // Nákup

```

```

    if(VsechnyObchody_buy == 0 && fast_ma > slow_ma && fast_ma1 < slow_ma1 && iLow
(Symbol(), 0, 1) > fast_ma)
    {
        ZavritObjednavku(OP_SELL);
        OtevritObjednavku(OP_BUY, GetVelikostLotu());
    }

    // Prodej
    if(VsechnyObchody_sell == 0 && fast_ma < slow_ma && fast_ma1 > slow_ma1 &&
iHigh(Symbol(), 0, 1) < fast_ma)
    {
        ZavritObjednavku(OP_BUY);
        OtevritObjednavku(OP_SELL, GetVelikostLotu());
    }
    return (0); // program prošel úspěšně
}

//+-----+
//| Uživatelske funkce
//+-----+

/**
 * Vrací pocet vseh otevrenych objednavek daneho typu (Nakupni/Prodejni)
 * @param int Typ
 * @param int Magic
 * @return int
 */
int GetVsechnyObchody(int Typ, int Magic)
{
    int pocitadlo = 0;
    for (int i = OrdersTotal() - 1; i >= 0; i--) //cyklus
    {
        if (!OrderSelect(i, SELECT_BY_POS, MODE_TRADES)) // i vybira pozici,
SELECT_BY_POS znaci razeni podle seznamu objednavek nikoliv ticketu, MODE_TRADES
jsou aktivni objednavky
        {
            Print(Nazev +" (OrderSelect Error) "+ ErrorDescription(GetLastError()));
        } else if(OrderSymbol() == Symbol() && (OrderType() == Typ || Typ ==
EMPTY_VALUE) && OrderMagicNumber() == Magic) { // parametr Magic identifikuje
vybrany order

```

```

        pocitadlo++;
    }
}
return(pocitadlo);
}

/**
 * Zavira zvoleny typ objednávky
 * @param int Typ
 */
void ZavritObjednavku(int Typ)
{
    int TypObjednavky;
    int Magic = MagicNumber;
    for(int i = OrdersTotal()- 1; i >= 0; i--)
    {
        OrderSelect(i, SELECT_BY_POS, MODE_TRADES);
        TypObjednavky = OrderType();
        if(OrderSymbol() == Symbol() && OrderMagicNumber() == Magic && Typ
        == TypObjednavky)
        {
            if(Typ == OP_BUY || Typ == OP_SELL)
            {
                if(!OrderClose(OrderTicket(), OrderLots(), OrderClosePrice(),
                SkluzVPlneni, Purple)) //ticket uzavrene objednávky, pocet lotu, uzaviraci cena,
                maximalni povolena slippage v pips, barva sipky na grafu
                Print(Nazev +" (OrderClose Error) "+
                ErrorDescription(GetLastError()));
            }
        }
    }
}

/**
 * Otevira objednávku daneho typu
 * @param int Typ
 * @param double Lot
 */
void OtevritObjednavku(int Typ, double Lot)

```



```

{
    int err;

    color barva;

    double cena, loc_stoploss = 0;
    double loc_profittarget = 0;
    string akce;
    RefreshRates(); //obnovi hodnoty na cenovem grafu

    int Magic = MagicNumber;

    // nastavení grafu a cen pro Nakup/Prodej
    if(Typ == OP_BUY){ cena = Ask; barva = Blue; akce = "Buy"; }
    if(Typ == OP_SELL){ cena = Bid; barva = Red; akce = "Sell"; }

    // Zkontrolovat, zda je pouzivane jeste jine EA
    while (IsTradeContextBusy()) Sleep(1000); //pokud ano, pozdrz exekuci programu
o 1000ms, neplati pro StrategyTester

    int loc_datetime = TimeCurrent();

    // Odeslat objednávku

    int loc_ticket = OrderSend(Symbol(), Typ, Lot, cena, SkluzVPlneni, 0, 0, "",
Magic, 0, barva);

    // instrument, typ objednávky, loty, cena, maximalni povoleny skluz, stop loss
level, profit target level, komentar, identifikator objednávky, doba expirace
objednavky pokud se nestaci vyplnit, barva sipky na grafu

    // dalsi pokus, pokud se predchozi objednávka nevyplni
    if (loc_ticket == -1)
    {
        while(loc_ticket == -1 && TimeCurrent() - loc_datetime < 60 &&
!IsTesting()) //casovy limit je 60 ms od posledniho updatu promenne loc_datetime,
plati pouze pro realne obchodovani a ne testovani
        {
            err = GetLastError();

            if (err == 148) return; // pokud je pocet aktivnich i cekajicich
objednavek vyssi nez limit brokera, pak cekej

            Sleep(1000);

            while (IsTradeContextBusy()) Sleep(1000); // znovu zkontrolovat, zda se
nepouziva jine EA

            RefreshRates();

```

```

        loc_ticket = OrderSend(Symbol(), Typ, Lot, cena, SkluzVPlneni, 0, 0, "",
Magic, 0, barva); // dalsi pokus o odeslani objednávky
    }
    if (loc_ticket == -1)
        Print(Nazev +" (OrderSend Error) "+ ErrorDescription(GetLastError()));
//jakmile casovy limit vyprsi, zobraz chybovou hlasku
    }

if (loc_ticket != -1) //pokud se objednávka zdarila, uchovej data o ni
{
    CasPosledniObjednavky[Typ]      = iTime(Symbol(), PERIOD_H1, 0);
    TicketPosledniObjednavky[Typ]   = loc_ticket;
    LotyPosledniObjednavky[Typ]     = Lot;

    // Nastav Stop-loss
    if(OrderSelect(loc_ticket, SELECT_BY_TICKET, MODE_TRADES))
    {
        loc_profittarget = NormalizovatTypDouble(GetProfitTarget(Typ));
        loc_stoploss = NormalizovatTypDouble(GetStopLoss(Typ));
        if(!OrderModify(loc_ticket, OrderOpenPrice(), loc_stoploss, 0 , 0,
Green)) //uprava posledni objednávky
            Print(Nazev +" (OrderModify Error) "+
ErrorDescription(GetLastError())); //pokud se nepovede, zobraz chybovou hlasku

        if(!OrderModify(loc_ticket, OrderOpenPrice(), OrderStopLoss (),
loc_profittarget , 0, Green)) //uprava posledni objednávky
            Print(Nazev +" (OrderModify Error) "+
ErrorDescription(GetLastError()));
    }
}

/**
 * Spocitej velikost lotu na zaklade money managementu
 * @return double
 */
double GetVelikostLotu()
{
    double loc_lot = VelikostLotu;

```

```

// Zjistí informace o symbolu a jeho nastavení pro loty
double loc_minlot = MarketInfo(Symbol(), MODE_MINLOT);
double loc_maxlot = MarketInfo(Symbol(), MODE_MAXLOT);
double loc_lotstep = MarketInfo(Symbol(), MODE_LOTSTEP);

//pomocna promenna pro interpretaci hodnoty promenne loc_lotstep
int vp = 0;
if(loc_lotstep == 0.01) vp = 2;
else vp = 1;

// aplikace money managementu
if(PouzivatMM == true)
    loc_lot = MathFloor(AccountBalance() * RiskovaneProcentoUctu / 100.0) /
1000.0;

loc_lot = NormalizeDouble(loc_lot, vp); // zaokrouhlení podle počtu
desetinnych mist za carkou promenne loc_lotstep

// Zkontroluj a pripadne prenavstav na max/minlot podle informaci o symbolu
if (loc_lot < loc_minlot) loc_lot = loc_minlot;
if(loc_lot > loc_maxlot) loc_lot = loc_maxlot;

// Navratova hodnota
return (loc_lot);
}

/**
 * Vrátí počáteční Stop-loss
 * @param int Typ
 * @return double
 */
double GetStopLoss(int Typ)
{
    double loc_sl = 0;
    double loc_risk = GetRozsahSLPT(Typ);
    if(Typ == OP_BUY) loc_sl = Ask - loc_risk - (Ask - Bid);
}

```

```

    if(Typ == OP_SELL) loc_sl = Bid + loc_risk + (Ask - Bid);
    return (loc_sl);
}

/**
 * Vrátí počáteční Profit-Target
 * @param int Typ
 * @return double
 */
double GetProfitTarget(int Typ)
{
    double loc_pt = 0;
    double loc_risk = GetRozsahSLPT(Typ);
    if(Typ == OP_BUY) loc_pt = Ask + loc_risk + (Ask - Bid);
    if(Typ == OP_SELL) loc_pt = Bid - loc_risk - (Ask - Bid);
    return (loc_pt);
}

/**
 * Vrací rozsah stop-lossu spočítaný pomocí indikátoru ATR
 * @param int Typ
 * @return double
 */
double GetRozsahSLPT(int Typ)
{
    if(Typ == OP_BUY) return(iATR(Symbol(), 0, PeriodaATR, Posun)*NasobekATR);
    if(Typ == OP_SELL) return(iATR(Symbol(), 0, PeriodaATR, Posun)*NasobekATR);
}

/**
 * Normalizovat typ Double
 * @param double price
 * @return double
 */
double NormalizovatTypDouble(double price)
{

```

```

    return (NormalizeDouble(price, Digits));
}

/**
 * Kontroluje, zda je vykreslování svíčky dokončené
 * @return bool
 * @param int timeframe
 * @param bool reset
 */
bool JeSvickaUkoncena(int timeframe,bool reset)
{
    static datetime posledniSvicka;
    if(timeframe==-1)
    {
        if(reset)
            posledniSvicka=0;
        else
            posledniSvicka=iTime(NULL,timeframe,0);
        return(true);
    }
    if(iTime(NULL,timeframe,0)==posledniSvicka) // cekej na dalsi svicku
        return(false);
    if(reset)
        posledniSvicka=iTime(NULL,timeframe,0);
    return(true);
}

```