

Tallinna Ülikool

Informaatika Instituut

**LIBISEVAL KESKMISEL MA89 PÕHINEVA
FINANTSNÕUSTAJA LOOMINE METATRADER
4 INFO- JA KAUPLEMISTERMINALILE**

Seminaritöö

Autor: Hardo Lass

Juhendaja: Jaagup Kippar

Tallinn 2012

Sisukord

Sissejuhatus.....	3
1 Forex.....	4
2 Platvormi valik.....	6
3 Materjalid internetis.....	7
4 Strateegia.....	8
5 Koodi kasutamine.....	10
5.1 Ettevalmistused.....	10
5.2 Ekspertnõustaja.....	10
5.3 Kliendi indikaatorid.....	13
6 Koodi kirjeldus.....	14
6.1 Ticks	14
6.2 Põhifunktsioonid.....	15
6.3 Ekspertnõustaja.....	16
6.4 Kliendi indikaatorid.....	17
7 Testimine.....	20
7.1 Tulemused.....	21
7.2 Järeldused.....	22
Kokkuvõte.....	23
Kasutatud kirjanduse loetelu.....	24
Lühendid ja mõisted.....	25
LISAD.....	26
Lisa 1: MA89.....	27
Lisa 2: MA.....	32
Lisa 3: MACD.....	33
Lisa 4: Fractals.....	35

Sissejuhatus

Oma töö teema valikul lähtusin kolmest peamisest põhimõttest, mida ka oma eluteekonna valikutel silmas pean. Esiteks uskumus, et ei ole olemas õnne, kui mingit müstilist muutujat, millele lootma jääda, ehk siis lotopiletite ostmine on kasutu tegevus. Teiseks, arvamus, et eesmärkide saavutamiseks on vajalik suunatud tegevus ja kolmandaks teadmine, et need kaks omavahel kombineerituna mõtestatuse ehk arukusega, annavad parima võimaliku kompoti.

Teema valikul lähtusin küll eelpool nimetatud kolmest põhimõttest, aga töö kirjeldamiseks kasutaksin ühiskonna ühte kõige edasiviivamat jõudu, laiskust, mille parim omadus on panna meid mõtlema, kuidas kõige vähema vaevaga saavutada parim tulemus.

Kauplemine finantsturgudel annab meile võimaluse mõista maailma majandust ja sellesse panustada, aga nagu iga kauplemisega, on edukuse aluseks teadmised, kogemused ja distsipliin. Kõige lihtsamini saavutatav on teadmistepagas, mida on võimalik ammutada külastades raamatukogusid või otsides internetist temakohaseid arutelusid. Kogemus saab tulla ainult läbi tegevuse ja distsipliini mõni inimene ei suudagi endas leida või arendada.

Ettevõtte MetaQuotes Software Corporation on teinud mitu versiooni spetsiaalselt finantsturgudel kauplemiseks mõeldud tarkvarast, mis võimaldavad soovijatel automatiseerida enda või teiste poolt välja mõeldud strateegiaid, aidates niimoodi hoida distsipliini ja vältida inimlikke vigu. Programmeerimiskeelena kasutatakse sellel platvormil MetaQuotes Language-t mis on loodud spetsiaalselt kauplemisstrateegiate kirjutamiseks.

Minu töö eesmärgiks on „100% Forex. Õpime ja teenime“ raamatu autorite poolt loodud libiseval keskmisel MA89 põhineva kauplemissüsteemi finantsnõustaja koostamine.

1 Forex

Forex on lühend väljendist *foreign exchange*, mille all peetakse silmas rahvuslike valuutade vahetuskurssidesse või hindadesse investeerimist. Tegemist on globaalse pankadevahelise määratud kursiga valuutavahetusturuga, kus kaubeldakse 24 tundi ööpäevas ja 5 päeva nädalas. Kõige aktiivsem on turg USA sessiooni avamise ja Euroopa sessiooni sulgemise vahel, mis langeb ajavahemikku 13:00 kuni 17:00.

Forex turu maht moodustab kuni 4 triljonit dollarit päevas, mis ületab kümneid kordi kogu USA väärtpaberiturgude päevase kauplemise mahu. Umbes 95% kogu turu mahust moodustab kauplemine kasu saamise eesmärgil.

Kauplemise põhimõtted on sama lihtsad nagu igal teisel turul, ostate valuutat (pikk positsioon), kui selle hind on madal ja müüte, kui hind on kõrge. Teise variandina on võimalik kõigepealt müüa kalli hinnaga laenatud kogus valuutat ja hiljem odavama hinnaga tagasi osta. Sellist tehingut nimetatakse müügiks või lühikeseks positsiooniks. Tehingud on alati kaheosalised, kus kõigepealt toimub positsiooni avamine ja hiljem sulgemine.

Kauplemine ise võib ju lihtne olla, aga selleks et avada Forex turul positsioon (avatud tehing), peab investori käsutuses olema vähemalt 100000 ühikut mingit kindlat valuutat. Võimaldamaks väikeinvestorite kaasamist, võeti kasutusele nn margin (tagatis) kauplemise põhimõte, mis tähendab, et tehingu sooritamiseks peab potentsiaalne turul osaleja üle kandma ainult osa kogu lepingusummast, mis ongi margin ehk garantiideposiit. Selline kauplemissüsteem võimaldab kontrollida rahasummasid, mis ületavad kuni 500 korda esialgset investeeringut.

Tänu sellele eripärale on Forex turg erainvestorite jaoks üks riskantsemaid. Kuna iga tehingu võimendus on niivõrd suur, siis ei tähenda see mitte ainult suurte summade teenimist väga väikese hinnakõikumise korral, vaid ka samasuuri kaotuseid. Hea näitena võimenduse mõjust saab kasutada maja ostmist, mille puhul esmane sissemakse on enamasti 10-20% kogusummast. Kui osta maja, mille väärtus on 100000 USD ja sissemakse 20000 USD, siis aitab pank puuduoleva osa katmisega läbi pangalaenu. Kuue kuu pärast, kui maja hind on tõusnud 120000 USD-ni, maja maha müües, on omakapital kahekordistunud. Paraku on sama ka langusega, kui hind langeb 80000 USD peale ja ostja on sunnitud maja maha müüma, siis on ta ilma ka rahast, mis sissemaksuks läks. Tööpõhimõte Forex turul on just selline, aga võimalus on kaubelda rahasummadega, mis ei ületa enam 5 korda investeeringut, vaid 500

korda.

Viimastel aastatel on Forex üks kõige kiiremini kasvavaid finantsvaldkondasid, mistõttu seda ka Euroopa Liidu poolt hoolikalt jälgitakse ja turg on rangelt reguleeritud. Forexi kiire kasvu üks põhjus on tarkvara areng, mis on tekitanud kauplemises pöördelisi uuendusi. Tehingute teostamine on lihtsam, kiirem ja kauplejad omavad head ülevaadet turust. (Mida kujutab endast valuutaturg FOREX & Mis on Forex)

2 Platvormi valik

Forex turul kauplemiseks mõeldud strateegiat finantsnõustajaks kirjutama hakates, tuli läbi teha pisike taustakontroll, et teada saada, milline platvorm vastab kõige paremini minu soovidele. Silma jäi päris hea ports erinevaid Forex turu jaoks mõeldud kauplemisplatvorme, millest levinumateks on MetaTrader 4, MetaTrader 5, cTrader, Ninja Trader, Apex Trader ja Mirror Trader.

Kuna soov oli luua automatiseeritud tarkvara kindla strateegia jaoks, siis langes kohe välja Mirror Trader, mille põhieesmärk on kasutada eelnevalt valmis kirjutatud ja testitud strateegiaid, lisades need enda kontole. Iseenesest kõlab see kõige mugavama lahendusena aga kaotab ära võimaluse asju enda moodi lahendada ja kindlasti ei sobi antud ülesande puhul.

Uurides cTraderit, Ninja Traderit ja Apex Traderit, jäi silma, et kõigi puhul on võimalik kirjutada ka ise strateegiaid ja kaubelda automatiseeritult just enda soovide järgi, aga ükski nendest ei võimalda eestikeelset kliendile mõeldud kasutajatuge, kui peaks olema soov enda strateegiaga päris turule kauplema minna.

Valik on juba suhteliselt nigel ja terve mõistus ütleb, et ajas tagasi ei ole mõtet minna ning seega peaks platvormiks jääma MetaTrader 5. Paraku võimaldab MetaTraderi viimane versioon ainult demo konto peal kauplemist ja päris turule lähipäevil minna ei saa, mis jätab valikuks ikkagi neljanda versiooni.

3 Materjalid internetis

MQL4 Strategy Language

Programmeerimiskeele MQL4 kommuun, kus on ka keele dokumentatsioon. Parim koht leidmaks abi finantsnõustajate, skriptide või indikaatorite kirjutamisel. Inglise keelne

<http://www.mql4.com/>

Investopedia

Investeeringisuudised ja mõistete andmebaas, mis puudutab finantsmaailma. Inglise keelne

<http://www.investopedia.com>

Forex turu vahendaja NetStock AS

Internetipõhise kauplemise vahendaja valuutavahetusturul. Informatsioon ja uudised Forex turu kohta. Eesti keelne

<http://www.forex.ee/>

Forex turu vahendaja Admiral Markets

Internetipõhise kauplemise vahendaja valuutavahetusturul. Informatsioon ja uudised Forex turu kohta. Eesti keelne

<http://www.admiralmarkets.ee>

Automated Trading Championship

Hea ülevaade automatiseeritud kauplemise võimalustest ja ka riskidest. Näidatakse kõigi osavõtjate ekspertnõustajate tehingute ajalugu ja graafikuid, mida analüüsid võib saada aimu töötavate või läbi kukkumisele määratud strateegiate kohta. Inglise keelne

<http://championship.mql5.com/>

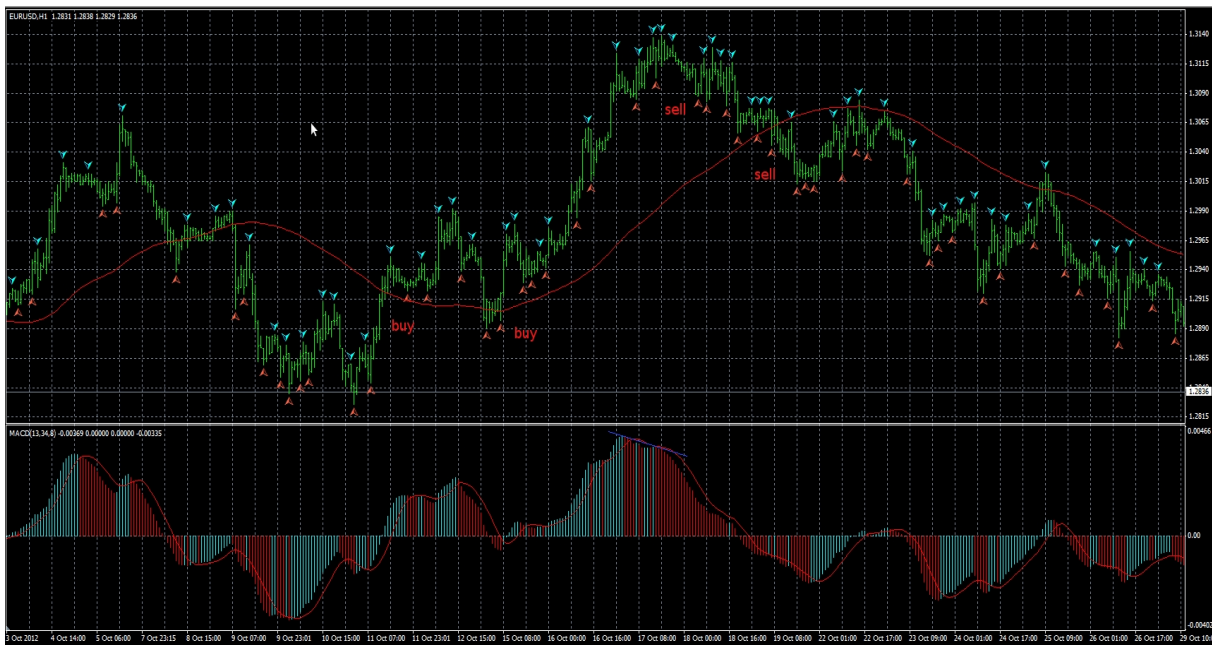
4 Strateegia

Süsteemis kasutatakse 89 perioodiga tavalist libisevat keskmist. Täiendavate indikaatoritena kasutatakse fraktaale ja MACD-d parameetritega 13, 34 ja 8. Süsteem on kasutatav kõigil kõrge likviidsusega turgudel ja ükskõik milliste ajaliste intervallidega graafikutel. Enne reaalsel kauplemist on soovitatav siiski jälgida konkreetse finantsinstrumendi käitumist, et selgitada välja võimalikud puudused, mis sõltuvad konkreetsest turust (futuurid, aktsiad, Forex).

Ostusignaali saame läbimurdel või kui signaalküünal sulgub kõrgemal kui MA89 (müügi puhul vastupidi). Stop Lossi asetame viimasest fraktaalist allapoole, signaalile vastassuunaliselt. Teistkordse ostusignaali saame, kui MA89 joon on suunatud üles, hind aga läheneb MA89 joonele ja seejärel põrkub. Sellisel juhul pole soovitatav kiirustada, vaid oodata tõusva küünla (härja küünla) sulgumist ja avada seejärel pikk positsioon.

Pärast positsiooni avamist tuleb jälgida kas hind formeerib uue maksimumi, samuti vaatleme indikaatori MACD näite – seni kuni formeeruvad uued, kõrgemad joonekesed, on trendil potentsiaali. Soovitatav on perioodiliselt nihutada Stoppi 5-10 pipsi võrra allapoole MA89-st. Kui trend areneb järsu kaldega, asetame Stop-orderi fraktaalidest allapoole. Sel juhul on aga oht, et väljume turult liiga vara ega suuda realiseerida täit kasumipotentsiaali. Kui MACD loob vastupidise divergentsi, siis võime pool positsioonist sulgeda (võimalik, et trend on end ammendanud), kuid mitte tervet positsiooni, sest peale korrigeerimise on võimalus, et hind jätkab tõusu ja sellisel juhul kaotaksime potentsiaalse kasumi. Positsioon suletakse täielikult kui on jõutud Stop-tasemeni või on ilmnenud pöördemärgi signaal (küünla sulguminegraafikul, MA89-st allpool) (Daražanov, Banov ja Kozarov, 100% Forex. Õpime ja teenime, 2010).

Pildil 1 on toodud näide kauplemisest kirjeldatud strateegia järgi tunnigraafiku alusel. 11 oktoobril ilmneb esimene ostusignaali, kui signaalküünal murrab läbi libisevast keskmisest hinnatasemel 1,2924. Avatakse ostupositsioon Stopiga allpool viimasest fraktaalist, tasemel 1,2842. Teistkordse ostusignaali saame 15 oktoobril, kui signaalküünlad põrkuvad suuna ülespoole võtnud libiseva keskmisega. 17-ndal oktoobril loob MACD vastupidise divergentsi ja suletakse pool positsiooni tasemel 1,3120. Teine pool positsioonist suletakse, kui signaalküünlad murravad uuesti läbi libisevast keskmisest või Stop tasemel, kui trendi järsu kalde tõttu seda fraktaale silmas pidades ülespoole nihutati.



Pilt 1: strategia kirjeldus

5 Koodi kasutamine

5.1 Ettevalmistused

Töö käigus valminud ekspertnõustaja käivitamiseks on vajalik MetaTrade 4, mille saab alla laadida näiteks aadressilt <http://www.admiralmarkets.ee>.

Vajalik on ka demokonto, mille saab registreerida tasuta eelpoolmainitud leheküljel. Kuna ekspertnõustajat on testitud ainult Forex turul, siis võib CFD konto kõrvale jätta ja teha tavalise demokonto. Konto loomisel tuleb täita kõik väljad õigete andmetega, valida asukohariik, kontotüüp EUR (Admiral.Standard) ja Deposiit ehk algsumma kontol. Iga inimene võib teha nii palju demokontosid kui soovib, nii et mänguruum on suur ja pole vaja muretseda, kui demokontol raha otsa saab. Peale registreerimist saadetakse automaatne email, mis sisaldab kasutajatunnust, parooli, investori parooli ja telefoni parooli.

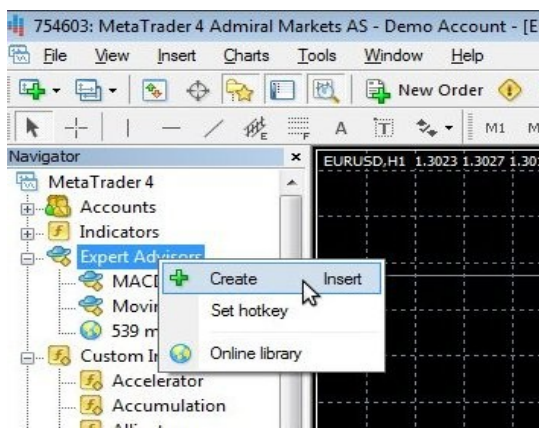
MetaTraderi käivitamise järgselt tuleb logida kasutajaga sisse, mida saab teha valides menüüst File-Login. Login on siin kasutaja mõistena, password on parool ja serveriks tuleks valida AM-Demo ning jääb veel kinnitada Login nuppu vajutades. Sisse logimise õnnestumisest annab märku Terminalil pesitsev Journal või Navigator, mille menüüst leiab Accounts ja kui seal all on olemas ka kaustaja nimi, oli sisse logimise katse edukas.

5.2 Ekspertnõustaja

Kui ettevalmistused said ilusti tehtud, siis on praeguseks hetkeks avatud MetaTrader ja ka kasutajaga ilusti sisse logitud.

File menüüst on leitav New Chart, mis avab rippmenüü valuutapaaride valikuga, kust tuleb valida EURUSD. Avaneb uus graafik, kus on näha selle valuutapaari hinna kõikumine suhteliselt pika perioodi vältel.

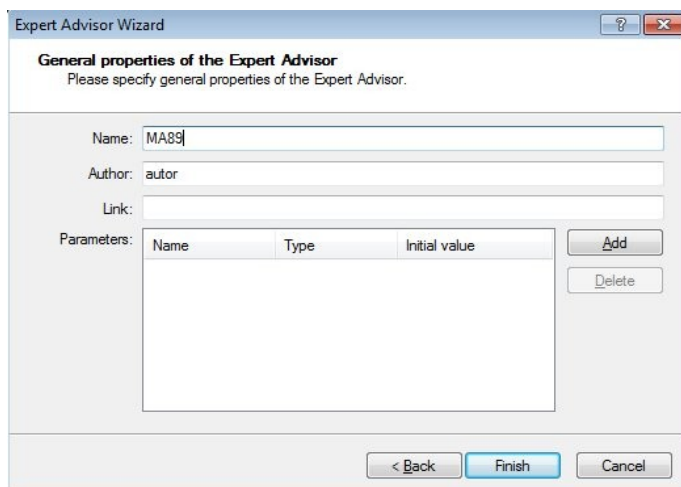
Ekspertnõustaja loomiseks, tuleb teha Navigatorist Expert Advisorsi peal parem hiireklikk, mispeale saab valida Create.



Pilt 2: uue ekspertnõustaja loomine



Pilt 3: tüübi valik

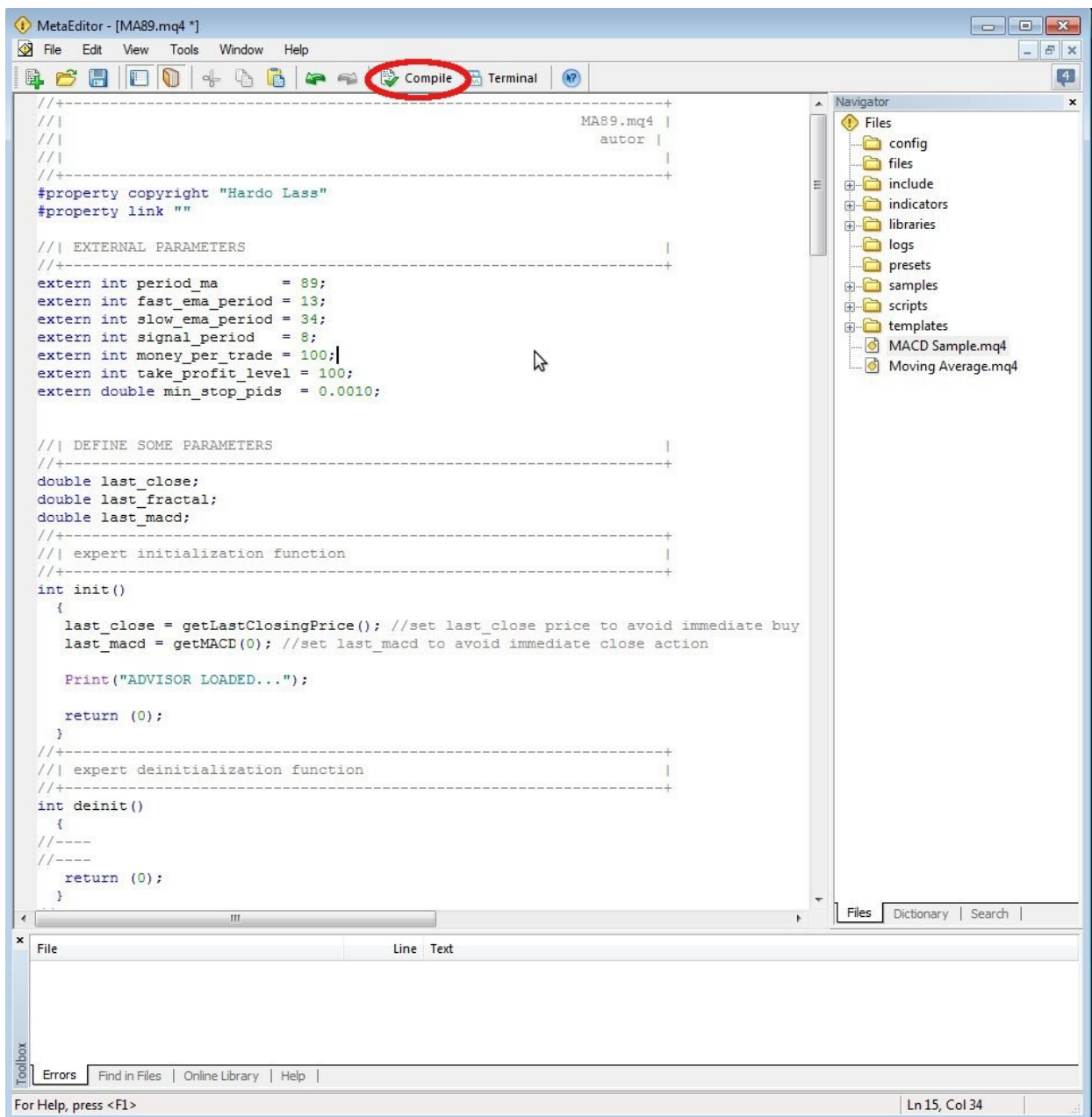


Pilt 4: nime valik

Edasi avaneb editor eeltäidetud tühjade funktsioonidega, mida antud hetkel vaja ei lähe, Peale sisu üle kopeerimisest koodiga (vt lisa 1), salvestamist ja kompileerist, tekib MetaTraderis Navigator menüü alla uus Expert Advisor. Antud hetkel siis MA89 nimeline.

Ekspertnõustaja käivitamiseks tuleb see *Navigator* menüüst lohistada soovitud graafikule või

avada topeltklikiga ja soovi korral muuta mõningaid seadeid ning kinnitada OK nupuga. Infot tehingute, vigade ja muu huvitava kohta leiab Terminali akendes *Trade*, *Account history* ja *Journal*.



Pilt 5: koodi kopileerimine

5.3 Kliendi indikaatorid

Kuigi Metatrader-ile on sisse ehitatud suur hulk indikaatoreid, mille seas leiduvad ka MA, MACD ja fraktaalid, on antud töö jaoks kirjutatud eraldi koodid (vt lisa 2, 3, 4), et ei peaks neid iga kord uuesti seadistama. Nende aktiveerimiseks tuleb toimida samamoodi nagu ekspertnõustaja puhul, kuid tüübi valikus (vt pilt 3) tuleb valida Custom Indicator. Peale kompileerimist tekivad meile MetaTraderis Navigator menüü alla uued Custom Indicatorid ja nende käivitamine toimub samamoodi nagu ekspertnõustajate puhul.

6 Koodi kirjeldus

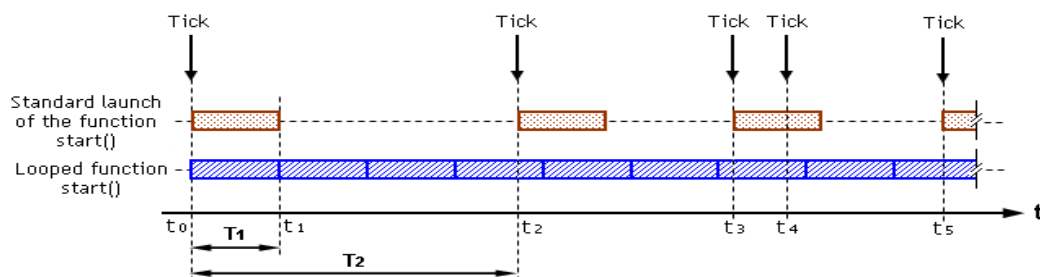
6.1 Ticks

MQL4 koodis programmide kirjutamiseks on eelnevalt vaja mõista kliendi terminali ja finantsturu vahelist suhtlust, kuna see käib interneti vahendusel ning turu olukord muutub ööpäevaringselt kiiremini, kui me seda infot tuvastada jõuame.

Kliendini jõuab info *tick*-ide kaudu, mis saadetakse vahendaja serverist ja sisaldab uut hinnakurssi kaubeldava valuutapaari kohta. Kuna serverite seadistused ja internetiühenduste kiirused on erinevad, siis on ka saabuvate *tick*-ide sagedus kõikuv. See on ääretult oluline teave ekspertnõustajate koostamisel, sest üldjuhul käivitatakse ekspertnõustaja *start()* funktsioon iga *tick* saabudes, mille peale tehakse kalkulatsioonid ja programm jääb jälle unelema. Enamuse ajast on programm ooteseisundis ja seda seetõttu, et server ei saada uut infot peale. Uut infot on aga võimalik ise serverilt küsida, tekitades *start()* funktsiooni tsüklil (General Characteristics of Complex Programs):

```
//-----  
int start()  
{  
    while(!IsStopped()){ // Until user stops execution of the program  
        RefreshRates(); // Refrest server information  
  
        //programmi põgi osa  
  
        Sleep(5); // Little break, 5 milliseconds  
    }  
    return(0);  
}  
//-----
```

Koodinäide 1: *start()*, turuinfo uuendamise tsüklil



Pilt 6: tickid standard ja tsüklilisel kujul

Peab tunnistama, et sellise tsükli lisamine ei mängi suurt rolli, kui vahendaja pakub hindasid 4 komakoha täpsusega. Hinnakursside muutused ei toimu sellisel juhul just kuigi tihti kiiremini kui 1 *tick* sekundis (pigem 1 *tick* 4 sekundi kohta). Samuti ei suuda sellise tsükliga toime tulla MetaTraderi testkeskkond, mistõttu on ekspertnõustajat testides see osa koodist välja

kommenteeritud.

6.2 Põhifunktsioonid

MQL4 on 3 eeldefineeritud funktsiooni, mida ei tohi kasutada teiste funktsioonide nimedena. Ära sai juba mainitud *start()*, millele lisaks on *init()* ja *deinit()*. Need funktsioonid erinevad teineteisest välja kutsumise poolest.

Init() kutsutakse välja(käivitatakse) peale kliendi terminali avamist, ajaloo üles laadimist, valuutapaari või ajaintervalli vahetamist, programmi kompileerimist, peale sisendparameetrite ja konto muutmist. Ehk siis alati, kui peaks käivitama kogu programmi uuesti. Üldjuhul pannakse *init()* funktsioonis paika tööks vajalikud muutujate parameetrid või tõmmatakse käima funktsioonid, mida programmi elutsükli jooksul läheb ainult alguses vaja. Oma töös olen *init()* funktsiooni kasutanud muutujate väärtustamiseks, et vältida *start()* käivitumist ilma õigete parameetriteta.

```
//+-----+
//| expert initialization function |
//+-----+
int init()
{
    symbol = Symbol();
    last_close = getLastClosingPrice(); //set last_close price to avoid immediate buy
    action
    last_macd = getMACD(0); //set last_macd to avoid immediate close action

    min_lot = MarketInfo(symbol, MODE_MINLOT);
    max_lot = MarketInfo(symbol, MODE_MAXLOT);
    one_lot = MarketInfo(symbol, MODE_MARGINREQUIRED);
    step = MarketInfo(symbol, MODE_LOTSTEP);
    stop_level = MarketInfo(symbol, MODE_STOPLEVEL);
    Print("ADVISOR LOADED...");

    return (0);
}
```

Koodinäide 2: *init()*

Start() kutsutakse välja iga kord peale uue *tick* saabumist. Kui *start()* ei ole oma tööd eelmise *tick* kallal lõpetanud, siis saabunud *tick* jäetakse vahele ja töö lõppedes oodatakse uut. Samuti ei kutsuta seda funktsiooni välja juhul, kui ekspertnõustajad on välja lülitatud või on avatud ekspertnõustaja seadistuste aken.

Deinit() käivitatakse programmi, kliendi terminali või valuutapaari akna sulgemisel, vahetult enne valuutapaari või ajaintervalli vahetamist, eduka kompileerimise, ekspertnõustaja sisendparameetrite ja konto muutmise järel.

Kõik kolm eeldefineeritud funktsiooni on võimalik defineerida ka parameetritega ja kutsuda

neid suvalisel ajal välja. Sisaldades parameetreid, peavad neil olema vaikimisi väärtused, kuna süsteem ise käivitab neid ilma. Programmi tööks ei ole vajalikud *init()* ja *deinit()* ning kõik kolm võivad asetseda koodis suvalises järjekorras. (Special Functions)

6.3 Ekspertnõustaja

Kauplemiseks finantsturul on antud töös vajalik ekspertnõustaja (Expert Advisor), mis enda kõhus teeb ära kõik kalkulatsioonid ja analüüsid, mis ühe tehingu sooritamiseks vajalikud on ja samuti hoiab silma peal, et õigel ajal tehingud uuesti sulgeda.

Eelnevalt oleme ära kirjeldanud strateegia, mille alusel hakkame enda programmijuppi koostama ja nagu ikka, siis teooria ja praktika lähevad kokku üldpildis, kuid detailides on ühesuse saavutamine palju keerulisem.

Otsides turule sisenemise hetke, käivitame iga *tick* käigus funktsiooni *isOrderAction()*, mille ülesanne on tagastada informatsiooni võimaliku tehingu sooritamiseks. Peamiselt kontrollib see funktsioon, kas viimase kahe sulgemishinna vahele jäi valuutapaari keskmine hind, ehk kas toimus keskmise hinna ületamine. Samuti kontrollitakse, mis suunas keskmise hinna ületamine toimus ja millises suunas hetkel keskmine hind liigub. Nende parameetrite alusel otsustatakse, kas avada ja kui üldse avada, siis millise positsiooni peaks avama.

Strateegia kirjelduses turule sisenemiseks vaadatakse ka keskmise hinnaga teistkordset pörkumist, mille puhul peaks keskmise hinna suund olema juba turu suunaga samale poole. Paraku sellist olukorda, kus saaks turul otsustada kahe järjestikuse signaali järel, on ülimalt keeruline tabada, kuna läbimurre toimub ühekordselt või jääb turg keskmise hinna kohal pikalt pendeldama ning õige signaali (teistkordse läbimurde) tuvastamine eeldab mõne uue indikaatori välja mõtlemist või teisest signalist üldse loobumist. Kui kasutada pendeldamisel tekkinud teist signaali, tekib olukord, kus tehakse väga palju tehinguid lühikese aja jooksul ja valdav enamus on vale suunalised ning söövad investeeritava raha ära lihtsalt seetõttu, et need suletakse Stop orderi või eksperdi enda poolt.

Peale turule sisenemist saab põhirolli uute korralduste jagamiseks enda kätte *manageTicket()*, mille ülesanne on jälgida turul toimuvat ja otsustada, kas avatud positsioonil nihutada Stop orderit, sulgeda positsioon või mitte midagi teha.

Stop orderit nihutatakse alati kui tekib uus, avatud positsioonile vastassuunaline fraktaal, mis on ka uue Stopi asukohaks.

Positsiooni sulgemise indikaatoriks on MACD põhiväärtuse ristumine MACD

signaalväärtusega, mis näitab tehingute suundade muutust väikese nihkega, andes mõista turul kauplemise suuna muutusest.

Strateegia kirjeldus näeb ette poole positsiooni sulgemise, kui MACD signaalid ristuvad ja rõhutab, et tuleb jätta avatuks teine pool, mis suletakse alles Stop orderi või valuutapaari keskmise hinna ületamisel, et välistada potentsiaalse kasumi kaotamist. See on osa, mida võib mõista mitmeti ja minu jaoks kõlab poole positsiooni sulgemine potentsiaalsest kasumist loobumisenä ja arvestades ka ülejäänud eksperdi tööd, mis nii ehk naa aeg ajalt toodab kahjumit, siis sulgedes terve positsiooni, on tõenäosus kasu teenida oluliselt suurem, kui oletada ja kaotada võidetust teine pool, mis võib viia kogukasumi nulli või miinusesse.

Lisaks kahele eelpoolmainitud funktsioonile on suur hulk teisi, mille ülesanne on toetada eksperdi tööd, varudes vajalikku infot ja tehes kalkulatsioone, aga nagu põhifunktsioonide peatükis juttu oli, siis algab ka konkreetse eksperdi töö *init()* funktsioonis, kus omistatakse tööks vajalikele parameetritele väärtused ja laetakse esimene info turu kohta.

Parema liigenduse huvides on enamus loogikast *start()* funktsioonist välja tõstetud ja jäetud sinna ainult kontroll eksperdile piisava taktikoguse kohta, mille olemasolul värskendatakse jällegi turuinfot ja käivitatakse *main()* funktsioon. Selle funktsiooni peamine ülesanne on otsustada, kas suunata töö edasi juba olemasolevate tehingute töötlemiseks, otsida võimalust turule siseneda või hoida ekspert eemal tegevusest kuna on leitud mõni tehing, mis ei sobi hetkel töötava eksperdi tööga. Koodi kogu eksperdi kohta leiab lisast 1.

6.4 Kliendi indikaatorid

Ekspertnõustaja kasutab oma töö tegemiseks erinevaid indikaatoreid, mille jälgimine on kasulik turust ülevaate saamiseks ka juhul, kui ise tehinguid tegema ei pea. Kuigi MetaTrader võimaldab kõiki kasutatud indikaatoreid graafikule kuvada ka menüü valikust, on need kirjutatud kliendi indikaatoritena, et ei peaks iga kord menüüst valides seadistama.

Indikaatorite kood koosneb valdavalt kolmest osast – indikaatori elementide seadistamine, indeksite seadistamine *init()* funktsioonis ja defineeritud puhvrite täitmine andmetega. Indikaatori elementide seadistamisel pannakse paika, kas hakatakse seda kuvama valuutapaariga ühes aknas või eraldi. Indikaatori puhul kasutatavate puhvrite arv, värvid ja paksused.

```
//---- indicator settings
#property indicator_chart_window
#property indicator_buffers 2
#property indicator_color1 Aqua
#property indicator_color2 Tomato
#property indicator_width1 1
#property indicator_width2 1
```

Koodinäide 3: indikaatori elementide seadistamine fraktaalide jaoks

Init() funktsioonis määratakse indeksitele, millisest puhvrists nad andmeid võtavad, millist tüüpi kujundeid nad välja hakkavad joonistama ja panakse külge sildi, mida kuvatakse hiljem andmete aknas. Indekseid on võimalik ühe indikaatori kohta seadistada 8 (vahemikus 0 – 7).

Kuna antud näites on tegemist DRAW_ARROW tüüpi kujundiga, siis saab anda kaasa ka kujundi, mida ta näitama hakkab. Kujundeid on võimalik valida vahemikust 32 – 255, mis loomulikult ei ole kõik nooled.

```
//+-----+
//| Custom indicator initialization function |
//+-----+
int init()
{
    SetIndexBuffer(0, ExtUpperBuffer);
    SetIndexStyle(0, DRAW_ARROW);
    SetIndexArrow(0, 218);
    SetIndexLabel(0, "Fractal Up");

    SetIndexBuffer(1, ExtLowerBuffer);
    SetIndexStyle(1, DRAW_ARROW);
    SetIndexArrow(1, 217);
    SetIndexLabel(1, "Fractal Down");

    return(0);
}
```

Koodinäide 4: *init()* seadistamine fraktaalide jaoks

Kõige keerulisem osa jääb ka seekord *start()* sektsiooni, kus täidetakse indeksite poolt kasutatavad puhvrid andmetega, mille alusel hakatakse kuvama eelpool defineeritud elemente(nooli, kriipse, joont vms). Selleks, et puhvrid oleksid kättesaadavad nii *init()* kui ka *start()* funktsioonile, tuleb need defineerida väljaspool neid funktsioone:

```
//---- indicator buffers
double ExtUpperBuffer[], ExtLowerBuffer[];
```

Koodinäide 5: puhvrite seadistamine fraktaalide jaoks

Nagu ka ekspertnõustaja puhul, käivitatakse *start()* iga *tick* saabumisel. Funktsioon vaatab üle, kui palju takte on läbi käidud, arvutab välja puuduolevate taktide koguse ja lisab puhvritele selle koguse takte soovitud informatsiooniga. Joonte ja noolekeste kuvamine vastavalt puhvis leiduvale sisule teeb ära juba Metatrader ja sellega enda pead rohkem vaevama ei pea.

```

int start()
{
    int limit, i;
    int counted_bars = IndicatorCounted();

    if(counted_bars > 0) counted_bars--;
    limit = Bars - counted_bars;

    for (i = 1; i <= limit; i++){
        ExtUpperBuffer[i] = iFractals(NULL,0,MODE_UPPER,i) + 0.0007; //array y location
        ExtLowerBuffer[i] = iFractals(NULL,0,MODE_LOWER,i) - 0.0007; //array y location
    }
    return(0);
}

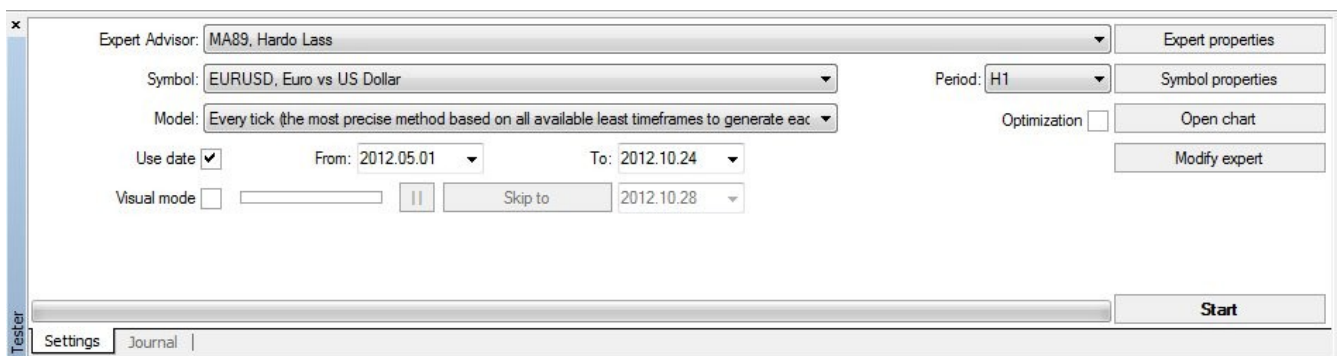
```

Koodinäide 6: *start()* funktsioon fraktaalide tarvis

Sama loogika alusel töötavad ka kõik teised kliendi indikaatorid. Konkreetsetes töös on kasutatud indikaatoritena libisevat keskmist perioodiga 89 (vt. *Lisa 2*), fraktaale (vt. *Lisa 4*) ja MACD parameetritega 13, 34 ja 8 (vt. *Lisa 3*).

7 Testimine

MetaTraderi strateegia testimiseks on võimalusi kaks. Suhteliselt aeganõudev, aga kahtlemata testimiseks kõige parem lahendus on otse demokontol ekspertnõustaja kasutamine. Olenevalt graafiku valikust võib tulemuste saamine võtta aega päevi, nädalaid või isegi aastaid ja seetõttu on targad poisid loonud spetsiaalse testimise keskkonna imiteerimaks reaalsel turgu, aga seda kiirendatud kujul. *Tester* keskkonda kasutades saab tagasisidet eksperdi töö kohta nii graafiliselt(kui on soovi avaldatud) kui *Results* kaudu. Antud töös on ülesande testimiseks kasutatud just seda vahendit. Leiab selle menüüst *View -> Strategy Tester* ning stardipositsioonil näeb ta välja selline:



Pilt 6: Strategy Tester

Seadistused pannakse paika vastavalt pildile:

- *Expert Advisor* – eelnevalt koostatud ja kompileeritud ekspertnõustaja, mida hakatakse testimata (valitud MA89)
- *Symbol* – valuutapaar, mida kasutatakse testimisel (valitud EURUSD)
- *Period* – testimisel kasutatav ajaline intervall(valitud H1, ehk tunnigraafik)
- *Model* – testimise meetod. Valida saab iga *tick*, kontrollpunktide ja avamishindade järgi testimist (valitud *Every tick*)
- *Use date* – võimaldab valida kuupäeva vahemikku testimiseks. Kui linnuke eemaldada kasutatakse kogu ajalugu. *From* ja *to* kuuluvad *use date* juurde (valitud 2012.05.01 – 2012.10.24)
- *Visual mode* – graafilise poole sisse lülitamine. Kogu testimine käib ka graafiliselt. Väga hea moodus analüüsima eksperdi käitumist. Miinuseks on testimise kiirus, mis saab pehmelt öeldes kannatada (ei kasuta)

- *Optimization* – võimaldab testida eksperdi tööd teiste parameetritega, väljastades positiivsete tulemustega testide parameetrid ja hulgaliselt statistikat (ei kasuta, kuna tulemused võivad muuta kogu strateegiat).
- Teised nupud – eksperdi või testi seadistamiseks vajalikud, kuid tehakse ülipeent testimist näiteks optimeerimise juures (ei kasuta).

Kui seadistused on paigas, siis on jäänud veel ainult **Start** nupp ja test hakkab tööle.

7.1 Tulemused

Olenevalt ajalise intervalli, graafilise liidese ja valitud ajavahemikust, võib testi kestvus varieeruda mõnest sekundist tundideni. Tulemusi näeb aknas *Results*. Selles aknas parema hiireklikiga mängides, on võimalik salvesta tulemused ka raportina, mis on html faili formaadis, sisaldades tehingute ajalugu, statistikat ja muid testi parameetreid. Antud ülesandes läbi viidud testi tulemused leiab failist *MA89_results.htm*, mis on originaalkujul.

Kuna eksperdi parameetrites on *money_per_trade* väärtuseks 100, siis toimuvad kõik tehingud kasutades 100% kontol olevat raha. Esimesed kaks tehingut on väga paljulubavad ja viivad portfelli (konto seis) 10000-lt 20929 peale. Järgneb kolm tehingut, mille tulemuseks on praktiliselt alguspunkti tagasi liikumine ja tuleb uut hoogu otsida 10971 pealt. Jällegi saadab edu ja kahe positiivse tehinguga on tulemuseks rekordiline 26290. Sellele viimasele positiivsele tehingule järgned hunnik negatiivseid tehinguid, millest õnnestub läbi murda veel ühel katsel ja jääda positiivsesse tootlusse. Kontol ilutseb pooleks päevaks isegi number, mis ületab 27000, aga sealt edasi on valdavalt negatiivse tulemiga tehingud, mis jätavad testi lõppedes kontole pool algsest summast, ehk 5027,05.

Statistika poole pealt näeme, et kokku tehti 42 tehingut, millest 23 olid müügipositsioonid ja nendest kasumlikud olid ainult 43.48%. Ostu positsioone avati 19 ja kasu tootis neist 26.32%. Üldse kokku oli 15 (35.71%) kasumlikku tehingut ja negatiivse tootlusega 27 (64.29%) tehingut.

Suurim tootlus tehingu kohta oli 9344.49, suurim kahjum 7071.34 ja keskmine kasum tehingust oli 3529.58 ning kahjum 2145.06.

Tulemuste lugemiseks tabelist on hea teada, mida mingi veerg tähendab:

– tegevuse number, kusjuures tegevuseks võib olla uue tehingu sooritamine või näiteks vana tehingu muutmine

Time – tegevuse sooritamise aeg

Type – tegevuse tüüp

Order – tehingu number, mille raames tegevust läbi viiakse

Size – tehingu suurus (*lots*)

Price – tehingu hind

S/L – Stop Loss, ehk hinnatase, kus väljutakse turult automaatselt. Vastassuunaline ennustatud hinna liikumisele. Kasutatakse kahjumite minimeerimiseks.

T/P – Take Profit, ehk hinnatase, kus väljutakse turult automaatselt. Samasuunaline ennustatud hinna liikumisele. Kasutatakse kasumite välja võtmiseks, kui turg jõuab mingile tasemele.

Profit – tehingust saadud kasum

Balance – kasutaja konto seis

7.2 Järeldused

Kuigi positiivsete tehingute keskmine tootlus ja maksimumtootlus on kõrgemad kui negatiivsete omad, jäävad positiivsed tugevalt alla enda esinemise sageduse poolest ja kaotused on ikkagi piisavalt suured, et lõpptulemust nii inetuks kujundada nagu ta parasjagu sai.

Vaeva tuleks näha nii negatiivsete tehingute sageduse vähendamisega kui ka kaotuste enda minimeerimisega ja sellest strateegiast võiks isegi midagi positiivse tootlusega saada. Hetkel jätab ta suhteliselt mannetu mulje ja kauplema temaga minna ei julge.

Kokkuvõte

Antud töö eesmärgiks oli luua libiseval keskmisel MA89 põhinevale kauplemissüsteemile ekspertnõustaja, mille võib lugeda täidetuks, olenemata õrnast pettumusest, mis jääb süsteemi sobimatuse tõttu hõljuma.

Töö käigus sai selgeks ühe hea kauplemissüsteemi detailsuse olulisus, mis paneb alustala kogu süsteemi käitumisele. Indikaatorid, mille alusel otsustatakse tegevuste käiku, peavad lisaks üheselt mõistetavusele, olema ka piisavalt unikaalsed, et vältida süsteemi üle koormamist vale või petliku informatsiooniga.

Põgusalt sai tutvust tehtud ka ekspertnõustaja testkeskkonnas pesitseva *optimization* funktsionaalsusega, mis töö enda kirjeldustest jäi küll välja aga on mainimist väärt, kuna tegemist on MetaTraderi ühe kõige võimsama tööriistaga ja õige kasutuse korral võib väikesest künnihobusest teha mitmesaja hobujõulise viljakombaini.

Töö edasiarenduse või parema strateegia välja töötamisel, on *optimization* kindlasti üks töövahendeid millest ei saa mööda vaadata. Tasuks visata ka pilk peale MetaTraderi kõige uuemale versioonile, millega mõne aja möödudes saab kindlasti reaalseid tehinguid sooritama hakata ning väheke suuremaks väljakutseks on ehk MetaQuotes Software Corp poolt korraldatav Automated Trading Championship, mida korraldatakse aastast 2006 ja käesoleva aasta üritus on järjega kuues. Seal kasutatavad ekspertnõustajad jooksevad just MetaTraderi viimas versiooni peal.

Kasutatud kirjanduse loetelu

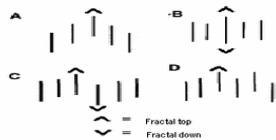
1. Daražanov, A., Banov, V. Ja Kozarov, A. (2010). *100% Forex. Õpime ja teenime*. Tallinn: Admiral Markets
2. Mis on Forex? (kuupäev puudub). Kasutamise kuupäev: 20.10.2012. a., allikas Admiral Markets: <http://www.admiralmarkets.ee/education/what-is-forex/>
3. Mida kujutab endast valuutaturg FOREX? (kuupäev puudub). Kasutamise kuupäev: 20.10.2012. a., allikas FOREX.ee: <http://www.forex.ee/forex-est/mida-kujutab-endast-valuutaturg-forex/>
4. General Characteristics of Complex Programs. (kuupäev puudub). Kasutamise kuupäev: 21.10.2012. a., allikas MQL4 Book: <http://book.mql4.com/special/index>
5. Special Functions. (kuupäev puudub). Kasutamise kuupäev: 27.10.2012. a., allikas MQL4 Book: <http://book.mql4.com/programm/special>
6. Investeerimisõpik. (kuupäev puudub). Kasutamise kuupäev: 28.10.2012. a., allikas LHV finantsportaal: <http://lhv.delfi.ee/academy/investmentguide>

Lühendid ja mõisted

MA - *Moving Average*, valuutapaari keskmine hind, arvatuna mingi perioodi alusel

MACD - *Moving Average Convergence/Divergence*, libisevate keskmiste koondavusel või lahknemisel põhinev indikaator

Fraktaal – kujund tehnilises analüüsis, mis moodustub 5 küünla põhjal, kus keskmine küünal on teistest eraldunud, moodustades noole taolise kujundi.



Stop Loss – positsiooni sulgemise hind kaupleja kahjude vähendamiseks, mis pannakse paika kauplemistehingut tehes (võimalus ka hiljem muuta)

Divergents – kahe vaadeldava indikaatori näitajate vastassuunaline liikumine

Küünaldiagramm – turul toimuvate muutuste kirjeldamise viis. Koosneb kahte sorti küünaldest – karud ja härjad. Karud iseloomustavad müüjate ülekaalu turul ja härjad ostjate ülekaalu. Nimetused on tulnud loomade ründestiilidest

Ostusignaali/müügisignaali – kaupleja poolt välja mõeldud olukord turul, kus ta siseneb või väljub turule

Futuurid – Futuur on pooltevaheline leping, mis kohustab ostma või müüma varem kokkulepitud alusvara varem kokkulepitud ajal ning varem kokkulepitud kohas kindla hinnaga.

Pips – valuutakursside koteeringute minimaalne muutus

Koteering – ühe riigi rahvusliku valuuta hulk, mis on vajalik ühe ühiku valuuta ostuks

LISAD

Lisa 1: MA89

```
//+-----+
//|                                     MA89.mq4 |
//|                                     hardo |
//|                                     |
//+-----+
#property copyright "Hardo Lass"
#property link ""

//| EXTERNAL PARAMETERS |
//+-----+
extern int period_ma      = 89;
extern int fast_ema_period = 13;
extern int slow_ema_period = 34;
extern int signal_period  = 8;
extern int money_per_trade = 100; //percent
extern double min_stop_pids = 0.0010;

//| DEFINE SOME PARAMETERS |
//+-----+
double last_close;
double last_fractal;
double last_macd;
double min_lot;
double max_lot;
double one_lot;
double step;
double stop_level;
double ticket_lots;
int ticket_type;
int last_ticket;
int ticket;
string symbol;

//+-----+
//| expert initialization function |
//+-----+
int init()
{
    symbol = Symbol();
    last_close = getLastClosingPrice(); //set last_close price to avoid immediate buy action
    last_macd = getMACD(0);           //set last_macd to avoid immediate close action
    updateMarketInfo();

    Print("ADVISOR LOADED...");

    return (0);
}

//+-----+
//| expert start function |
//+-----+
int start()
{
    if(Bars < period_ma) // Not enough bars
    {
        Alert("Not enough bars in the window. EA doesn't work.");
        return;
    }

    //while(!IsStopped()){ // Until user stops execution of the program
        RefreshRates();
        updateMarketInfo();
        main();

        // Sleep(5); // Little break, 5 milliseconds
        //}
    return(0);
}

//+-----+

void main()
{
```

```

double before_last_close = last_close;
double average, macd;
string order_type;
double direction;
int total=0;
ticket = -1;
ticket_type = -1; // Type of selected order (Buy=0, Sell=1,
None=-1)

for (int i = OrdersTotal()-1; i >= 0; i--){
//for(int i = 0; i <= OrdersTotal(); i++){
    if (OrderSelect(i, SELECT_BY_POS) == true){ // order exists
        if (OrderSymbol() != symbol) continue; // another security
        if (OrderType() > 1){ // pending order found
            Alert("Pending order detected. EA doesn't work.");
            return (0); // Exit start()
        }
        total++;
        if (total > 1){ // NO MORE THAN ONE ORDER CAN BE MANAGED (Metatrader rule)
            Alert("Several market orders. EA doesn't work.");
            return (0); // Exit start()
        }
        ticket = OrderTicket(); // selected ticket
        ticket_type = OrderType(); // type of selected ticket
        ticket_lots = OrderLots(); // amount of lots
    }
}

alertEquityIfChanged();

direction = getAvarageDirection();
average = getAvarage();
last_close = getLastClosingPrice();
last_fractal = getLastFractal();

bool result = manageTicket(); //close position if needed

if (ticket != -1 && result == false) return (0); //we have a pending order

int order_action = isOrderAction(last_fractal, average, last_close, before_last_close,
direction);
int oldest_bar = Bars-1;
if (order_action != -1){
    ticket = createOrder(order_action);
}
}

void updateMarketInfo()
{
    min_lot = MarketInfo(symbol, MODE_MINLOT);
    max_lot = MarketInfo(symbol, MODE_MAXLOT);
    one_lot = MarketInfo(symbol, MODE_MARGINREQUIRED);
    step = MarketInfo(symbol, MODE_LOTSTEP);
    stop_level = MarketInfo(symbol, MODE_STOPLEVEL);
}

void alertEquityIfChanged()
{
    if (last_ticket != ticket) Alert("ACCOUNT EQUITY: ",AccountEquity());
    last_ticket = ticket;
}

int createOrder(int order_type)
{
    RefreshRates();
    string symbol = Symbol();
    double free_money = AccountFreeMargin();
    double leverage = AccountLeverage();
    double lots = calculateLots(symbol, free_money, leverage, min_lot, max_lot, one_lot,
step);
    int slippage = 100;
    int ticket = -1;
    double stop_loss;

    if (lots > 0) {
        if (order_type == OP_BUY){

```

```

        stop_loss = calculateStopLoss(OP_BUY, Bid, last_fractal, stop_level);
        ticket = OrderSend(symbol, OP_BUY, lots, Ask, slippage, stop_loss, 0, "ASK", 0, 0,
Green); //Opening Buy
    }
    if (order_type == OP_SELL){
        stop_loss = calculateStopLoss(OP_SELL, Ask, last_fractal, stop_level);
        ticket = OrderSend(symbol, OP_SELL, lots, Bid, slippage, stop_loss, 0, "BID", 0, 0,
Tomato); //Opening Sell
    }
}

return (ticket);
}

bool manageTicket()
{
    bool response = false;
    double stop_loss_lower;
    double stop_loss_upper;
    double current_stop_loss;
    double macd0;
    double macd1;
    double macd_signal_0;
    double macd_signal_1;

    if (ticket == -1 || OrderSelect(ticket, SELECT_BY_TICKET)==false) return (response);

    macd0 = getMACD(0);
    macd_signal_0 = getMACD(0, MODE_SIGNAL);

    macd1 = getMACD(1);
    macd_signal_1 = getMACD(1, MODE_SIGNAL);

    if (ticket_type == OP_BUY && macd1 > macd_signal_1 && macd0 < macd_signal_0){
        response = OrderClose(ticket, ticket_lots, Ask, 50);
    }
    if (ticket_type == OP_SELL && macd1 < macd_signal_1 && macd0 > macd_signal_0){
        response = OrderClose(ticket, ticket_lots, Bid, 50);
    }

    if (!response){
        current_stop_loss = OrderStopLoss();

        stop_loss_lower = calculateStopLoss(OP_BUY, Bid, getLastFractalByMode(MODE_LOWER),
stop_level);
        stop_loss_upper = calculateStopLoss(OP_SELL, Ask, getLastFractalByMode(MODE_UPPER),
stop_level);

        if (ticket_type == OP_BUY && stop_loss_lower > current_stop_loss){
            OrderModify(ticket,OrderOpenPrice(),stop_loss_lower,OrderTakeProfit(),0,Blue);
        }
        if (ticket_type == OP_SELL && stop_loss_upper < current_stop_loss){
            OrderModify(ticket,OrderOpenPrice(),stop_loss_upper,OrderTakeProfit(),0,Blue);
        }
    }
    return (response);
}

bool isOrderAction( double last_fractal, double average, double last_closing_price, double
before_last_close, double direction )
{
    int is_action = -1;
    if (last_fractal > average && average > last_closing_price && average < before_last_close
&& direction >= 0){
        is_action = OP_SELL;
    }
    if (last_fractal < average && average < last_closing_price && average > before_last_close
&& direction <= 0){
        is_action = OP_BUY;
    }
    return (is_action);
}

double calculateLots(string symbol, double free_money, double leverage, double min_lot, double
max_lot, double one_lot, double step)
{
    double money = money_per_trade/100* free_money;

```

```

double lots = MathFloor(money/leverage/min_lot/one_lot/step)*step; //calculate lost
if (min_lot * one_lot > money_per_trade){ //too few lots
    return (0.0);
}
if (max_lot < lots){ //too much lots
    return (max_lot * one_lot);
}
if (lots * one_lot > free_money){ //not enough free money
    return (0.0);
}

if (money_per_trade == 100){ //avoid not enough money error (code 134)
    lots = lots - min_lot*2;
}

return (lots);
}

double calculateStopLoss(int type, double price, double fraktal, double stop_level)
{
    double stop_loss = fraktal; //default

    double server_stop_shift = stop_level * Point;
    double stop = MathAbs(price - fraktal);

    if (stop < min_stop_pids){ //user defined minimum stop pids
        stop = min_stop_pids;
    }

    if (stop < server_stop_shift){ //server defined minimum stop pids
        stop = server_stop_shift;
    }

    if (type == OP_BUY){
        stop_loss = price - stop;
    }
    if (type == OP_SELL){
        stop_loss = price + stop;
    }
}

return (NormalizeDouble(stop_loss, Digits));
}

double getAvarage(int shift = 0) //Simple Moving Avarage (MA)
{
    int ma_shift = 0;
    int ma_method = MODE_SMA;
    int applied_price = PRICE_CLOSE;
    return( iMA(NULL, 0, period_ma, ma_shift, ma_method, applied_price, shift) );
}

double getAvarageDirection()
{
    double avarage0 = getAvarage(0);
    double avarage1 = getAvarage(1);
    return(avarage0 - avarage1);
}

double getLastClosingPrice()
{
    return ( Close[1] );
}

double getMACD(int shift, int mode = MODE_MAIN) //Moving Avarage Convergence/Divergence (MACD)
{
    int applied_price = PRICE_CLOSE;
    return ( iMACD(NULL, 0, fast_ema_period, slow_ema_period, signal_period, applied_price,
mode, shift) );
}

double getLastFractal()
{
    int mode = MODE_UPPER;
    for (int i=1; i<=200; i++){
        if ( iFractals(NULL,0,MODE_UPPER,i) > 0 ){
            break;
        }
    }
}

```

```

        if ( iFractals(NULL,0,MODE_LOWER,i) > 0 ){
            mode = MODE_LOWER;
            break;
        }
    }
    return (iFractals(NULL, 0, mode, i));
}

double getLastFractalByMode(int mode)
{
    double fraktal;
    for (int i=1; i<=200; i++){
        fraktal = iFractals(NULL,0,mode,i);
        if ( fraktal > 0 ){
            break;
        }
    }
    return (fraktal);
}

```

Lisa 2: MA

```
//+-----+
//|                                     MA89_ma.mq4 |
//|                                     hardo |
//+-----+
#property copyright "Hardo Lass"
#property link ""

//---- indicator settings
#property indicator_chart_window
#property indicator_buffers 1
#property indicator_color1 Red
#property indicator_width1 1

//---- indicator parameters
extern int MA_Period=89;
extern int MA_Shift=0;

//---- indicator buffers
double ExtMapBuffer[];
int ExtCountedBars = 0;

//+-----+
//| Custom indicator initialization function |
//+-----+
int init()
{
    SetIndexStyle(0, DRAW_LINE);
    SetIndexShift(0, MA_Shift);
    IndicatorDigits(MarketInfo(Symbol(), MODE_DIGITS));

    IndicatorShortName("SMA(" + MA_Period + ")");
    SetIndexDrawBegin(0, MA_Period - 1);
    SetIndexBuffer(0, ExtMapBuffer);
    //---- initialization done
    return(0);
}

//+-----+
//| Moving Averages |
//+-----+
int start()
{
    if(Bars <= MA_Period) return(0);
    ExtCountedBars = IndicatorCounted();
    if (ExtCountedBars < 0) return(-1);
    if (ExtCountedBars > 0) ExtCountedBars--;

    double sum = 0;
    int i, pos = Bars - ExtCountedBars - 1;

    if(pos < MA_Period) pos = MA_Period;

    for(i = 1; i < MA_Period; i++, pos--){
        sum += Close[pos];
    }
    while(pos >= 0){
        sum += Close[pos];
        ExtMapBuffer[pos] = sum/MA_Period;
        sum -= Close[pos+MA_Period-1];
        pos--;
    }

    if(ExtCountedBars < 1){
        for(i = 1; i < MA_Period; i++) ExtMapBuffer[Bars - i] = 0;
    }
    return(0);
}
```


Lisa 3: MACD

```
//+-----+
//|                                     MA89_macd.mq4 |
//|                                     hardo |
//|                                     |
//+-----+
#property copyright "Hardo Lass"
#property link ""
//---- indicator settings
#property indicator_separate_window
#property indicator_buffers 4

#property indicator_color2 Aqua
#property indicator_color3 Red
#property indicator_color4 Red

#property indicator_width2 1
#property indicator_width3 1
#property indicator_width4 1

//---- indicator parameters
extern int FastEMA=13;
extern int SlowEMA=34;
extern int SignalSMA=8;

//---- indicator buffers
double MacdBuffer[], MacdBufferDown[], MacdBufferUp[];
double SignalBuffer[];

//+-----+
//| Custom indicator initialization function |
//+-----+
int init()
{
    SetIndexBuffer(0, MacdBuffer);
    SetIndexStyle(0, DRAW_HISTOGRAM);

    SetIndexBuffer(1, MacdBufferUp);
    SetIndexStyle(1, DRAW_HISTOGRAM);

    SetIndexBuffer(2, MacdBufferDown);
    SetIndexStyle(2, DRAW_HISTOGRAM);

    SetIndexBuffer(3, SignalBuffer);
    SetIndexStyle(3, DRAW_LINE);
    SetIndexDrawBegin(3, SignalSMA);
    IndicatorDigits(Digits + 1);

    IndicatorShortName("MACD(" + FastEMA + "," + SlowEMA + "," + SignalSMA + ")");
    SetIndexLabel(1, "MACD_UP");
    SetIndexLabel(2, "MACD_DOWN");
    SetIndexLabel(3, "Signal");

    return(0);
}

//+-----+
//| Moving Averages Convergence/Divergence |
//+-----+
int start()
{
    int limit, i;
    int counted_bars = IndicatorCounted();

    if(counted_bars > 0) counted_bars--;
    limit = Bars - counted_bars;

    for(i = 0; i < limit; i++){
        MacdBuffer[i]=iMA(NULL, 0, FastEMA, 0, MODE_EMA, PRICE_CLOSE, i)-iMA(NULL, 0, SlowEMA,
0, MODE_EMA, PRICE_CLOSE, i);
    }
    for(i = 0; i < limit; i++){
        MacdBufferDown[i]=0.0; MacdBufferUp[i]=0.0;
        if(i >= 1){
```

```
        if(MacdBuffer[i + 1] - MacdBuffer[i] >= 0){
            MacdBufferDown[i] = MacdBuffer[i];
        }
        else{
            MacdBufferUp[i]=MacdBuffer[i];
        }
    }
}
for(i = 0; i < limit; i++){
    SignalBuffer[i] = iMAOnArray(MacdBuffer, Bars, SignalSMA, 0, MODE_SMA, i);
}
return(0);
}
```

Lisa 4: Fractals

```
//+-----+
//|                                     MA89_fractals.mq4 |
//|                                     hardo |
//+-----+
#property copyright "Hardo Lass"
#property link ""

//---- indicator settings
#property indicator_chart_window
#property indicator_buffers 2
#property indicator_color1 Aqua
#property indicator_color2 Tomato
#property indicator_width1 1
#property indicator_width2 1

//---- indicator buffers
double ExtUpperBuffer[], ExtLowerBuffer[];

//+-----+
//| Custom indicator initialization function |
//+-----+
int init()
{
    SetIndexBuffer(0, ExtUpperBuffer);
    SetIndexStyle(0, DRAW_ARROW);
    SetIndexArrow(0, 218);
    SetIndexLabel(0, "Fractal Up");

    SetIndexBuffer(1, ExtLowerBuffer);
    SetIndexStyle(1, DRAW_ARROW);
    SetIndexArrow(1, 217);
    SetIndexLabel(1, "Fractal Down");

    return(0);
}

//+-----+
//| Fractals |
//+-----+
int start()
{
    int limit, i;
    int counted_bars = IndicatorCounted();

    if(counted_bars > 0) counted_bars--;
    limit = Bars - counted_bars;

    for (i = 1; i <= limit; i++){
        ExtUpperBuffer[i] = iFractals(NULL,0,MODE_UPPER,i) + 0.0007; //array y location
        ExtLowerBuffer[i] = iFractals(NULL,0,MODE_LOWER,i) - 0.0007; //array y location
    }
    return(0);
}
```