

VYSOKÁ ŠKOLA POLYTECHNICKÁ JIHLAVA

Aplikovaná informatika

NÁVRH A IMPLEMENTACE OBCHODNÍHO BOTA
V JAZYCE MQL5

Bakalářská práce

Autor práce: Petr Matoušů

Vedoucí práce: Ing. Marek Musil

Jihlava 2024

Vysoká škola polytechnická Jihlava

Tolstého 16, 586 01 Jihlava

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Autor práce: **Petr Matoušů**

Studijní program: Aplikovaná informatika

Obor: Aplikovaná informatika

Garant studijního programu: Ing. Lenka Kuklišová Pavelková, Ph.D.

Název práce: **Návrh a implementace obchodního bota v jazyce MQL5**

Vedoucí práce: Ing. Marek Musil

Cíl práce: Bakalářská práce se bude zabývat návrhem a implementací obchodního bota v jazyce MQL5. Tento jazyk se používá na programování obchodních systémů na platformě MetaTrader 5. Cílem práce je vytvořit funkčního bota, který bude schopen automaticky obchodovat na finančních trzích na základě předem definovaných strategií. Výsledkem práce bude funkční obchodní bot, který dokáže automaticky vyhledávat obchodní příležitosti a provádět obchody na základě definovaných pravidel. V rámci práce bude rovněž provedena analýza a testování vytvořeného bota na historických datech.

Abstrakt

Bakalářská práce se zaměřuje na návrh a implementaci obchodního systému v prostředí MetaTrader 5 s využitím jazyka MQL5. Cílem práce je vytvořit funkční obchodního bota schopného automatického obchodování na forexovém trhu. Strategie obchodního systému bude zahrnovat kombinaci technických indikátorů, konkrétně exponenciálního klouzavého průměru, relative strength index a engulfing pattern. První část práce se věnuje úvodnímu seznámení s forexovým trhem, MetaTrader 5 a MQL5. Dále jsou popsány klíčové technické indikátory, které budou použity v návrhu obchodní strategie. Hlavní část práce se zabývá samotnou implementací obchodního systému. Jsou zde popsány postupy vstupu do obchodů, využití technických indikátorů, správa kapitálu a testování systému na historických datech. Výsledkem práce je funkční obchodní systém, který může sloužit samostatným investorům a obchodníkům k automatizaci obchodování na forexovém trhu. Zhodnocení efektivity implementovaných strategií a jejich přínosu pro dosažení finančních cílů je součástí závěrečného hodnocení práce.

Klíčová slova

Obchodní bot, Automatický obchodní systém, Forex, MQL5, MetaTrader 5

Abstract

Bachelor's thesis focuses on the design and implementation of a trading system in the MetaTrader 5 environment using the MQL5 language. The aim of the thesis is to create a functional trading bot capable of automatic trading in the forex market. The trading system's strategy will involve a combination of technical indicators, specifically the exponential moving average, relative strength index, and engulfing pattern. The first part of the thesis introduces the forex market, MetaTrader 5, and MQL5. Key technical indicators to be used in the design of the trading strategy are also described. The main part of the thesis deals with the actual implementation of the trading system. Procedures for entering trades, utilizing technical indicators, capital management, and testing the system on historical data are described. The result of the work is a functional trading system that can serve individual investors and traders for automating trading in the forex market. Evaluating the effectiveness of implemented strategies and their contribution to achieving financial goals is part of the final assessment of the thesis.

Keywords

Trading bot, Automated trading system, Forex, MQL5, MetaTrader 5

Prohlašuji, že předložená bakalářská práce je původní a zpracoval/a jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil/a autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů, v platném znění, dále též „AZ“).

Byl/a jsem seznámen/a s tím, že na mou bakalářskou práci se plně vztahuje **AZ**, zejména § 60 (školní dílo).

Podle § 47b zákona o vysokých školách souhlasím se zveřejněním své práce podle Směrnice pro vedení, vypracování a zveřejňování závěrečných prací na VŠPJ, a to bez ohledu na výsledek obhajoby.

Beru na vědomí, že VŠPJ má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **souhlasím** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom/a toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem VŠPJ, která má právo ode mě požadovat přiměřený příspěvek na úhradu nákladů, vynaložených vysokou školou na vytvoření díla (až do jejich skutečné výše), z výdělku dosaženého v souvislosti s užitím díla či poskytnutím licence.

V Jihlavě dne 8. dubna 2024

.....

Podpis studenta/ky

Obsah

Seznam obrázků.....	6
Seznam zkratk.....	7
Úvod.....	8
1 Základní teoretické pojmy.....	9
1.1 Forexový trh.....	9
1.2 MetaTrader 5	11
1.3 MQL5.....	12
1.4 Automatické obchodní systémy	13
1.5 Money Management.....	14
1.6 Skalpování.....	15
1.7 Exponenciální klouzavý průměr	16
1.8 Relative strength index.....	17
1.9 Engulfing pattem	17
2 Obchodní strategie	19
2.1 Vstup a výstup z obchodů.....	19
2.2 Správa financí.....	19
3 Implementace	21
3.1 Základní architektura	21
3.2 Technické indikátory	25
3.3 Získání dat ze svíček	28
3.4 Správa obchodních operací.....	29
3.5 Řízení obchodních pozic	30
4 Testování.....	35
Závěr	37
Seznam použité literatury	38
Přílohy.....	39

Seznam obrázků

Obr. 1: Obchodní hodiny na forexovém trhu	10
Obr. 2: Prostředí MetaTrader5	11
Obr. 3: Zobrazení EMA v grafu.....	16
Obr. 4: Zobrazení RSI pod grafem	17
Obr. 6: Úspěšnost obchodů v různých hodinách pro časový rámec M5.....	35

Seznam zkratek

EMA	Exponencionální klouzavý průměr
FX	Forex
MQL5	MetaQuotes Language 5
MT5	MetaTrader 5
RSI	Relative strength index
SEČ	Středoevropský čas

Úvod

Bakalářská práce se zaměřuje na návrh a implementaci obchodního bota v jazyce MQL5 pro platformu MetaTrader 5. Hlavní motivací byla fascinace možnostmi, jak mohou moderní technologie převést obchodní strategie do automatického formátu a umožnit tak efektivní reakci na neustále se měnící tržní podmínky, bez nutnosti neustálého dohledu. Práce představuje pokus o vytvoření nástroje, který by dokázal spojit svět algoritmů s dynamickým prostředím forexového trhu.

Současná řešení se často setkávají s limitací v adaptabilitě a možnostech přizpůsobení, práce směřuje k vytvoření obchodního bota, jenž by nejen s přesností reagoval na aktuální tržní podmínky, ale také poskytoval uživatelům prostor pro personalizaci obchodních strategií a parametrů. Jádrem této práce je integrace exponenciálního klouzavého průměru, relative strength index, a engulfing vzoru jako klíčových prvků obchodní strategie. Tyto indikátory jsou základem pro rozvoj automatizovaného obchodování nabízející pevný základ pro přesné obchodní rozhodnutí založené na osvědčených tržních principech.

Přípravná fáze výzkumu zahrnovala průzkum literatury a analýzu tržních dat, což umožnilo konstrukci bota, jenž je schopen prokazatelně fungovat v reálných tržních podmínkách. Práce tak reflektuje nejen aktuální potřeby trhu, ale také otevírá diskuzi o budoucnosti automatizovaného obchodování a jeho potenciálu poskytnout obchodníkům a investorům významnou konkurenční výhodu.

V kontextu této práce se bot bude věnovat reakci na trh pomocí finančních indikátorů a správě finančního portfolia s maximální mírou soběstačnosti. Důvodem pro vývoj právě tohoto bota je rostoucí potřeba efektivnějších, přesnějších a více adaptabilních obchodních systémů na finančních trzích. Cílem je najít a implementovat co nejefektivnější obchodní strategii, která dokáže reagovat na aktuální situaci a automaticky vyhledávat obchodní příležitosti, přičemž testování na historických datech odhalí potenciální úspěšnost budoucích obchodů. Výsledek práce nabízí potenciál pomoci samostatným investorům a obchodníkům dosahovat lepších výsledků bez lidské chyby a zvyšuje jejich konkurenční výhodu. Kromě toho mohou výsledky této práce sloužit finančním institucím ke zlepšení jejich obchodních strategií a zvýšení výkonnosti na trzích, což ukazuje širší aplikabilitu a význam tohoto výzkumu.

1 Základní teoretické pojmy

Kapitola slouží jako úvod do světa forexového trhu, platformy MetaTrader 5 a programovacího jazyka MQL5, přičemž nastavuje pevné teoretické základy pro pochopení následujících kapitol zabývajících se praktickou implementací a testováním obchodního bota. V této sekci jsou systematicky představeny klíčové pojmy a mechanismy, které jsou nezbytné pro navigaci ve složitém prostředí automatizovaného obchodování. Začínáme podrobným pohledem na forexový trh, jeho strukturu, důležitost a fungování, což je klíčové pro pochopení dynamiky, s níž náš obchodní bot bude interagovat. Následuje představení platformy MetaTrader 5, jejíž rozhraní a nástroje poskytují základ pro vývoj a spuštění algoritmických obchodních strategií. Dále je věnována pozornost MQL5, pokročilému programovacímu jazyku, který umožňuje detailní a flexibilní vývoj obchodních robotů a indikátorů. Kapitola také zahrnuje základní pojmy spojené s automatickými obchodními systémy, money managementem, skalpováním a dalšími relevantními technikami a strategiemi, které jsou aplikovány v kontextu vytváření efektivních obchodních algoritmů. S pevným teoretickým základem budeme schopni lépe porozumět komplexnosti a výzvám spojeným s návrhem a implementací obchodního bota během práce.

1.1 Forexový trh

Forexový trh, zkráceně FX trh nebo forex, představuje globální decentralizovaný trh, na kterém se obchoduje s měnami. Je to největší a nejlikvidnější finanční trh na světě, kde účastníci mohou nakupovat, prodávat, směňovat a spekulovat s různými měnami. Forexový trh operuje 24 hodin denně, pět dní v týdnu, a jeho prostřednictvím se realizují transakce v řádech bilionů dolarů denně (XTB, 2020).

Charakteristika Forexového trhu (XTB, 2020):

- **Globalita:** Forex je otevřený nepřetržitě 24 hodin denně, což umožňuje obchodníkům reagovat na události v reálném čase napříč různými časovými zónami. Tato kontinuální dostupnost podporuje likviditu a umožňuje flexibilitu v obchodování.
- **Likvidita:** Likvidita je jednou z klíčových vlastností forexového trhu. Díky své obrovské velikosti mohou obchodníci snadno nakupovat a prodávat bez významného vlivu na cenu. To znamená, že transakce mohou být prováděny s minimálním rozdílem mezi nákupní a prodejní cenou.
- **Různorodost účastníků:** Forexový trh přitahuje širokou škálu účastníků, včetně centrálních bank, institucionálních investorů, maloobchodníků a spekulantů. Různorodost přispívá k dynamice trhu a umožňuje různým hráčům sdílet a reagovat na informace rychle a účinně.

Důležitost forexu ve světové ekonomice

Forexový trh hraje klíčovou roli ve světové ekonomice tím, že umožňuje mezinárodní obchod a financování. Není omezen na konkrétní geografickou oblast, což usnadňuje globální obchod a pohyb kapitálu mezi různými zeměmi. Vývoj kurzů měn na forexu odráží ekonomické události a trendy, což umožňuje analytikům a obchodníkům předvídat směr ekonomik a trhů.

Vzhledem k rostoucí globalizaci a propojení ekonomik se forexový trh stal klíčovým nástrojem pro řízení měnových rizik, diverzifikaci portfolia a dosahování finančních cílů. Jeho důležitost v moderní ekonomice zdůrazňuje potřebu porozumění jeho charakteristik a funkcí pro každého, kdo se zajímá o finanční trhy a investování (XTB, 2020).

Obchodní hodiny na forexovém trhu

Forexový trh, otevřený nepřetržitě po většinu pracovního týdne, přináší jedinečný dynamický charakter, který je zásadní pro obchodníky a investory. Obchodní hodiny na forexovém trhu jsou rozděleny do několika klíčových sekcí, zohledňující časové pásmy hlavních finančních center (Stibor, 2023):

1. Asie (Tokyo)
 - Časové pásmo: 01:00-10:00 SEČ
 - Charakteristika: Asie je prvním finančním centrem, které otevírá forexový trh. Obchodní aktivita bývá nižší než v dalších sekcích, ale může být významně ovlivněna událostmi z předešlého dne.
2. Evropa (Londýn)
 - Časové pásmo: 09:00-18:00 SEČ
 - Charakteristika: Londýn je jedním z nejaktivnějších obchodních center na světě. Otevření Londýna přináší nárůst likvidity a volatility na trhu, což je období často využívané pro výrazné obchodní příležitosti.
3. Severní Amerika (New York)
 - Časové pásmo: 14:00-23:00 SEČ
 - Charakteristika: Překryv s Londýnem vytváří jedno z nejaktivnějších období obchodování. New York přináší další vlnu likvidity a často reaguje na události z Evropy.
4. Pacifik (Sydney)
 - Časové pásmo: 23:00-08:00 SEČ
 - Charakteristika: Ačkoliv aktivita není tak výrazná jako v Londýně nebo New Yorku, Sydney otevírá nový obchodní den a představuje poslední příležitost před víkendem.

23	24	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
										London													
															New York								
		Sydney																					
		Tokyo																					

Obr. 1: Obchodní hodiny na forexovém trhu

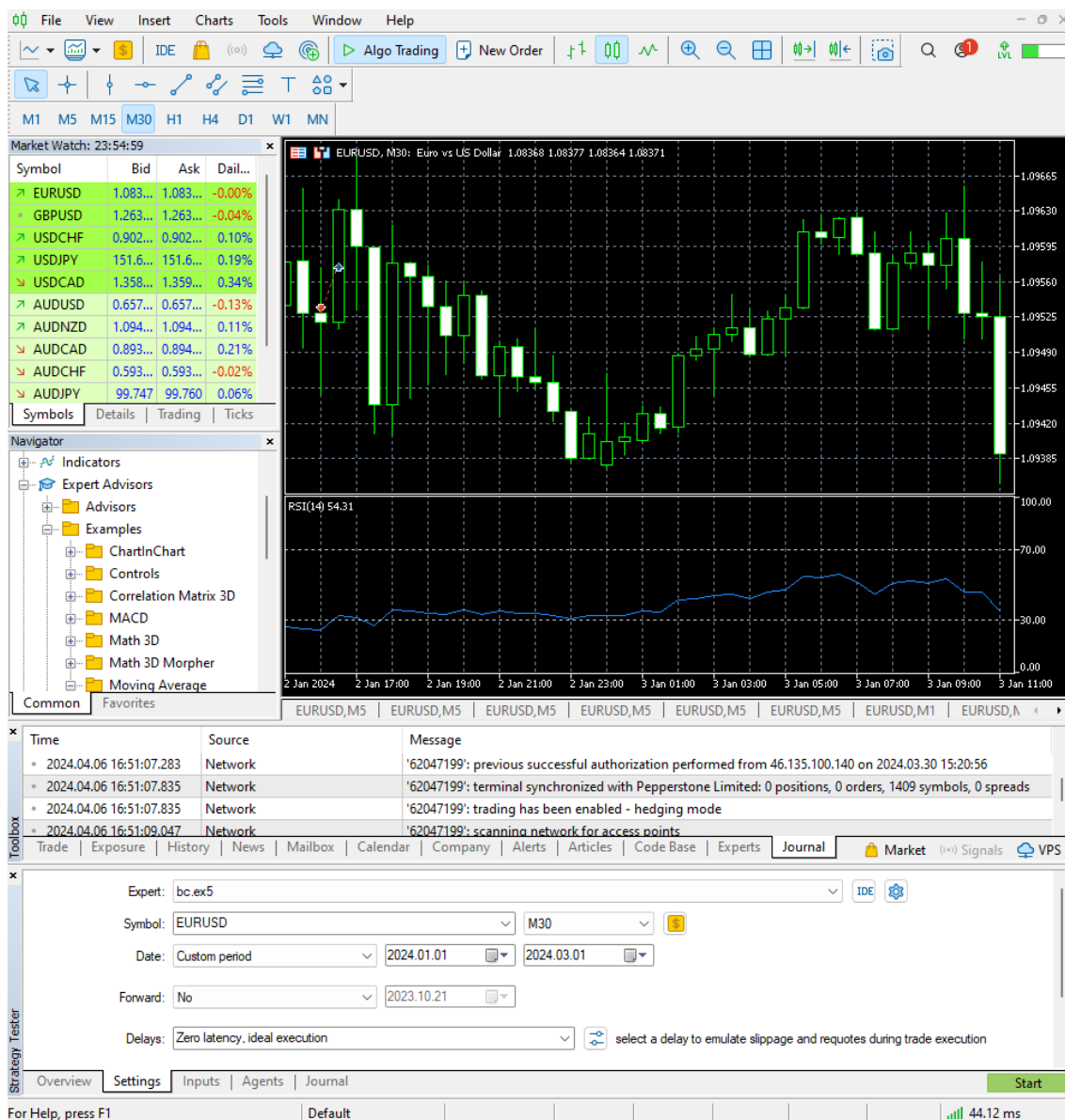
Zdroj: Michal Stibor, www.forex-zone.cz(2023)

Vliv obchodních hodin na strategie

Důkladné porozumění obchodním hodinám je klíčové pro správné plánování a implementaci obchodní strategie. Obchodníci často využívají specifických časových oken, kdy se překrývají klíčová centra, aby maximalizovali šance na úspěch. Tato dynamika otevírá možnosti pro různé obchodní strategie založené na časovém rámci a volatilitě v průběhu dne.

1.2 MetaTrader 5

Obchodní platforma MetaTrader 5, vyvinutá společností MetaQuotes Software Corp., je známa svou schopností efektivně sloužit potřebám jak nováčků, tak zkušených obchodníků na globálním finančním trhu. MetaTrader 5, nástupce populární platformy MetaTrader 4, nabízí široké spektrum finančních instrumentů včetně měn, akcií, dluhopisů a dalších derivátů, čímž umožňuje obchodníkům diverzifikaci jejich investičních strategií a maximalizaci výnosů z různých tržních segmentů. S 24hodinovým přístupem k tržím poskytuje obchodníkům z celého světa přístup k univerzálnímu obchodnímu prostředí (MetaQuotes Ltd., 2024).



Obr. 2: Prostředí MetaTrader5

Zdroj: Vlastní zpracování

Jedním z klíčových prvků, které odlišují MetaTrader 5 od jiných obchodních platform, je jeho integrované vývojové prostředí pro programování v MQL5. Toto umožňuje obchodníkům a vývojářům vytvářet vlastní obchodní boty a analytické indikátory, které dokáží automatizovat obchodní procesy. Taková úroveň personalizace a automatizace představuje revoluci v způsobu, jakým jednotlivci přistupují k obchodování, a otevírá nové možnosti pro strategické experimentování a optimalizaci (MetaQuotes Ltd., 2024).

Funkce MetaTraderu 5 (MetaQuotes Ltd., 2024):

- Multiměnová podpora: MT5 umožňuje obchodovat s různými měnami a finančními nástroji, což zahrnuje forex, komodity, akcie a indexy.
- Pokročilé grafické nástroje: Platforma poskytuje pokročilé grafy a analytické nástroje, včetně různých časových rámců a typů grafů.
- Algoritmické obchodování: S využitím jazyka MQL5 mohou uživatelé vytvářet vlastní obchodní strategie a indikátory nebo plně automatizované obchodní systémy.
- Rozhraní pro technickou analýzu: MT5 obsahuje širokou škálu technických indikátorů, grafických objektů a analytických nástrojů pro podporu fundovaných obchodních rozhodnutí.
- Ekonomický kalendář a zprávy: Uživatelé mají přístup k ekonomickému kalendáři a zprávám, které mohou ovlivnit finanční trhy.
- Bezpečnost a šifrování: MT5 vyniká bezpečnostními opatřeními, zahrnujícími šifrování dat a bezpečný přístup k obchodním účtům.
- České rozhraní: Velké lákadlo pro spoustu tuzemských obchodníků.

1.3 MQL5

MQL5 je programovací jazyk speciálně navržený pro použití v MetaTrader 5. Jeho hlavním účelem je usnadnit tvorbu obchodních botů, indikátorů a skriptů s důrazem na jednoduchost, efektivitu a rozšiřitelnost. Programátoři využívají MQL5 k vytváření automatizovaných obchodních systémů, které reagují na tržní podmínky a samostatně provádějí obchody. Uživatelská komunita a podpora pro MQL5 jsou nepostradatelnými zdroji pro jakékoli programátory či obchodníky, kteří se chtějí naučit pracovat s tímto jazykem nebo zdokonalit své dovednosti. Online fóra, tutoriály a příručky jsou plné rad, příkladů kódů a řešení, což usnadňuje nováčkům rychlý start a všem umožňuje sdílet znalosti a inovace (MetaQuotes Ltd., 2020).

Klíčové charakteristiky (MetaQuotes Ltd., 2020):

- Objektově orientovaný přístup: MQL5 využívá objektově orientovaný model programování, což zjednodušuje a zpřehledňuje strukturu kódu.
- Syntaxe podobná C++: Syntaxe MQL5 je podobná jazyku C++, což usnadňuje programátorům s předchozími zkušenostmi v programování.
- Funkce pro technickou analýzu: MQL5 poskytuje širokou paletu vestavěných funkcí pro technickou analýzu, což zahrnuje indikátory, oscilátory a nástroje pro analýzu časových řad.
- Správa obchodů: S MQL5 lze pohodlně řídit a spravovat obchody, včetně otevírání, uzavírání a modifikace obchodních pozic.

- Komunikace s externími zdroji: MQL5 podporuje komunikaci s externími zdroji dat a informací, což umožňuje dynamickou aktualizaci strategií na základě aktuálních informací.
- Backtesting a optimalizace: Pro ověření výkonnosti strategií poskytuje MQL5 možnosti backtestingu na historických datech a optimalizace parametrů strategií.

1.4 Automatické obchodní systémy

Automatické obchodní systémy představují klíčový prvek moderního finančního obchodování, kde se algoritmické strategie stávají stále důležitějšími nástroji pro obchodníky. Tyto systémy, také známé jako obchodní roboti či bota, využívají předem definované algoritmy a pravidla k automatickému rozhodování o vstupu, výstupu a správě obchodů na finančních trzích (Janáč, 2016).

Klíčové charakteristiky (Janáč, 2016):

- Algoritmický přístup: AOS pracují na základě přesně definovaných algoritmů a pravidel, což umožňuje systematický a objektivní přístup k obchodování.
- Rychlost a efektivita: Díky automatizaci jsou AOS schopny reagovat na tržní události okamžitě, což umožňuje rychlé provedení obchodů bez emocionálních vlivů.
- Backtesting a optimalizace: Před spuštěním do reálného obchodování mohou AOS projít backtestingem, což umožňuje odhadnout výkonnost strategie na historických datech a optimalizovat ji pro aktuální podmínky.
- Diverzifikace a správa rizik: AOS mohou být navrženy tak, aby obchodovaly s různými nástroji nebo strategiemi, což umožňuje diverzifikaci portfolia a lepší správu rizik.
- 24/5 Obchodování: Systémy mohou provádět obchody nepřetržitě, což je klíčový prvek v dnešním globálním finančním prostředí s různými časovými pásmy.

Etapy vytváření AOS (Janáč, 2016)

1. Vytvoření strategie, vstupů a výstupů.
2. Naprogramování strategie.
3. Testování a odhalování chyb.
4. Hledání ziskového nastavení.
5. Testy robustnosti (není nutné).
6. Spuštění AOS na demo účtu, kontrola zisku a případné úpravy nastavení.
7. Spuštění AOS na reálném účtu s minimální velikostí lotu (standardizovaná jednotka měření velikosti obchodu), kontrola zisku a případné úpravy nastavení.
8. Navýšení velikosti lotu na původně plánovanou hodnotu.
9. Po určité době zkontrolujeme ziskovost, pokud se liší od backtestů, zkusíme optimalizovat nastavení.

Hodnocení a optimalizace výkonnosti obchodního bota

Efektivní hodnocení a pečlivá optimalizace jsou klíčovými kroky v procesu vývoje úspěšných algoritmických obchodních strategií. Ukážeme si metody a postupy, které nám umožňují systematicky analyzovat výkonnost obchodního bota a zdokonalit jeho chování (Janáč, 2016).

Metody Hodnocení Výkonnosti (Janáč, 2016):

- Zpětné testování: neboli backtesting, představuje klíčový nástroj pro posouzení, jak by strategie fungovala na historických datech. To umožňuje analyzovat výsledky v různých tržních podmínkách a identifikovat silné i slabé stránky strategie. Během zpětného testování se sledují klíčové metriky, včetně ziskovosti, abychom získali komplexní pohled na strategii.
- Simulované obchodování: často označované jako "paper trading", představuje prostředek pro ověření strategie v reálném čase bez skutečné finanční expozice. Metoda umožňuje testovat strategii na aktuálních tržních podmínkách a zjišťovat, jak by se chovala bez rizika finančních ztrát.
- Validace na reálných tržních datech: Abychom získali realistický obraz o výkonnosti obchodního bota, je důležité provádět validaci na aktuálních tržních datech. Fáze hodnocení zahrnuje porovnání výsledků strategie na historických datech s jejím skutečným chováním na trhu. Tímto způsobem lze odhalit rozdíly mezi historickými a reálnými tržními podmínkami.

Optimalizační Postupy (Janáč, 2016):

- Genetické algoritmy: představují sofistikovaný přístup k hledání optimálních parametrů strategie. Inspirují se procesy evoluce a umožňují systematicky procházet prostor parametrů za účelem nalezení nejlepších kombinací.
- Grid search a Random search: jsou jednoduché, avšak efektivní metody pro systematické prohledávání prostoru parametrů. Grid search systematicky zkoumá kombinace parametrů v daných intervalech, zatímco random search náhodně vybírá sady parametrů.
- Evoluční algoritmy: jsou komplexním přístupem k optimalizaci strategií. Tyto algoritmy využívají principy přirozeného výběru k vytváření nových generací strategií s cílem dosáhnout optimálních výsledků.

1.5 Money Management

Money management je klíčovým prvkem, který pečlivě řídí alokaci kapitálu a ovlivňuje způsob, jakým obchodní bot zachází s finančními prostředky. Efektivní money management je klíčovým prvkem v minimalizaci rizika a dlouhodobé udržitelnosti obchodní strategie. V kontextu popsání obchodní strategie pro skalpování na jednominutovém grafu jsou následující aspekty money managementu zvláště významné (Stibor, 2023):

Velikost obchodu v procentech kapitálu

Rozhodujícím pravidlem je určení procentuální části celkového kapitálu, která bude investována do jednotlivých obchodů. Stanovení optimálního procenta se pohybuje obvykle mezi jedním až třemi procenty kapitálu na jeden obchod. Strategie zabraňuje příliš velkým ztrátám v případě neúspěšné série obchodů a zároveň udržuje flexibilitu pro další investice.

Dynamická adaptace

Velikost obchodu může být nastavena tak, aby se dynamicky přizpůsobovala aktuálnímu stavu kapitálu. To znamená, že s rostoucím kapitálem může být zvýšit velikost obchodu a naopak. Tento dynamický přístup maximalizuje využití kapitálu a umožňuje rychlou reakci na změny v obchodním prostředí.

Správné nastavení zastavení ztrát a uzavírání obchodů

Správná alokace kapitálu zahrnuje precizní umístění zastavení ztrát a cílů pro uzavírání obchodů. Zastavení ztrát by mělo reflektovat volatilitu trhu a strategii, zatímco cíle pro uzavírání obchodů by měly být definovány na základě aktuálních tržních podmínek. Tímto způsobem se money management stává ochranou proti nečekaným výkyvům na trhu.

Diverzifikace portfolia

Rozložení kapitálu do různých aktiv a strategií představuje klíčový prvek pro snížení celkového rizika. Diverzifikace portfolia umožňuje, aby ztráty v jednom směru byly vyváženy zisky v jiném. Strategie maximalizuje stabilitu a minimalizuje dopady případných nepříznivých událostí.

1.6 Skalpování

Obchodní strategie zaměřující se na krátkodobé obchody s cílem získat malé zisky z drobných pohybů na trhu. Vyžaduje nepřetržité sledování trhu, což může být pro člověka náročné. Skalpování je zvláště efektivní v obdobích vysoké tržní likvidity, kdy malé cenové pohyby nabízejí potenciál pro rychlé zisky. Skalpeři obvykle využívají jednominutové až pětiminutové grafy, přičemž každý obchod trvá od několika sekund do několika minut (Investopedia, 2022).

Klíčové charakteristiky (Investopedia, 2022):

- Rychlost a frekvence obchodů: Skalpování vyžaduje rychlé rozhodování a provedení obchodů, často s velmi krátkým trváním. Obchodníci provádějí mnoho obchodů během krátké doby s cílem vytěžit malé cenové změny.
- Krátké časové rámce: Skalpeři často pracují s krátkými časovými rámy, jako jsou 1 nebo 5 minut. Analyzují drobné pohyby na trhu a pokouší se využít i ty nejmenší cenové změny.
- Malé zisky, malé ztráty: Cílem skalpování není dosahovat velkých zisků z jednoho obchodu, ale spíše vytvářet malé zisky, které se mohou akumulovat během řady obchodů. Zároveň se pokouší minimalizovat ztráty.
- Silná psychická odolnost: Skalpování vyžaduje vysokou koncentraci, rychlé rozhodování a schopnost rychle reagovat na tržní změny. Obchodníci musí být psychicky odolní a schopni zvládat riziko.

1.7 Exponenciální klouzavý průměr

Klouzavé průměry jsou nejrozšířenějšími indikátory. Ukazují průměrnou hodnotu ceny aktiva za dané časové období. Oproti jednoduchému klouzavému průměru dává EMA větší váhu aktuálnějším cenám, což umožňuje rychlejší reakci na aktuální tržní podmínky. Křížení cenového grafu a EMA může poskytnout signál o změně trendu (Maverick, 2022).



Obr. 3: Zobrazení EMA v grafu

Zdroj: Vlastní zpracování

Klíčové charakteristiky (Maverick, 2022):

- Vážené hodnoty: EMA přiděluje větší váhu nedávným datům, což zajišťuje pružnou reakci na aktuální tržní podmínky.
- Identifikace trendů: Jelikož klade důraz na nejnovější data, EMA se často využívá k identifikaci směru trendu. Křížení EMA s cenovým grafem může posloužit jako signál změny trendu.
- Generování obchodních signálů: Dynamika EMA umožňuje generování obchodních signálů na základě křížení s cenovým grafem, což může být klíčové pro vstup a výstup z obchodů.
- Hladkost křivek: V porovnání s klasickým klouzavým průměrem je EMA hladší, což pomáhá odhalit skutečné trendy na trhu bez rušivého šumu.

1.8 Relative strength index

Oscilátor používaný v technické analýze k měření rychlosti a změny cenového pohybu. RSI poskytuje informace o překoupenosti nebo přeprodanosti aktiva, což může naznačovat potenciální změnu směru cenového trendu. Nejčastěji používaná délka tohoto období je 14 dní (Stibor, 2023).



Obr. 4: Zobrazení RSI pod grafem

Zdroj: Vlastní zpracování

Klíčové charakteristiky (Fernando, 2023):

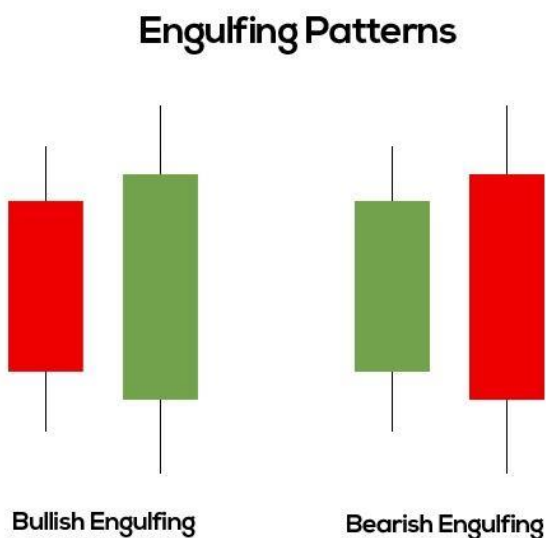
- Měření síly tržních pohybů: RSI měří relativní sílu ztrát a zisků během určitého období, obvykle 14 obchodních dnů. Na základě toho vytváří hodnotu RSI na stupnici od 0 do 100.
- Překoupené a přeprodané podmínky: RSI přiřazuje hodnoty nad 70 oblasti překoupených podmínek, což může naznačovat, že je aktivum pravděpodobně předražené. Hodnoty pod 30 zase signalizují přeprodané podmínky, kdy by cena mohla být podhodnocená.
- Generování obchodních signálů: Překročení hodnot 70 nebo 30 může být považováno za potenciální signál k vstupu nebo výstupu z obchodu. Například, když RSI klesne pod 30, může to naznačovat možné přeprodané podmínky a obrát cen nahoru.
- Divergence a konvergence: RSI může odhalovat divergence nebo konvergence s cenovým grafem. Tento jev může poskytnout další informace o síle nebo slabosti trendu.

1.9 Engulfing pattern

Engulfing Pattern je svíčková formace, která signalizuje možný obrát v cenovém trendu na finančních trzích. Formace se skládá ze dvou svíček, kde druhá svíčka zcela pohlcuje první, což naznačuje sílu a dominanci nového směru pohybu. Zaujímá klíčové místo mezi obchodnickými vzory a často slouží jako silný signál pro obchodníky při identifikaci možných změn v tržní dynamice. Vzor se dělí na dva hlavní typy: býčí (bullish) engulfing a medvědí (bearish) engulfing (Chen, 2022).

Klíčové charakteristiky (Chen, 2022):

- **Býčí engulfing:**
 - Vzor se objevuje na konci klesajícího trendu.
 - První svíčka je menší a má červené tělo, což ukazuje na pokračování klesajícího trendu.
 - Druhá svíčka je větší a má zelené tělo, které úplně pohltí tělo předchozí červené svíčky.
 - Signalizuje potenciální obrat trendu od klesajícího k vzestupnému.
- **Medvědí engulfing:**
 - Vzor se vyskytuje na konci vzestupného trendu.
 - První svíčka je menší a má zelené tělo, což naznačuje pokračování vzestupného trendu.
 - Druhá svíčka je větší a má červené tělo, které úplně pohltí tělo předchozí zelené svíčky.
 - Naznačuje možný obrat trendu od vzestupného trendu ke klesajícímu trendu.
- **Klíčové aspekty:**
 - Vzor má větší význam, když se objeví po dlouhém trendu, ať už vzestupném nebo klesajícím.
 - Vyšší obchodní objem během druhé svíčky může potvrdit sílu obrátového signálu.
 - Obchodníci často hledají další potvrzení obratu trendu v následujících svíčkách nebo pomocí dalších technických indikátorů.



Obr. 5: Engulfing Patterns

Zdroj: www.thinkmarkets.com(2023)

2 Obchodní strategie

Strategie navržená pro obchodního bota využívá metodu skalpování na krátkých časových rámcích. Klíčem k úspěchu je použití kombinace tří fundamentálních technických indikátorů: EMA s periodou 200, RSI s periodou 15 a Engulfing Patternu. Tato specifická sestava indikátorů poskytuje obchodnímu botu komplexní nástroj pro efektivní analýzu trhu a umožňuje mu identifikovat obchodní příležitosti s vysokou pravděpodobností zisku.

2.1 Vstup a výstup z obchodů

Klíčové aspekty rozhodovacího procesu, kterými bot určuje optimální okamžiky pro vstup a výstup z tržních pozic, jsou blíže popsány v této kapitole. Základem pro tyto rozhodnutí jsou algoritmy, které analyzují tržní data a identifikují signály indikující potenciální zisk. Detailní popis implementace těchto rozhodovacích algoritmů naleznete v následující kapitole, kde je rozvedena logika za použitím technických indikátorů jako jsou EMA, RSI a svíčkových formací pro generování obchodních signálů. Bot tedy vstoupí do obchodu za následujících podmínek:

Rostoucí trend

- Uzavření nad 200 EMA: Bot vstoupí do obchodu, pokud poslední svíčka uzavřela nad 200 EMA, což signalizuje sílu rostoucího trendu.
- RSI 14 nad 50 body: Přítomnost trendu je potvrzena, pokud RSI přesahuje hodnotu 50 bodů, což indikuje pozitivní momentum na trhu.
- Býčí Engulfing Pattern: Další potvrzení rostoucího trendu poskytuje býčí Engulfing Pattern, což je silný signál nástupu býčích sil na trh.

Klesající trend:

- Uzavření pod 200 EMA: V případě klesajícího trendu bot vstoupí do obchodu, pokud poslední svíčka uzavřela pod 200 EMA, což signalizuje sílu klesajícího trendu.
- RSI 14 pod 50 body: Přítomnost trendu je potvrzena, pokud RSI klesá pod hodnotu 50 bodů, což indikuje negativní momentum na trhu.
- Medvědí Engulfing Pattern: Další potvrzení klesajícího trendu poskytuje medvědí Engulfing Pattern, což je silný signál nástupu medvědí sil na trh.

Zastavení ztrát je nastaveno pod hranici vstupní svíčky při rostoucím trendu a nad hranici vstupní svíčky při klesajícím trendu. Uzavření obchodu se ziskem je nastaveno na dvojnásobek pipů (nejmenší možný pohyb ceny na forexovém trhu) zastavení ztrát, pokud je tedy zastavení ztrát nastaveno na -10 pipů, uzavření obchodu se ziskem je nastaveno na 20 pipů.

2.2 Správa financí

V rámci správy financí při automatizovaném obchodování se uživatelé mohou rozhodovat mezi dvěma základními strategiemi, které ovlivňují způsob, jakým jsou nastavovány objemy objednávek. První možností je využití fixního objemu objednávky pro každý obchod, zatímco druhou možností je aplikace procentuálního rizika vzhledem k zůstatku na účtu. Oba přístupy mají své specifické výhody a nevýhody, které je třeba pečlivě zvážit při volbě nejvhodnější strategie pro správu financí.

Fixní objem objednávky

Výhody:

- Jednoduchost a předvídatelnost: Implementace fixního objemu do obchodního bota zajišťuje jednoduchost a předvídatelnost objemů obchodů, což usnadňuje kontrolu rizik a alokaci kapitálu.
- Stabilita: Fixní objem zajistí konstantní velikost pozic bez ohledu na výkyvy trhu, což může botu pomoci udržet stabilitu v obchodní strategii.

Nevýhody:

- Omezená flexibilita: Použití fixního objemu nezohledňuje změny v kapitálu nebo v tržních podmínkách, což může vést k neoptimálnímu využití kapitálu a zvýšenému riziku. Potenciální
- Nadměrné riziko: V případě neočekávaných tržních pohybů může fixní objem vést k nadměrnému riziku, pokud není adekvátně nastaven vzhledem k aktuálním podmínkám.

Procentuální riziko zůstatku na účtu

Výhody:

- Automatická adaptace: Dynamicky nastavovaný objem na základě procentuálního rizika umožňuje botu automaticky upravovat velikost pozic v souladu s aktuální výší kapitálu, čímž maximalizuje efektivitu řízení rizik.
- Ochrana kapitálu: Strategie založená na procentuálním riziku přirozeně chrání kapitál tím, že snižuje objem obchodů v případě ztrát, čímž zabraňuje významným poklesům na účtu.

Nevýhody:

- Složitost implementace: Výpočet optimálního objemu obchodu na základě procentuálního rizika vyžaduje složitější algoritmy, což zvyšuje nároky na programování bota.
- Variabilita objemu: Pro obchodního bota může být náročnější reagovat na časté změny objemu obchodů, což vyžaduje pokročilé algoritmy pro správu objednávek a alokaci kapitálu.

Při spuštění obchodního bota na trh je klíčové zvážit, která z těchto strategií nejlépe vyhovuje obchodnímu modelu a jaké jsou očekávané tržní podmínky. Výběr správné strategie pro správu financí může výrazně přispět k úspěchu obchodního bota tím, že optimalizuje riziko a zvyšuje potenciální ziskovost.

3 Implementace

Kapitola se věnuje detailnímu popisu kódu obchodního bota. Proces implementace začíná definicí základního nastavení bota, včetně unikátních identifikátorů a obchodních parametrů, které umožňují jeho efektivní fungování v dynamickém prostředí finančních trhů.

Podrobně je rozebráno použití technických indikátorů, jako jsou exponenciální klouzavý průměr, relative strength index a svíčkového vzoru engulfing, které jsou základem pro generování obchodních signálů. Následuje zaměření na získání dat z posledních svíček, což je nezbytné pro analýzu cenových pohybů a identifikaci potenciálních obchodních příležitostí.

Velký důraz je kladen na správu pozic, kde jsou popsány mechanismy pro nastavení stop-loss a take-profit úrovní, což jsou klíčové komponenty pro efektivní řízení rizik. Je ukázáno, jak je možné dynamicky upravovat objem obchodů s ohledem na aktuální tržní podmínky a zbývající kapitál, čímž je zvýšena efektivita a bezpečnost obchodní strategie.

3.1 Základní architektura

Hlavní soubor obchodního bota obsahuje definici obchodní strategie, inicializaci obchodních indikátorů, řízení rizik, a nastavení obchodních časů.

Input

Sekce Input slouží jako rozhraní pro uživatele, umožňující nastavení parametrů bota před jeho spuštěním na obchodní platformě. Segment Input je základem pro personalizaci bota umožňující uživatelům doladění jeho chování bez nutnosti zasahovat přímo do programového kódu.

```
//+-----+
//| Input |
//+-----+
sinput group "EA GENERAL SETTINGS"
input ulong MagicNumber = 101;
input int Deviation = 30;
input ENUM_ORDER_TYPE_FILLING FillingPolicy = ORDER_FILLING_FOK;

sinput group "INDICATOR SETTINGS"
input int MAPeriod = 200;
input int RSI_Period = 14;

sinput group "RISK MANAGEMENT"
input double FixedVolume = 0;
input double RiskPercent = 3;

sinput group "TIMER SETTINGS"
input uchar StartHour = 0;
input uchar StartMinute = 0;
input uchar EndHour = 0;
input uchar EndMinute = 0;
```

Kód nabízí uživateli pokročilé možnosti pro přizpůsobení. Jeho hlavním cílem je umožnit detailní konfiguraci EA podle individuálních obchodních preferencí. Rozdělení konfiguračních parametrů do specifických skupin usnadňuje orientaci v nastaveních a jejich úpravu.

EA General Settings

- **MagicNumber:** Unikátní identifikátor pro EA, aby rozlišil své obchody od obchodů jiných EA.
- **Deviation:** Maximální povolená odchylka od požadované ceny při zadávání obchodů.
- **FillingPolicy:** Politika vyplnění obchodů, např. ORDER_FILLING_FOK znamená, že obchod musí být vyplněn celý nebo vůbec.

Indicator Settings

- **MAPeriod:** Doba použitá pro výpočet klouzavého průměru.
- **RSI_Period:** Doba použitá pro výpočet indexu relativní síly.

Risk Management

- **FixedVolume:** Pevně stanovený objem pro obchodování, pokud není použito procentuální riziko.
- **RiskPercent:** Procento kapitálu, které může být riskováno v každém obchodu.

Timer Settings

- **StartHour a StartMinute:** Hodina a minuta začátku obchodního okna.
- **EndHour a EndMinute:** Hodina a minuta konce obchodního okna.

OnInit

Funkce OnInit je volána při inicializaci bota na obchodní platformě. Zde se provádí nastavení obchodních parametrů, jako jsou typy vyplnění příkazů, povolené odchylky a magické číslo, stejně jako inicializace technických indikátorů, které budou použity v obchodní strategii. Funkce zajišťuje, že všechny nezbytné komponenty jsou správně nastaveny a připraveny. V případě, že inicializace selže nebo pokud jsou indikátory nesprávně nastaveny, OnInit vrátí chybový kód, který zastaví další spuštění EA, čímž předchází potenciálním problémům během obchodování.

```
//+-----+
//| Expert initialization function |
//+-----+
Trade.SetFillingType(FillingPolicy);
Trade.SetDeviation(Deviation);
Trade.SetMagicNumber(MagicNumber);

glTimeBarOpen = D'1971.01.01 00:00';

int EMAHandle = EMA.Init(_Symbol,PERIOD_CURRENT,MAPeriod);
if(EMAHandle == -1){return(INIT_FAILED);}

int RSIHandle = RSI.Init(_Symbol,PERIOD_CURRENT,RSI_Period);
if(RSIHandle == -1){return(INIT_FAILED);}

return(INIT_SUCCEEDED);
```

Inicializace handle pro EMA a RSI indikátor, díky kterému můžeme později přistupovat k výsledkům. Pokud inicializace selže, funkce ji ukončí a vrátí INIT_FAILED. Pokud vše proběhlo bez problémů, funkce vrací INIT_SUCCEEDED, označující, že bot je připraven k obchodování.

OnTick

Hlavní logika obchodování se nachází ve funkci OnTick, která je aktivována při každé změně ceny (novém "ticku") na trhu. Funkce neustále monitoruje tržní podmínky a rozhoduje o vstupu či výstupu z obchodů na základě definovaných obchodních pravidel a signálů technických indikátorů. V OnTick se provádí analýza tržních dat, jako jsou ceny uzavření a hodnoty indikátorů, a na základě této analýzy se umísťují nebo upravují obchodní příkazy. Funkce je nezbytná pro realizaci obchodní strategie v reálném čase reagující na tržní změny s vysokou rychlostí a přesností.

```
//-----//
// NEW BAR CONTROL //
//-----//

bool newBar = false;

//Check for New Bar
if(glTimeBarOpen != iTime(Symbol(), PERIOD_CURRENT, 0))
{
    newBar = true;
    glTimeBarOpen = iTime(Symbol(), PERIOD_CURRENT, 0);
}
```

Kontrola nové svíčky: Podmínka ověřuje, zda se čas otevření poslední zaznamenané svíčky glTimeBarOpen liší od času otevření aktuální svíčky na trhu.

```
if(newBar == true)
{
    DelayOnMarketClose(_Period);

    //-----//
    // PRICE & INDICATORS //
    //-----//

    //Price
    Bar.Refresh(_Symbol, PERIOD_CURRENT, 3);
    double close1 = Bar.Close(1);
    double close2 = Bar.Close(2);

    //Normalization of close price to tick size
    double tickSize=
    SymbolInfoDouble(_Symbol, SYMBOL_TRADE_TICK_SIZE);
    close1 = round(close1/tickSize) * tickSize;
    close2 = round(close2/tickSize) * tickSize;

    //EMA
    EMA.RefreshMain();
    double ema1= EMA.main[1];

    //RSI
    RSI.RefreshMain();
    double rs1=RSI.main[1];
```

```
//ENGULFING
bool bullEng =
ENG.IsBullishEngulfing(_Symbol, PERIOD_CURRENT);
bool bearEng =
ENG.IsBearishEngulfing(_Symbol, PERIOD_CURRENT);
```

Funkce DelayOnMarketClose kontroluje, zda trh není zavřený. Do proměnných close1 a close2 se ukládá cena uzavření posledních dvou svíček. Aby byly ceny uzavření kompatibilní s minimálním pohybem ceny (tickem) na trhu, jsou normalizovány podle velikosti ticku. Do ema1 a rsi1 je získána hodnota z poslední svíčky. Funkce IsBullishEngulfing a IsBearishEngulfing zjišťují, zda aktuální tržní data odpovídají vzoru býčího nebo medvědího engulfinu.

```
//-----//
// TRADE PLACEMENT //
//-----//
```

```
//Entry Signals && Order Placement Execution
String entrySignal=
ENentrySignal(ema1,rsi1,bullEng,bearEng,close1);
Comment("EA #", MagicNumber, " | ",entrySignal, " SIGNALS
DETECTED");
```

```
bool timer =
GetTimerSignal(StartHour,StartMinute,EndHour,EndMinute);
```

Funkce Enentrysignal získává vstupní signál (LONG, SHORT, nebo NO_TRADE) na základě kombinace hodnot EMA, RSI, vzorů engulfinu a ceny uzavření. Funkce GetTimerSignal ověřuje, zda aktuální čas spadá do povoleného obchodního okna. Tím se zajišťuje, že obchody se provádějí pouze v určených hodinách.

```
if(Trade.SelectPosition(_Symbol)==false && timer && (entrySignal
== "LONG" || entrySignal == "SHORT"))
{
    double stopLoss =PM.CalculateStopLoss(_Symbol, entrySignal,
PERIOD_CURRENT,close2);
    double takeProfit =
PM.CalculateTakeProfit(_Symbol,entrySignal,PERIOD_CURRENT,
stopLoss, close1);
    double volume =
PM.RiskManagement(_Symbol,FixedVolume,RiskPercent,
MathAbs(stopLoss - close1));
    Print("Volume",volume);
    ulong ticket = 0;

    if(entrySignal == "LONG") ticket =
Trade.Buy(_Symbol,volume);
    else if(entrySignal == "SHORT") ticket =
Trade.Sell(_Symbol,volume);

    //SL & TP Trade Modification
    if(ticket > 0)
    {
        Print("sl ",stopLoss,"tp ",takeProfit);
        Trade.ModifyPosition(_Symbol,ticket,stopLoss,takeProfit,
entrySignal);
    }
}
```

Bot kontroluje, zda aktuálně nevlastní pozici pro daný symbol, časová kontrola je v souladu s obchodními hodinami a byl detekován vstupní signál pro LONG nebo SHORT pozici. Jsou vypočítány hodnoty stop-loss a take-profit na základě aktuálních tržních podmínek a definované obchodní strategie. Metoda pro výpočet objemu obchodu je určena podle inputu uživatele. Na základě vstupního signálu bot umístí buď nákupní (LONG) nebo prodejní (SHORT) příkaz. Pokud byl obchod úspěšně umístěn (ticket > 0), bot modifikuje stop-loss a take-profit hodnoty pro danou pozici.

3.2 Technické indikátory

Obsahuje implementaci technických indikátorů, jako jsou EMA a RSI. Kód vytváří základní třídy pro indikátory, umožňující snadné použití a aktualizaci hodnot indikátorů pro obchodní rozhodnutí.

```
//+-----+
//|Base class|
//+-----+
class CIndicator
{
protected:
    int      handle;

public:
    double   main[];

    CIndicator(void);

    virtual int Init() {return(handle); }
    void      RefreshMain(void);
};

CIndicator::CIndicator(void)
{
    ArraySetAsSeries(main,true);
}

void CIndicator::RefreshMain(void)
{
    ResetLastError();

    if(CopyBuffer(handle,0,0,VALUES_TO_COPY,main) < 0)
        Print("FILL_ERROR",GetLastError());
}
```

Třída CIndicator je šablonou pro vytváření indikátorů. Virtuální metoda Init() vrací prázdný handle, pokud není v potomcích třídy definován jinak. Konstruktor inicializuje pole main[] jako sérii, což znamená, že nejnovější data budou na indexu 0. RefreshMain() aktualizuje data v main[] pomocí CopyBuffer, zpracovává chyby při kopírování. Třída umožňuje snadné rozšíření pro specifické indikátory přepsáním metody Init a přidáním dalších metod pro zpracování dat indikátorů.

```
//+-----+
//|                                     EMA                                     |
//+-----+
class CEMA : public CIndicator
{
public:
    // Constructor
    CEMA(void) : CIndicator() {}

    // Overridden Init method for EMA initialization
    virtual int Init(string symbol, ENUM_TIMEFRAMES timeframe,
        int period)
    {
        handle = iMA(symbol, timeframe, period, 0, MODE_EMA,
            PRICE_CLOSE);
        if(handle == INVALID_HANDLE)
        {
            Print("Cannot initialize EMA handle for period ",
                period);
            return(INIT_FAILED);
        }
        Print("EMA handle initialized");
        return(handle);
    }
};
```

Třída CEMA, která je potomkem třídy CIndicator, slouží k inicializaci EMA. Metoda Init pak využívá vestavěnou funkci iMA k získání handle pro EMA. Pokud inicializace proběhne úspěšně, metoda vrátí handle indikátoru. V případě selhání vypíše chybovou zprávu a signalizuje chybu.

```
//+-----+
//|                                     RSI                                     |
//+-----+
class CRSI : public CIndicator
{
public:
    // Constructor
    CRSI(void) : CIndicator() {}

    // Overridden Init method for RSI initialization
    virtual int Init(string symbol, ENUM_TIMEFRAMES timeframe,
        int period)
    {
        handle = iRSI(symbol, timeframe, period, PRICE_CLOSE);
        if(handle == INVALID_HANDLE)
        {
            Print("Cannot initialize RSI handle for period ",
                period);
            return(INIT_FAILED);
        }
        Print("RSI handle initialized");
        return(handle);
    }
};
```

Třída CRSI slouží k inicializaci RSI. Funguje na stejném principu jako CEMA.

```
//+-----+
//|                                     Engulfing                                     |
//+-----+
class CENG : public CIndicator{

public:
    CENG(void):CIndicator(){}

    bool IsBullishEngulfing(string symbol,
        ENUM_TIMEFRAMES timeframe) {
        double prevOpen[1], prevClose[1], currentOpen[1],
        currentClose[1];
        // Loading open and close prices for the previous candle
        if(CopyOpen(symbol, timeframe, 2, 1, prevOpen) > 0 &&
            CopyClose(symbol, timeframe, 2, 1, prevClose) > 0) {
            // Loading open and close prices for the current
            candle
            if(CopyOpen(symbol, timeframe, 1, 1, currentOpen) > 0
                &&CopyClose(symbol, timeframe, 1, 1, currentClose)>0)
            {
                // Checking conditions for Bullish Engulfing
                if(prevClose[0] < prevOpen[0] && currentClose[0]
                    currentOpen[0] && currentOpen[0] < prevClose[0]
                    && currentClose[0] > prevOpen[0]) {
                    return true;
                    // Bullish Engulfing Pattern detected
                }
            }
        }
        return false; // Bullish Engulfing Pattern not detected
    }

    bool IsBearishEngulfing(string symbol,
        ENUM_TIMEFRAMES timeframe) {
        double prevOpen[1], prevClose[1], currentOpen[1],
        currentClose[1];
        // Loading open and close prices for the previous candle
        if(CopyOpen(symbol, timeframe, 2, 1, prevOpen) > 0 &&
            CopyClose(symbol, timeframe, 2, 1, prevClose) > 0) {
            // Loading open and close prices for the current
            candle
            if(CopyOpen(symbol, timeframe, 1, 1, currentOpen) > 0
                &&CopyClose(symbol,timeframe, 1, 1, currentClose)> 0)
            {
                // Checking conditions for Bearish Engulfing
                if(prevClose[0] > prevOpen[0]&& currentClose[0] <
                    currentOpen[0] && currentOpen[0]> prevClose[0] &&
                    currentClose[0] < prevOpen[0]) {
                    return true;
                    // Bearish Engulfing Pattern detected
                }
            }
        }
        return false; // Bearish Engulfing Pattern not detected
    }
};
```

Třída CENG, odvozená od CIndicator, je navržena pro detekci vzorů Bullish a Bearish Engulfing vzoru. Metody získávají data o předchozích dvou svíčkách, které poté porovnávají. V případě potvrzení jednoho ze vzorů je příslušné metody vráceno true.

```
string ENentrySignal(double ema, double rsi, bool bullEng,
                    bool bearEng, double close1)
{
    string str = "";

    if(close1>ema && rsi>50 && bullEng==true){str = "LONG";}
    else if(close1<ema && rsi<50 && bearEng==true)
        {str = "SHORT";}
    else {str = "NO TRADE";}

    if(str=="LONG" || str=="SHORT")
    {
        Print(str, " SIGNAL DETECTED");
    }
    return str;
}
```

Funkce ENentrySignal je navržena k určení obchodního signálu na základě kombinace technických indikátorů a svíčkových vzorů. Rozhoduje o vstupu do dlouhé (LONG), krátké (SHORT) pozice, nebo žádné (NO_TRADE) pozice. Podmínky pro vstupní signál jsou vysvětlené v 2. kapitole.

3.3 Získání dat ze svíček

Třída CBar umožňuje efektivní přístup k datům o svíčkách. Konstruktor třídy inicializuje pole pro práci s daty jako s časovou řadou, a metoda Refresh umožňuje načíst specifický počet svíček pro daný symbol a časový rámec. Metody jako Time, Open, High, Low, Close, TickVolume, Spread a Volume pak poskytují rychlý přístup k jednotlivým údajům.

```
//+-----+
//|      CBar Class - Bar Data(OHLC, Time, Volume, Spread)
//+-----+

class CBar
{
public:
    MqlRates    bar[];

    CBar(void);
    void Refresh(string pSymbol,
        ENUM TIMEFRAMES pTimeframe,
        int pBarsToRefresh);

    datetime    Time(int pShift)
        {return(bar[pShift].time);}
    double      Open(int pShift)
        {return(bar[pShift].open);}
    double      High(int pShift)
        {return(bar[pShift].high);}
    double      Low(int pShift)
```

```

        {return(bar[pShift].low);}
double    Close(int pShift)
        {return(bar[pShift].close); }
long      TickVolume(int pShift)
        {return(bar[pShift].tick_volume);}
int        Spread(int pShift)
        {return(bar[pShift].spread);}
long      Volume(int pShift)
        {return(bar[pShift].real_volume);}

};

CBar::CBar(void)
{
    ArraySetAsSeries(bar,true);
}

void CBar::Refresh(string pSymbol, ENUM_TIMEFRAMES pTimeframe,
int pBarsToRefresh)
{
    CopyRates(pSymbol,pTimeframe,0,pBarsToRefresh,bar);
}

```

3.4 Správa obchodních operací

Soubor obsahuje metody související s řízením obchodů, včetně kontroly obchodních časů pomocí funkce GetTimerSignal a odložení obchodu při uzavření trhu metodou DelayOnMarketClose. Tyto funkce zajišťují, že obchodní operace probíhají v optimálních obchodních hodinách a podmínkách.

```

//Trading timer
bool GetTimerSignal(uchar pStartHour, uchar pStartMinute, uchar
pEndHour, uchar pEndMinute)
{
    //if all argument are zero, we are not using the timer therefore
    we return true
    if(pStartHour==0 && pStartMinute==0 && pEndHour==0 &&
        pEndMinute==0) return(true);

    //current time
    MqlDateTime time;
    TimeCurrent(time);

    //convert to minutes
    int startTime = pStartHour*60+pStartMinute;
    int endTime = pEndHour*60+pEndMinute;
    int currentTime = time.hour*60+time.min;

    //CASE 1:18-21h
    if((startTime<endTime && currentTime>=startTime &&
        currentTime<=endTime) ||

    //CASE 2:18-3h
    (startTime>endTime && (currentTime>=startTime ||

```

```

        currentTime <= endTime))) {

    return(true); }

    return(false);
}

```

Funkce GetTimerSignal v určuje, zda aktuální čas spadá do nastaveného obchodního intervalu. Pokud jsou všechny vstupní parametry nulové, je obchodování povoleno kdykoliv. Funkce porovnává aktuální čas s definovaným začátkem a koncem obchodního intervalu, přičemž zohledňuje situace, kdy obchodní interval přesahuje půlnoc (například obchodování od 18:00 do 3:00). Vrací true pro časy spadající do nastaveného intervalu, což umožňuje realizovat transakce a false v opačném případě.

```

// Delay program execution when market is closed
void DelayOnMarketClose(ENUM_TIMEFRAMES pTimeframe)
{
    //current time
    MqlDateTime time;
    TimeCurrent(time);

    if(MQLInfoInteger(MQL_TESTER) && time.hour==0)
    {
        if(pTimeframe >=PERIOD_H4) Sleep(1800000); //30 min
        else Sleep(120000); //2 min
    }
}

```

Funkce DelayOnMarketClose odloží provádění programu v testovacím prostředí, pokud je detekováno potenciální uzavření trhu (když je aktuální hodina 0). Zpoždění je nastaveno na 30 minut pro časové rámce 4 hodiny a delší, a na 2 minuty pro kratší časové rámce, čímž simuluje pauzy v obchodování a pomáhá zajistit realističtější výsledky při backtestingu strategií.

3.5 Řízení obchodních pozic

Poskytuje metody pro výpočet stop-loss a take-profit hodnot, správu rizika a určení objemu obchodu na základě definované strategie a aktuálních tržních podmínek. Umožňuje flexibilní správu pozic s ohledem na riziko a potenciální zisk.

```

class CPM
{
public:
    MqlTradeRequest request;
    MqlTradeResult result;

    CPM(void);

    double CalculateStopLoss(string pSymbol, string
pEntrySignal, ENUM_TIMEFRAMES timeframe, double close2);

    double CalculateTakeProfit(string pSymbol, string
pEntrySignal, ENUM_TIMEFRAMES timeframe, double
stopLossPrice, double close1);
}

```

```
double RiskManagement(string pSymbol, double pFixedVol,
double pRiskPerc, double pSLInPricePoints);

double CalculateVolumeRiskPerc(string pSymbol, double
pRiskPerc, double pSLInPricePoints);

double VerifyVolume(string pSymbol, double pVolume);

};
```

Třída CPM je navržena k poskytování funkcí pro řízení obchodů.

- MqlTradeRequest request: Struktura pro požadavky na obchod, jako jsou objednávky a obchodní operace.
- MqlTradeResult result: Struktura pro výsledky obchodních operací, včetně informací o úspěchu či neúspěchu a chybových kódů.

```
//+-----+
//|      CPM Class Methods
//+-----+
```

```
CPM::CPM(void)
{
    ZeroMemory(request);
    ZeroMemory(result);
}
```

Konstruktor třídy CPM inicializuje proměnné request a result na nulové hodnoty pomocí funkce ZeroMemory. Funkce efektivně vymaže obsah těchto struktur.

```
double CPM::CalculateStopLoss(string pSymbol, string
pEntrySignal, ENUM_TIMEFRAMES timeframe, double close2)
{
    double stopLoss = 0.0;
    double tickSize = SymbolInfoDouble(pSymbol,
                                        SYMBOL_TRADE_TICK_SIZE);
    double pointSize= SymbolInfoDouble(pSymbol,SYMBOL_POINT);

    stopLoss = close2;

    stopLoss = round(stopLoss/tickSize) * tickSize;
    Print("SL: ",stopLoss);
    return stopLoss;
}
```

Metoda CalculateStopLoss slouží k výpočtu hodnoty stop-loss pro obchod. Hodnota stop-loss je nastavena na cenu zavření druhé nejnovější svíčky (close2). Poté je normalizována (zaokrouhlena) na nejbližší možnou hodnotu, která je kompatibilní s velikostí ticku pro daný symbol.

```
double CPM::CalculateTakeProfit(string pSymbol, string
pEntrySignal, ENUM_TIMEFRAMES timeframe, double stopLossPrice,
double close1)
{
    double takeProfit = 0.0;
    double tickSize =
    SymbolInfoDouble(pSymbol,SYMBOL_TRADE_TICK_SIZE);
```

```

double slDistance = 0.0;

// Calculate the SL distance from the opening price in pips
if(pEntrySignal == "LONG")
{
    slDistance = (close1 - stopLossPrice);
    takeProfit = close1 + (slDistance * 2);
}
else if(pEntrySignal == "SHORT")
{
    slDistance = (stopLossPrice - close1);
    takeProfit = close1 - (slDistance * 2);
}
else
{
    Print("Error loading entry signal.");
}

// Rounding TP to a valid number of decimal places
takeProfit = round(takeProfit/tickSize) * tickSize;
Print("TP: ",takeProfit);
return takeProfit;
}

```

Metoda CalculateTakeProfit slouží k výpočtu hodnoty takeProfit pro obchod. Pokud je vstupní signál LONG, je hodnota stopLossPrice odečtena od hodnoty zavírací ceny svíčky (close1). Následně se nastaví take-profit na dvojnásobek této vzdálenosti nad zavírací cenou. Pro signál SHORT je výpočet prováděn opačně. Hodnota takeProfit je před odesláním normalizována. V případě chyby je vypsána chybová hláška.

```

//+-----+
//|      Risk management
//+-----+

double CPM::RiskManagement(string pSymbol,double pFixedVol,
                           double pRiskPerc,double pSLInPricePoints)
{
    double volume = 0;

    if(pFixedVol > 0)        volume = pFixedVol;
    else if(pRiskPerc > 0)
        volume=CalculateVolumeRiskPerc(pSymbol,
        pRiskPerc,pSLInPricePoints);

    return VerifyVolume(pSymbol,volume);
}

```

Metoda RiskManagement umožňuje flexibilní správu rizika tím, že nechá uživatele volit mezi pevným objemem obchodu a objemem určeným podle procentuálního rizika celkového kapitálu na obchodním účtu. Pokud je pFixedVol větší než 0, znamená to, že uživatel specifikoval pevný objem obchodu, který se má použít bez ohledu na riziko. Tento objem je přímo přiřazen proměnné volume. Pokud je pFixedVol rovno 0 a pRiskPerc je větší než 0, použije se metoda CalculateVolumeRiskPerc k výpočtu objemu obchodu. Na závěr je zavolána metoda VerifyVolume, která kontroluje, zda je objem v souladu s minimálními a maximálními limitními hodnotami stanovenými pro daný obchodní nástroj na trhu, a případně objem upraví, aby odpovídal těmto požadavkům.

```
double CPM::CalculateVolumeRiskPerc(string pSymbol,
                                   double pRiskPercent, double pSLInPricePoints)
{
    //Maximum monetary amount of our equity that we are willing to
    //lose in a single trade
    double maxRisk =
        pRiskPercent * 0.01 * AccountInfoDouble(ACCOUNT EQUITY);

    //Value of 1 tick movement for 1 LOT in your account deposit
    //currency
    Double tickValue=
        SymbolInfoDouble(pSymbol, SYMBOL_TRADE_TICK_VALUE);

    //max risk amount divided by SL in points - it gives the money
    //risked per point
    double riskPerPoint =
        maxRisk/(pSLInPricePoints/SymbolInfoDouble(pSymbol,
        SYMBOL_POINT));

    //Money that we risk per point divided by the value of 1 tick
    //movement will give the lots required to hit the desired loss per
    //point
    double lotsRisked = riskPerPoint / tickValue;

    Print("SL in pips: ", pSLInPricePoints, ", Max risk: ", maxRisk,
        ", Tick value: ", tickValue, ", risk Per Point: ", riskPerPoint, "
        lots risked: ", lotsRisked);

    return (lotsRisked);
}
```

Metoda CalculateVolumeRiskPerc počítá objem obchodu v lotech na základě procentuálního rizika z celkového kapitálu. Nejprve určí maximální částku, kterou obchodník riskuje na základě jeho rizikové preference, a pak vypočítá, kolik peněz obchodník riskuje na každý pip. Vydělením této hodnoty hodnotou pohybu ticku pro 1 lot dostane potřebný objem obchodu. Na závěr vypíše důležité informace a navrátí lotsRisked.

```

//Verify and adjust the volume - so it is the minimum required
ammount
double CPM::VerifyVolume(string pSymbol,double pVolume)
{
double minVolume = SymbolInfoDouble(pSymbol, SYMBOL_VOLUME_MIN);
double maxVolume = SymbolInfoDouble(pSymbol, SYMBOL_VOLUME_MAX);
double          stepVolume          =          SymbolInfoDouble(pSymbol,
SYMBOL_VOLUME_STEP);
double initialMarginRate, maintenanceMarginRate;

// Retrieve margin rates for the symbol
SymbolInfoMarginRate(pSymbol, ORDER_TYPE_BUY, initialMarginRate,
maintenanceMarginRate);

double          contractSize          =          SymbolInfoDouble(pSymbol,
SYMBOL_TRADE_CONTRACT_SIZE);
// Calculate margin required for one lot
double marginPerLot = contractSize * initialMarginRate;
double freeMargin = AccountInfoDouble(ACCOUNT_MARGIN_FREE);
// Calculate the maximum possible volume based on the free margin
and margin per lot
double  maxPossibleVolume  =  freeMargin /  (marginPerLot  *
SymbolInfoDouble(pSymbol, SYMBOL_VOLUME_MIN));
maxPossibleVolume=maxPossibleVolume/100;
maxPossibleVolume=MathFloor(maxPossibleVolume/stepVolume)*stepVo
lume;

double verifiedVol;
if(pVolume>maxPossibleVolume) verifiedVol = maxPossibleVolume;
else if(pVolume < minVolume)  verifiedVol = minVolume;
else if(pVolume > maxVolume)  verifiedVol = maxVolume;
else                          verifiedVol = MathRound(pVolume /
stepVolume) * stepVolume;

Print("Adjusted Volume: ", verifiedVol, ", Free Margin: ",
freeMargin, ", Margin Per Lot: ", marginPerLot, ", Initial Margin
Rate: ", initialMarginRate, "contractSize: ",contractSize,"
maxPossibleVolume: ",maxPossibleVolume);

return(verifiedVol);
}

```

Metoda VerifyVolume ověřuje a případně upravuje objem obchodu, aby odpovídal minimálním a maximálním limitům stanoveným pro daný symbol a zároveň nepřesahoval finanční možnosti účtu. Nejdříve je načten minimální, maximální objem obchodu a minimální objem pro navýšení obchodu pro daný symbol. Poté získává maržové sazby pro symbol, počítá marži potřebnou pro jeden lot a upravuje a normalizuje maximální možný objem. Nakonec porovnává původní objem s limity objemu.

4 Testování

Kapitola představuje klíčovou fázi v procesu vývoje obchodního bota, kde se teoretické principy a implementační strategie setkávají s realitou finančních trhů. V této kapitole se zaměřujeme na komplexní testování funkčnosti, výkonnosti a odolnosti obchodního bota v různých tržních podmínkách. Simulace se odehrává na demo účtu následně testují bota v prostředí, které napodobuje reálné tržní podmínky, ale bez rizika finanční ztráty. Tento přístup nabízí možnost identifikovat slabé stránky ve strategii a provést potřebné úpravy pro zlepšení celkové výkonnosti bota.

Demo účet typu Razor byl založen u makléře PepperStone. Účet typu Razor je vhodnější pro skalpovací strategie, jelikož poskytuje malý rozdíl otevíracích a zavíracích cen. Cílem je ověřit, zda bot splňuje předem stanovené cíle obchodní strategie, a zda je schopen konzistentně generovat zisk, zatímco efektivně řídí rizika.

U následujících testů byla data svíček byla brána z období 1. 6. 2023 až 1. 1. 2024 na instrumentu EURO/USD. Počáteční kapitál činil 10 000 EURO s fixně nastaveným objemem obchodu na 0,01 lotu.

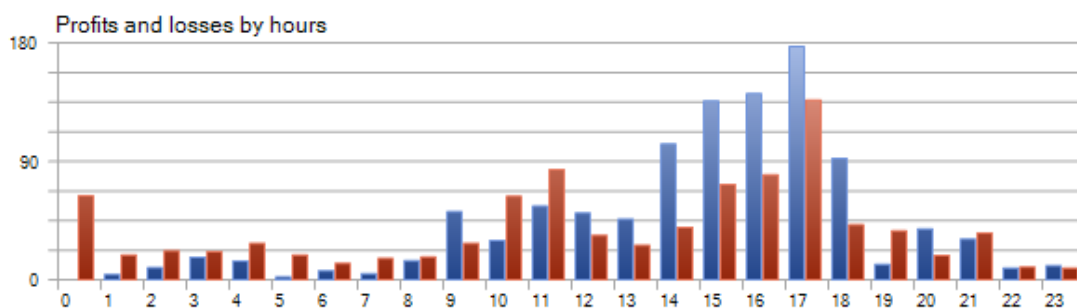
Nejprve tedy bota otestujeme na významných časových rámcích:

Tab. 1: Výsledky prvního testování

Časový rámec	M1	M5	M15	M30	H1	H2
Celkový výdělek v eurech	-792,86	10,95	-14,64	-101,98	47,04	47,04

Zdroj: Vlastní zpracování

Po dokončení prvního kola testování se ukázalo, že dosažené výsledky nesplňují očekávání. Analýzou přehledu jednotlivých obchodů byly identifikovány časové intervaly, během kterých docházelo k nejvyššímu počtu ztrátových transakcí. Tato zjištění umožňuje úpravu strategie tak, aby byly obchody realizovány pouze v těch časových intervalech, které přinášely lepší výsledky.



Obr. 5: Úspěšnost obchodů v různých hodinách pro časový rámec M5

Zdroj: Vlastní zpracování

V rámci druhého kola testování byly tedy upraveny obchodní intervaly pro jednotlivé časové rámce a vynechány ty, které se ani po úpravě neprojeví jako ziskové.

Tab. 2: Výsledky druhého testování

Časový rámec	M5	M15	M30
Obchodní intervaly	12:00-18:00	12:00-19:00	15:00-18:00
Celkový výdělek v eurech	175,45	76,86	78,14

Zdroj: Vlastní zpracování

Je zřetelné, že provedené změny vedly k výraznému omezení neziskových obchodů. Tyto výsledky naznačují potenciální úspěch na trhu. V následujícím kroku se zaměříme na aplikaci strategie založené na procentuálním riziku (RiskPercent), což nám umožní ještě lépe spravovat a minimalizovat rizika spojená s obchodními operacemi.

Tab. 3: Výsledky třetího testování

Časový rámec	M5	M15	M30
Celkový výdělek v eurech při RiskPercent 1	494,56	375,03	313,94
Celkový výdělek v eurech při RiskPercent 3	3 885,76	391,88	1 333,28
Celkový výdělek v eurech při RiskPercent 5	5 419,04	511,45	2 121,40

Zdroj: Vlastní zpracování

Výsledná data, prezentovaná v poslední tabulce, odhalují pozitivní trend v naší strategii. Nicméně je třeba podotknout, že výsledky pro časový rámec M30 by mohly být potenciálně zkreslené kvůli menšímu počtu provedených obchodů ve srovnání například časovým rámcem M5 v daném testovacím období. Tato skutečnost nás upozorňuje na potřebu dalšího zkoumání a možná i přizpůsobení strategie, aby byla zajištěna její robustnost a spolehlivost přes různé časové rámce a tržní podmínky.

Závěr

V rámci této práce jsme se podrobně věnovali vývoji a implementaci obchodního bota v programovacím jazyce MQL5, určeného pro platformu MetaTrader 5. Práce zahrnovala teoretický úvod do forexového trhu, detailní rozbor využitých technických indikátorů a komplexní proces návrhu, vývoje a testování obchodní strategie.

Hlavním cílem bylo vytvořit efektivního obchodního bota, který by na základě předem definovaných pravidel automaticky prováděl obchody. Za tímto účelem byla využita kombinace technických indikátorů: exponenciálního klouzavého průměru, relative strength index a svíčkových formací engulfing, které byly klíčové pro identifikaci obchodních příležitostí.

Během procesu testování byla využita historická data k evaluaci účinnosti bota a jeho schopnosti adaptace na různé tržní podmínky. Provedená testování odhalila klíčové aspekty, které bylo třeba upravit pro optimalizaci výkonnosti bota, což vedlo k dalším iteracím vývoje a finálnímu ladění obchodní strategie.

Získané výsledky ukázaly, že navržený obchodní bot je schopen generovat zisk v daném tržním prostředí, a to zejména po implementaci strategie založené na procentuálním riziku, která výrazně přispěla k lepšímu řízení rizik a větší efektivitě obchodů. Strategie umožnila botu flexibilně reagovat na změny v tržních podmínkách a optimalizovat velikost pozic s ohledem na aktuální výkonnost a kapitálovou dostupnost.

Přestože výsledky dosažené v této práci demonstrují úspěch a efektivitu navrženého obchodního bota, existuje řada oblastí, ve kterých by mohl být bot dále vylepšen. Jedním z klíčových aspektů pro budoucí rozvoj je implementace pokročilé metody money managementu, která by odměňovala delší obchody. Metoda by nejen zlepšila celkovou ziskovost obchodního bota tím, že by se soustředila na obchody s vyšším potenciálem zisku, ale také by pomohla snížit riziko spojené s krátkodobými obchody. Tento přístup by zahrnoval strategii, během které by bot zvyšoval objem obchodu za každý pip pohybu v prospěch obchodu.

Další potenciální vylepšení by mohlo zahrnovat rozšíření analytických nástrojů bota pro lepší rozpoznávání trendů a obrátových bodů trhu, což by umožnilo efektivnější reakci na tržní signály a vylepšenou strategii vstupů a výstupů. Integrace umělé inteligence nebo strojového učení by mohla poskytnout další možnosti pro sofistikovanější analýzu trhu a přizpůsobení obchodních strategií v reálném čase.

V závěru lze konstatovat, že práce splnila svůj primární cíl vytvoření funkčního a efektivního obchodního bota, který demonstruje potenciál automatizovaného obchodování na forexovém trhu. Podařilo se nám prokázat, že s pečlivým návrhem, testováním a promyšlenou implementací lze vytvořit nástroj, který poskytuje obchodníkům významné usnadnění obchodování na forexovém trhu.

Bakalářská práce představuje cenný příspěvek do oblasti automatizovaného obchodování, nabízí podrobný pohled na proces vývoje obchodního bota a jeho aplikaci v praxi. Zkušenosti a poznatky získané během této práce poskytují pevný základ pro další výzkum a rozvoj v oblasti algoritmického obchodování a mohou sloužit jako inspirace pro budoucí projekty v této vyvíjející se disciplíně.

Seznam použité literatury

- CHEN, James, 2022. Bullish Engulfing Pattern: Definition, Example, and What It Means. Investopedia [online] [vid. 2024-01-05]. Dostupné z: <https://www.investopedia.com/terms/b/bullishengulfingpattern.asp>
- CHEN, James. What Is Forex Trading? A Beginner's Guide. Investopedia [online]. 2023 [cit. 2023-11-27]. Dostupné z: <https://www.investopedia.com/articles/forex/11/why-trade-forex.asp>
- FERNANDO, Jason, 2023. Relative Strength Index (RSI) Indicator Explained With Formula. Investopedia [online] [cit. 2024-01-05]. Dostupné z: <https://www.investopedia.com/terms/r/rsi.asp>
- ING. JAROSLAV JAROLÍM, 2021. Klouzavé průměry – jaké je využití v technické analýze? KRYPTOMAGAZIN.cz [online]. Dostupné z: <https://kryptomagazin.cz/klouzave-prumery-jake-je-vyuziti-v-technicke-analyze/>
- JANÁČ, Radek. Automatické obchodní systémy aneb Forex Robot. Traderi.cz, 2016. ISBN 978-80-263-1055-6.
- MAVERICK, J.B., 2022. How Is the Exponential Moving Average (EMA) Formula Calculated? Investopedia [online] [cit. 2024-01-05]. Dostupné z: <https://www.investopedia.com/ask/answers/122314/what-exponential-moving-average-ema-formula-and-how-ema-calculated.asp>
- MetaQuotes Ltd., 2020. MQL5 documentation. Mql5.com [online] [vid. 2024-01-05]. Dostupné z: <https://www.mql5.com/en/docs>
- MetaQuotes Ltd, 2024. Online Forex and exchange trading with MetaTrader 5. Metatrader5.com [online] [cit. 2024-04-07]. Dostupné z: <https://www.metatrader5.com/en/trading-platform>
- SEGAL, Troy, 2022. The Ins and Outs of Forex Scalping. Investopedia [online] [cit. 2024-01-05]. Dostupné z: <https://www.investopedia.com/articles/forex/11/beginners-guide-forex-scalping.asp>
- STIBOR Michal, 2023, FOREX – Jak zbohatnout a nekrást. 3. vydání. Grada Publishing. ISBN 9788027139354.
- THINKMARKETS, 2023. Bullish and Bearish Engulfing Candlesticks | ThinkMarkets | EN. ThinkMarkets [online] [cit. 2024-03-30]. Dostupné z: <https://www.thinkmarkets.com/en/learn-to-trade/indicators-and-patterns/general-patterns/bullish-bearish-engulfing-candlestick-patterns/>
- XTB, 2020. Obchodování na Forexu | Co je to Forex? | Vše, co musíte znát o Forexu. XTB Česká republika [online] [vid. 2024-01-05]. Dostupné z: <https://www.xtb.com/cz/vzdelavani/co-je-to-forex>

Přílohy

V elektronické verzi této bakalářské práce jsou jako přílohy k dispozici následující zdrojové kódy, uložené v souboru matousu_bc.zip:

- bc.mq5
- Bar.mqh
- Indicators.mqh
- Position_management_bc.mqh
- Trade.mqh