

§ 5 МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ НОВЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Милованов М. М.

РАЗРАБОТКА И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ИНСТРУМЕНТАЛЬНОЙ СИСТЕМЫ ТЕСТИРОВАНИЯ ТОРГОВОГО АЛГОРИТМА

Аннотация: Современное развитие фондового рынка предполагает развитие инфраструктуры, обновление программного обеспечения, новые инструменты. В связи с этим совершение торговых операций на бирже также должно нести в себе новые методы. Развитие современных информационных технологий позволяет использовать компьютерные ресурсы для исследования поведения акции, анализа и оптимизации алгоритма для совершения сделок на фондовом рынке по тем или иным правилам. В статье описывается разработка инструментальной системы для тестирования торговых алгоритмов. В качестве методологии использован метод наблюдения. В качестве объекта наблюдения выступает финансовый актив - акция, фьючерс, индекс. В статье описывается создание программного продукта, который, имея функционал систем разработки торговых алгоритмов, включал бы все преимущества прототипного программирования для создания, тестирования и оптимизации торговых алгоритмов, упрощая тем самым работу конечного пользователю. Разрабатываемая информационная система тестирования торговых алгоритмов имеет высокую скорость за счет использования языка программирования LUA.

Ключевые слова: анализ данных, программное обеспечение, программирование, .NET, LUA, C#, фондовый рынок, тестирование, оптимизация, разработка

Review: The current development of the stock market requires development of infrastructure, software upgrades and new tools. In this regard the commission of trading on the stock exchange must also carry out the new methods. The development of modern information technology allows using computer resources for the study of the behavior of the stock market, for analysis and optimization of transaction algorithms on the stock market based on certain rules. The article describes the development of a tool for testing trading algorithms. As methodology the authors used the method of observation. As the object of observation the authors used financial assets such as stocks, futures, indexes. The article describes building of a software product that includes the features of a system of trading algorithm development combined with all the benefits of prototype software

for creating, testing and optimization of trading algorithms, simplifying the end-user experience. The developed information system for testing trading algorithms has high performance due to use of LUA programming language.

Keywords: *stock market, C#, LUA, .NET, programming, software, data analysis, testing, optimization, development*

Существует довольно большое количество программных средств, позволяющих протестировать торговый алгоритм. Программные средства можно разделить на следующие категории:

1. Инструменты, встроенные в торговый терминал
 - Qpile – встроенный язык в наиболее популярный терминал для торговли QUIK, язык довольно просто. Однако у этой простоты помимо очевидного плюса в легком освоении, есть очевидные минусы, как ограниченный функционал, сложность отладки.
 - LUA – еще один несложный язык для терминала QUIK. Имеет больший функционал, чем Qpile. Однако большую популярность еще не приобрел. Сам код работает непосредственно в терминале QUIK.
2. Сторонние коробочные продукты со схожим функционалом, который заключается в возможности тестирования торговых систем на исторических данных, в возможности реализации алгоритмов в виде программного кода на одном из языков (обычно C#), возможность интеграции с торговым терминалом.
 - 2.1. Wealth Lab. На текущий день данная программа признается одной из самых мощных для технического анализа. Она является полноценным средством разработки, анализа и оптимизация торговых алгоритмов. Существует два варианта программы: версия Developer – для программирования и оптимизации торговых стратегий и версия PRO – с возможностью совершения торговых операций через американского брокера Fidelity Investment. Существует визуальный конструктор стратегий. Стоимость использования данной программы составляет около 800\$, а также присутствует абонентская плата за версию PRO. Язык для написания кода – C#.
 - 2.2. Tradematic. Хорошо адаптированная под русскоязычное население программа. Обладает широким функционалом, сопоставимым с лучшими торговыми системами, однако широкого распространения не имеет. Встроенный конструктор позволяет создавать стратегии без изучения языков. Доступны для написания алгоритмов языки – C#, Visual Basic, J#. JScript. Программа бесплатна для тестирования торговых алгоритмов, для подключения к бирже требуется абонентская плата.
 - 2.3. Metastock. Содержит обширную библиотеку индикаторов и средства для создания собственных формул. Программа имеет модульный принцип – модуль Expert Adviser для генерации сигналы на покупку и продажу, модуль System Tester для создания и тестирования торговые стратегии и т.д. Есть простой встроен-

- ный язык макрокоманд, который в силу своей простоты не позволяет создавать сложные гибкие системы.
- 2.4. Omega. Довольно популярная в прошлом программа со встроенным языком EasyLanguage. Программа имеет широкие гибкие возможности и настройки, однако крайне не устойчива и сложна в настройке.
 - 2.5. TS Lab. Довольно популярная программа для создания, анализа торговых систем. Имеет платную версию для торговли на бирже, однако версия для тестирования бесплатна. Встроенный визуальный редактор позволяет создавать системы, однако, сложные системы возможно создавать только с помощью языка платформы .NET.
 3. Внешние программы, разработанные программистом самостоятельно. В этом случае можно использовать любой язык программирования, который знаком человеку. Несколько лет назад пользовалась популярностью связка терминала QUIK и Microsoft Excel, в котором алгоритм реализовывался на языке Visual Basic. В настоящий момент существует русскоязычная разработка Stock#. Это библиотека для разработки торговых роботов. Основана на платформе .NET. Довольно мощная и гибкая библиотека для написания роботов, однако не программисту будет довольно сложно в ней разобраться

У всех рассмотренных программных продуктов примерно похожий функционал, однако, для конечного пользователя, не знакомого с языком программирования, использование данных продуктов значительно усложнено, а их высокая стоимость часто не позволяет воспользоваться функционалом программ. К тому же многие платформы для разработки торговых систем очень требовательны к ресурсам компьютера.

Проблема освоения таких программ в основном заключается в трудности понимания принципов объектно-ориентированного программирования, необходимости запоминать порой сложные структуры классов и т.д. Однако использование прототипного программирования в качестве объектно-ориентированной модели позволяет значительно упростить возможность программирования торговой системы.

В связи с этим появилась идея создания программного продукта, который, имея функционал систем разработки торговых алгоритмов, включал бы все преимущества прототипного программирования для создания, тестирования и оптимизации торговых алгоритмов, упрощая тем самым работу конечного пользователю.

Для этих целей наиболее подходит довольно молодой язык программирования – LUA. Этот язык несет в себе множество парадигм. Он обеспечивает довольно скромный набор базовых механизмов, которые могут быть расширены для решения различных задач, а не набор сложных жёстких спецификаций, обеспечивающих программирование в единой парадигме. В LUA нет в явном виде механизма наследования, однако его реализация возможна с помощью метатаблиц. LUA реализует набор дополнительных функций, такие как Garbage Collector (сборщик мусора), принудительное приведение типов, которое является важным для пользователей, минимальный набор примитивных типов и др. Сам язык легко адаптируется к большинству приложений.

Именно поэтому было принято решение использовать связку LUA и приложение на платформе .NET для разработки системы тестирования и оптимизации торговых алгоритмов. Структура связи разработанного на LUA торгового алгоритма и непосредственно самой программы для исполнения алгоритма, его тестирования и оптимизации довольно простая (Рис. 1)

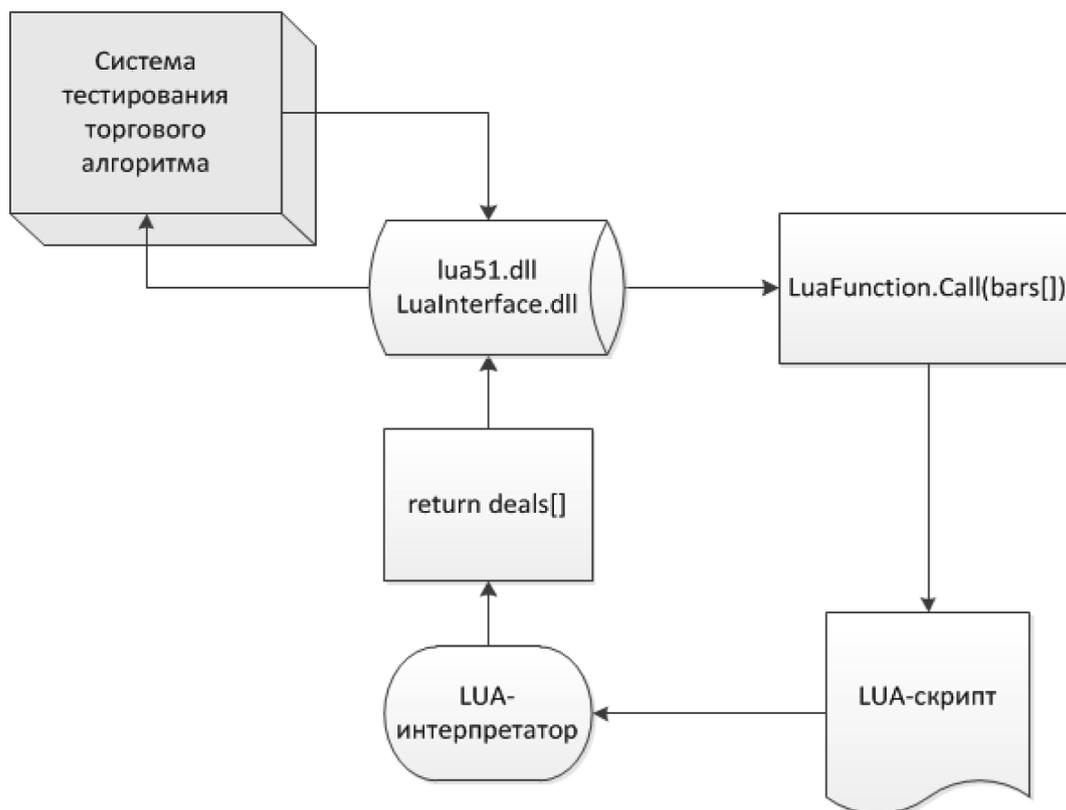


Рис. 1. Схема обмена данными между приложением и LUA-скриптом

Любой скрипт написанный на LUA должен содержать в себе метод-функцию `runStrategy(bars)`, в качестве параметров в которую передаётся таблица `bars`. Таблица `bars` представляет собой ассоциативный массив данных о свечах и имеет следующую структуру:

```

bars = {
  { year, month, day, dayOfWeek, hour, minute, second, open, high, low,
    close, volume , openInt},
  ...
}
    
```

где `year`-год, `month`-месяц, `day`-календарный день, `dayOfWeek`-день недели, `hour`-час, `minute`-минута, `second`-секунда, `open`-цена открытия свечи, `high`-максимальное значение свечи, `low`-минимальное значение свечи, `close`- цена закрытия свечи, `volume`-объем, `openInt`-открытый интерес.

Алгоритм работа программа следующий:

1. Написанный LUA-скрипт запускается на исполнение.
2. Формируется ассоциативный массив bars, получаемый из файла с набором данных о свечах.
3. Создается объект класса Lua и объект класса LuaFunction, по средствам библиотек динамических Lua51.dll и LuaInterface.dll.
4. Происходит вызов функции runStrategy() Lua-скрипта с передачей в него параметров – сформированного ассоциативного массива bars в виде Lua-таблицы.
5. Интерпретатор LUA исполняет скрипт, возвращая в приложение таблицу сделок.
6. Программа выводит на графике свечей сделки в графическом формате (Рис. 2).

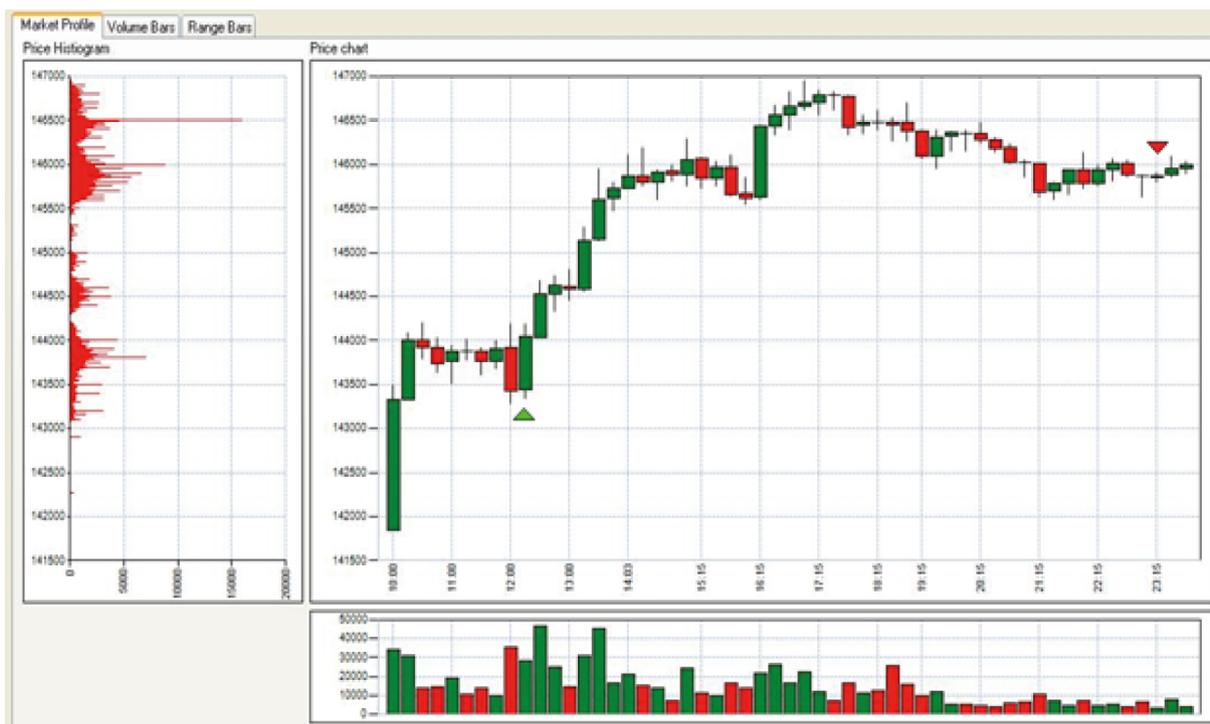


Рис. 2. Визуальный интерфейс графика зависимости цены от времени

В настоящий момент разработан механизм визуализации сделок, который включает в себя отображение сделок графика, сделок на графике [1, 2], гистограмму объемов[3], гистограмму профиля рынка (Рис. 2).

Помимо этого существует возможность запустить скрипт LUA без программной оболочки для визуализации, а лишь с помощью использования командной строки и интерпретатора LUA. Для этого необходимо лишь вызвать команду запуска интерпретатора и передать в качестве параметра имя файла (Рис. 3):

```
lua5.1.exe strategy.lua
```

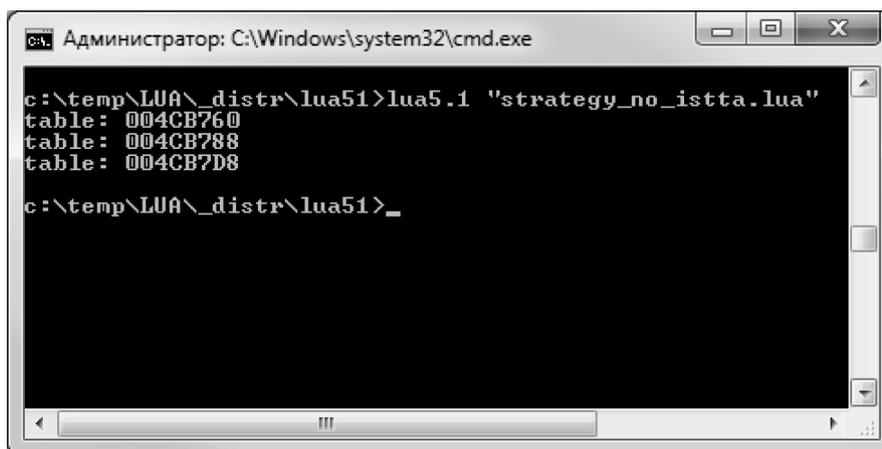


Рис. 3. Интерпретатор LUA

Как видно из Рис. 3 LUA-скрипт выполнил свою работу и вывел на экран 3 таблицы – 3 сделки. Таблица сделок, совершенных скриптом имеет следующий формат:

```
deals = {
  {year, month, day, hour, minute, second, price, count, direction},
  ...
}
```

где year-год, month-месяц, day-календарный день, hour-час, minute-минута, second-секунда, price-цена, по которой совершилась сделка, count-количество купленных акций, direction-направление сделки.

Разрабатываемая информационная система тестирования торговых алгоритмов [3] имеет преимущество по сравнению с аналогами – прежде всего это скорость. Интерпретатор LUA работает очень быстро [4], а разработка торгового алгоритма не занимает большого количества времени. Данная информационная система развивается, прорабатывается интерфейс программы, интегрируются методы визуализации сделок, реализуется механизм тестирования и оптимизация торговых алгоритмов для подбора оптимальных параметров.

Библиография :

1. Милованов М.М. Прогнозирования поведения инструментов финансовых рынков с помощью рефлексивных процессов // Электронный научный журнал “Финансы и учет”.-2014.-Выпуск 4(26) Октябрь-Декабрь. С. 21-23. [Электронный ресурс]. – Режим доступа: http://www.finance-and-accounting.ingnpublishing.com/archive/2014/release-4-26-octoberdecember/milovanov_m_m_prognozirovaniya_povedeniya_instrumentov_finansovyh_rynkov_s_pomow_yu_refleksivnyh_processov/

2. Милованов М.М. Применение рефлексивного анализа как основание для краткосрочного прогнозирования поведения финансовых рынков // Экономика и менеджмент инновационных технологий. 2015. № 2 [Электронный ресурс]. URL: <http://ekonomika.snauka.ru/2015/02/8324> (дата обращения: 07.04.2015).
3. Милованов М.М. Применение технического анализа для исследования внутридневных трендов. Теплотехника и информатика в образовании, науке и производстве: сборник докладов IV Всероссийской научно-практической конференции студентов, аспирантов и молодых учёных (ТИМ'2015) с международным участием, посвящённой 95-летию основания кафедры и университета (Екатеринбург, 26–27 марта 2015 г.). – Екатеринбург: УрФУ, 2015. – 264 с.
4. Милованов М.М. Информационные технологии. Электронный учебно-методический комплекс / Электрон. дан.-Новокузнецк: СибГИУ, 2014.:ил.-1 электронный DVD диск (DVD-ROM); № гос. регистрации 0321400939
5. Коробейников А.Г., Гришенцев А.Ю., Кутузов И.М., Пирожникова О.И., Соколов К.О., Литвинов Д.Ю. Разработка математической и имитационной моделей для расчета оценки защищенности объекта информатизации от несанкционированного физического проникновения // Кибернетика и программирование. - 2014. - 5. - С. 14 - 25. DOI: 10.7256/2306-4196.2014.5.12889. URL: http://www.e-notabene.ru/kp/article_12889.html
6. Голосовский М.С. Информационно-логическая модель процесса разработки программного обеспечения // Программные системы и вычислительные методы. - 2015. - 1. - С. 59 - 68. DOI: 10.7256/2305-6061.2015.1.14119.
7. Пономарев Д.Ю. Программная система для распределения нагрузки информационных систем // Кибернетика и программирование. - 2013. - 5. - С. 29 - 36. DOI: 10.7256/2306-4196.2013.5.9762. URL: http://www.e-notabene.ru/kp/article_9762.html
8. Лабковская Р.Я., Козлов А.С., Пирожникова О.И., Коробейников А.Г. Моделирование динамики чувствительных элементов герконов систем управления // Кибернетика и программирование. - 2014. - 5. - С. 70 - 77. DOI: 10.7256/2306-4196.2014.5.13309. URL: http://www.e-notabene.ru/kp/article_13309.html
9. В.В. Голенков, Д.В. Шункевич, И.Т. Давыденко Семантическая технология проектирования интеллектуальных решателей задач на основе агентно-ориентированного подхода // Программные системы и вычислительные методы. - 2013. - 1. - С. 82 - 94. DOI: 10.7256/2305-6061.2013.01.7.

References:

1. Milovanov M.M. Prognozirovaniya povedeniya instrumentov finansovykh rynkov s pomoshch'yu refleksivnykh protsessov // Elektronnyi nauchnyi zhurnal "Finansy i uchet".-2014.-Vypusk 4(26) Oktyabr'-Dekabr'. S. 21-23. [Elektronnyi resurs]. – Rezhim dostupa: http://www.finance-and-accounting.ingnpublishing.com/archive/2014/release-4-26-octoberdecember/milovanov_m_m_prognozirovaniya_povedeniya_instrumentov_f finansovykh_rynkov_s_pomow_yu_refleksivnyh_processsov/
2. Milovanov M.M. Primenenie refleksivnogo analiza kak osnovanie dlya kratkosrochnogo prognozirovaniya povedeniya finansovykh rynkov // Ekonomika i menedzhment innovatsionnykh tekhnologii. 2015. № 2 [Elektronnyi resurs]. URL: <http://ekonomika.snauka.ru/2015/02/8324> (data obrashcheniya: 07.04.2015).

3. Milovanov M.M. Primenenie tekhnicheskogo analiza dlya issledovaniya vnutridnevnykh trendov. Teplotekhnika i informatika v obrazovanii, nauke i proizvodstve: sbornik dokladov IV Vserossiiskoi nauchno-prakticheskoi konferentsii studentov, aspirantov i molodykh uchenykh (TIM'2015) s mezhdunarodnym uchastiem, posvyashchennoi 95-letiyu osnovaniya kafedry i universiteta (Ekaterinburg, 26–27 marta 2015 g.). – Ekaterinburg: UrFU, 2015. – 264 s.
4. Milovanov M.M. Informatsionnye tekhnologii. Elektronnyi uchebno-metodicheskii kompleks / Elektron. dan.-Novokuznetsk: SibGIU, 2014.:il.-1 elektronnyi DVD disk (DVD-ROM); № gos. registratsii 0321400939
5. Korobeinikov A.G., Grishentsev A.Yu., Kutuzov I.M., Pirozhnikova O.I., Sokolov K.O., Litvinov D.Yu. Razrabotka matematicheskoi i imitatsionnoi modelei dlya rascheta otsenki zashchishchennosti ob'ekta informatizatsii ot nesanksionirovannogo fizicheskogo proniknoveniya // Kibernetika i programmirovaniye. – 2014. – 5. – С. 14 - 25. DOI: 10.7256/2306-4196.2014.5.12889. URL: http://www.e-notabene.ru/kp/article_12889.html
6. Golosovskii M.S. Informatsionno-logicheskaya model' protsessa razrabotki programmnoho obespecheniya // Programmnye sistemy i vychislitel'nye metody. – 2015. – 1. – С. 59 - 68. DOI: 10.7256/2305-6061.2015.1.14119.
7. Ponomarev D.Yu. Programmnyaya sistema dlya raspredeleniya nagruzki informatsionnykh sistem // Kibernetika i programmirovaniye. – 2013. – 5. – С. 29 - 36. DOI: 10.7256/2306-4196.2013.5.9762. URL: http://www.e-notabene.ru/kp/article_9762.html
8. Labkovskaya R.Ya., Kozlov A.S., Pirozhnikova O.I., Korobeinikov A.G. Modelirovaniye dinamiki chuvstvitel'nykh elementov gerkonov sistem upravleniya // Kibernetika i programmirovaniye. – 2014. – 5. – С. 70 - 77. DOI: 10.7256/2306-4196.2014.5.13309. URL: http://www.e-notabene.ru/kp/article_13309.html
9. V.V. Golenkov, D.V. Shunkevich, I.T. Davydenko Semanticheskaya tekhnologiya proektirovaniya intellektual'nykh reshatel'ei zadach na osnove agentno-orientirovannogo podkhoda // Programmnye sistemy i vychislitel'nye metody. – 2013. – 1. – С. 82 - 94. DOI: 10.7256/2305-6061.2013.01.7.