Правительство Российской Федерации

Федеральное государственное автономное

Образовательное учреждение

высшего профессионального образования

«Национальный исследовательский университет

«Высшая школа экономики».

Факультет бизнес-информатики

Кафедра бизнес-аналитики

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

На тему: «Создание прототипа системы автоматического анализа и принятия решений на фондовой бирже»

Студент группы № 472

Солдатов Г. И.

Научный руководитель

Кирсанов А. П., ст.н.с.

Рецензент

Марон А. И., ст.н.с.

Москва, 2013.

Оглавление

[Введение 3](#_Toc357623514)

[Глава 1. Исследование существующих решений проблемы 6](#_Toc357623515)

[Описание торговой площадки 6](#_Toc357623516)

[Применяемые при разработке торговых роботов ИС 10](#_Toc357623517)

[Брокерские системы 11](#_Toc357623518)

[Специализированные системы анализа рыночных данных 13](#_Toc357623519)

[Неспециализированные системы анализа рыночных данных 15](#_Toc357623520)

[Способы прогнозирования цены 15](#_Toc357623521)

[Трендовые индикаторы 16](#_Toc357623522)

[Осцилляторы 20](#_Toc357623523)

[Арбитраж 22](#_Toc357623524)

[Использование данных фундаментального анализа 22](#_Toc357623525)

[Глава 2. Описание разработанного прототипа торговой системы 23](#_Toc357623526)

[Алгоритм 23](#_Toc357623527)

[Программная платформа 25](#_Toc357623528)

[Язык программирования 26](#_Toc357623529)

[Особенности реализации алгоритма 27](#_Toc357623530)

[Параметры 27](#_Toc357623531)

[Определение торгового времени 27](#_Toc357623532)

[Открытие позиций 27](#_Toc357623533)

[Добор до лимита 28](#_Toc357623534)

[Разделение функций проверки сигналов 28](#_Toc357623535)

[Вывод данных 28](#_Toc357623536)

[Назначение функций 28](#_Toc357623537)

[Итоговый алгоритм торговой системы 28](#_Toc357623538)

[Глава 3. Оценка эффективности прототипа торговой системы 31](#_Toc357623539)

[Выбор способа тестирования робота 31](#_Toc357623540)

[Алгоритм тестирования 31](#_Toc357623541)

[Тестирование торговой системы 36](#_Toc357623542)

[Заключение 40](#_Toc357623543)

[Библиографический список 42](#_Toc357623544)

[Приложение 1. Исходный код прототипа торговой системы 44](#_Toc357623545)

[Главный файл 44](#_Toc357623546)

[Вспомогательный файл lib.qpl 52](#_Toc357623547)

[Приложение 2. Исходный код программы-тестировщика 55](#_Toc357623548)

[Пояснение к коду 66](#_Toc357623549)

# Введение

За последние два десятилетия, благодаря бурному развитию информационных и телекоммуникационных технологий, возник и приобрел популярность подход к торговле ценными бумагами, заключающийся в использовании специализированных программных средств. Подобный класс ПО называют биржевыми роботами или механическими торговыми системами (МТС). По разным оценкам, в 2010 году на долю высокочастотных роботов пришлось от 60 до 80 процентов сделок на американских рынках ценных бумаг, а на российском срочном рынке FORTS – порядка половины от общего числа сделок[1].

Для эффективного ведения деятельности на бирже необходимо наличие торговой системы, эффективной в контексте получения прибыли и содержащей точный набор правил входа на рынок и выхода с него, за счет чего нивелируется неопределенность и психологический фактор.

Разработка МТС, в свою очередь, происходит с целью реализации торговой системы с помощью программных и технических средств в виде самостоятельно работающего алгоритма. Говоря об элементах роботов, можно выделить 3 основных модуля, отвевающих за получение и анализ информации, а также управление ордерами.

Для получения данных чаще всего используется терминалы систем удаленного доступа к торгам, предоставляемых брокерскими компаниями или самой биржей. В случае, если другие модули торгового робота реализуются средствами других программ, отдельное внимание необходимо уделить механизмам передачи данных, таким, как DDE или ODBC.

Анализ рыночных данных подразумевает их оценку в соответствии с торговой стратегией. Для реализации этого элемента МТС, в первую очередь, необходимо выбрать программную среду и соответствующий ей язык программирования. Кроме того, необходимо определить набор применяемых индикаторов, условия открытия и закрытия позиций, рабочий таймфрейм и ряд других деталей, относящихся непосредственно к стратегии.

Последний из основных модулей торгового робота отвечает за работу с биржевыми заявками в случае получения сигналов на открытие или закрытие позиций. Несмотря на меньшие, по сравнению с другими блоками, размеры, критически важным является обеспечение надежности его работы, поскольку сбои при работе с заявками могут оказаться не менее опасными, чем ошибки, допущенные при анализе данных.

Кроме того, допустима разработка дополнительных компонентов МТС, Так, для повышения удобства пользовательского интерфейса возможен вывод рыночных данных и совершенных сделок в таблицы или на графики. Для проведения анализа эффективности робота и корректировки его параметров имеет смысл реализовать запись и хранение статистики сделок. Еще одним вариантом для расширения функциональности является добавление блока дистанционного управления.



**Рис. 1. Элементы механической торговой системы.**

Использование роботов дает ряд преимуществ по сравнению с традиционной торговлей:

* быстрота работы, позволяющая практически моментально принимать торговые решения;
* масштабируемость, дающая, с одной стороны, возможность одновременно отслеживать динамику десятков и даже сотен ценных бумаг, а с другой – неограниченно расширять функционал робота;
* точность вычислений, исключающая появление математических ошибок, при условии правильности заложенной логики;
* строгое следование заложенной стратегии, исключающее отступ от стратегии из-за психологического или какого-либо другого фактора;
* возможность безостановочной работы на протяжении всей торговой сессии.

В то же время, торговые системы обладают рядом значительных недостатков, ограничивающих возможности по их применению:

* относительная сложность разработки, связанной с необходимостью знания языков программирования;
* уязвимость к ошибкам в коде и логике функционирования робота;
* невозможность реализации значительной части стратегий, связанных с фундаментальным анализом рынка или с использованием интуитивного подхода;
* отсутствие универсальности из-за непостоянства динамики ценных бумаг.

Суммируя эти особенности, можно сказать, что биржевые роботы являются полезным инструментом для участников торгов. В то же время, их применение сопряжено с рядом дополнительных рисков, в связи с чем, необходим постоянный контроль над их работой со стороны человека.

Целью настоящей выпускной работы является разработка и оценка эффективности нового алгоритма прогнозирования цен на фондовых рынках. Её выполнение подразумевает решение следующих задач:

* исследование существующих методов прогнозирования цен, а также программных и технических средств для реализации прототипа МТС;
* выбор средств реализации и алгоритма с последующей разработкой прототипа торговой системы на их основе;
* тестирование прототипа.

Объектом исследования является Московская биржа (бывшая ММВБ-РТС). В качестве же предмета исследования выступают существующие средства для ведения автоматизированной торговли и применяемые в ней алгоритмы.

Структурно, выпускная работа описывает выполнение поставленных задач в порядке их перечисления. Каждой из задач соответствует одна глава ВКР.

# Глава 1. Исследование существующих решений проблемы

## Описание торговой площадки

В настоящее время Московская биржа лидирует по оборотам торгов не только в России, но и во всей Восточной Европе. Структура биржи объединяет пять рынков, отличающихся по находящимся в обращении ценным бумагам[2]:

* фондовый (состоит из секторов «Основной рынок», «STANDARD», «Classica»);
* срочный;
* валютный;
* денежный;
* товарный.

Участниками торгов на валютном и денежном рынках, а также в секторе «Classica» выступают профессиональные участники рынка, являющиеся юридическими лицами, – дилеры, институциональные инвесторы, управляющие компании. Товарный рынок, в свою очередь, представляет интерес для организаций, относящихся к реальному сектору экономики.

Практический интерес для частного инвестора, таким образом, представляют оставшиеся два сектора фондового и срочный рынок FORTS. Доступ к торгам можно получить путем заключения договора о брокерском обслуживании с обладающей соответствующей лицензией компанией.

Спектр торгуемых на фондовом рынке ценных бумаг включает акции, государственные и корпоративные облигации, ПИФы и депозитарные расписки; на рынке FORTS обращаются производные ценные бумаги – фьючерсы и опционы.

Торги на Московской бирже проводятся полностью в электронном формате. Центральным элементом обеспечивающей инфраструктуры всех секторов биржи, за исключением срочного рынка, является торгово-клиринговая система ASTS. Функциональность этого программного комплекса включает:

* проведение торгов различными ценными бумагами и в соответствии с различными правилами;
* депозитарный учет средств и ценных бумаг участников торгов, а также проведение клиринговых расчетов;
* обеспечение полного жизненного цикла ценных бумаг;
* взаимодействие с внешними информационными системами;
* предоставление доступа удаленным пользователям.

Эти же задачи для срочного рынка решает другая ИС, также называющаяся FORTS.

В настоящее время существует шесть схем подключения к информационной системе Московской биржи[3]:

1. Прямое подключение через терминалы ММВБ. Биржа предлагает около десятка программных решений для участия в торгах и обмена данными.
2. Подключение с использованием внешнего программно-технического средства (ВПТС). Для подключения новой информационной системы требуется разрешение биржи. К категории ВПТС могут относиться как сторонние программы, так и собственные разработки участника торгов.
3. Подключение ВПТС с использованием сервера доступа. Отличием от предыдущей схемы является расположение данного сервера на стороне клиента, а не биржи, что позволяет оптимизировать затраты времени и использование каналов связи.
4. Подключение ВПТС через интернет. Сервера доступа в данном варианте не применяются.
5. Размещение торговых роботов в дата-центре биржи. Основным преимуществом колокации является уменьшение времени, необходимого для отправки торговых приказов и получения информации, что очень востребовано, в частности, при высокочастотной торговле и маркет-мейкинге.
6. Подключение по протоколу Financial Information Exchange (FIX). Этот протокол широко применяется во всем мире для передачи финансовой информации и проведения транзакций.

Данные схемы показывают способы обмена данными между биржей и профессиональными участниками рынка, поскольку физические лица не имеют возможности прямого выхода к торгам.

Варианты подключения к информационной системе FORTS, в целом, аналогичны представленным выше, несмотря на отличия в применяемых программных и технических средствах.

Брокерские информационные системы, являющиеся основным типом средств для обмена данными с биржей, рассматриваемым в данной работе, относятся к категории ВПТС. Типовая архитектура ИС и их связь с биржей на примере системы ASTS представлены на рис. 2[3].



**Рис. 2. Подключение к ASTS с использованием ВПТС.**

## Применяемые при разработке торговых роботов ИС

В данном разделе рассматривается часть программных средств, применяемых при разработке, тестировании и эксплуатации МТС. Исследование будет проводиться с позиции частного инвестора, и, соответственно, сравниваться будут системы, используемые при торговле на фондовом и срочном рынках Московской биржи. В целом, данное программное обеспечение можно разделить на две группы:

* брокерские ИС;
* системы анализа данных.

### Брокерские системы

Основной задачей данного типа программного обеспечения является обеспечение доступа к торговым площадкам клиентов брокерских компаний. Возможности этих систем включают:

* предоставление котировок, стаканов цен, сделок и иных рыночных данных;
* отправка и контроль клиентских заявок;
* учет клиентских портфелей и лимитов по операциям;
* информационная поддержка: новостные ленты, связь с брокером;
* обеспечение информационной безопасности.

Для ведения алгоритмической торговли брокерские системы предоставляют набор встроенных технических индикаторов и средств графического анализа, встроенные языки программирования, а также интерфейсы экспорта и импорта данных. В совокупности, возможности этих систем, как правило, позволяют реализовать все основные элементы МТС.

Стандартной схемой, по которой строится архитектура подобных информационных систем, является двухуровневая клиент-серверная модель. Клиентская часть включает как пользовательские терминалы, так и служебные, предназначенные для системных администраторов и иных сотрудников бэк-офиса.

В рамках выпускной работы рассмотрены следующие системы брокерского обслуживания: Quik, NetInvestor, TRANSAQ, MetaTrader 5. Их сравнительная характеристика представлена в таблице 1[4] [5] [6] [7].

**Табл. 1. Реализация возможностей в исследуемых ИС.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Функционал | Quik | NetInvestor | TRANSAQ | MetaTrader 5 |
| Доступ на фондовый рынок | Есть | Есть | Есть | Нет |
| Доступ на FORTS | Есть | Есть | Есть | Есть[9] |
| Маржиналь-ная торговля | Торговля «с плечом», короткие позиции | Торговля «с плечом», короткие позиции[8] | Торговля «с плечом», короткие позиции | Торговля «с плечом», короткие позиции[9] |
| Условные заявки | Стоп-лосс, тейк-профит, комбинированная | Стоп-лосс, тейк-профит[8] | Стоп-лосс, тейк-профит, комбинированная[10] | Стоп-лосс, тейк-профит[11] |
| Язык программи-рования | Встроенный QPILE, поддержка скриптов на Lua | Нет | ATF | MQL5 |
| Создание пользователь-ских индикаторов | Только в коде, без вывода на график | Нет | Только в коде, без вывода на график | В коде и с выводом на график; имеются базы готовых индикаторов |
| Экспорт и импорт данных | DDE, ODBC, импорт транзакций | DDE, ODBC, импорт транзакций[12] | DDE, DBC Signal, импорт транзакций | В файлы с расширениями .XML, .CSV, .XLSX |
| Связь с системами анализа данных | Wealth-Lab, MetaStock, S#, TSLab, Excel и пр. | MetaStock, TradeStation, Excel | Wealth-Lab, MetaStock, TradeStation, S#, Excel | Excel |

Каждая из рассмотренных информационных систем предоставляет схожий набор базовых возможностей, достаточный для ведения алгоритмической торговли. Различия между ними находятся при более детальном изучении, поэтому предпочтение той или иной системе можно отдать только в случае наличия конкретных требований к разрабатываемому роботу, к примеру, применение определенного технического индикатора.

### Специализированные системы анализа рыночных данных

Подобные программные средства обычно используются в связке с брокерскими системами, поскольку сами, как правило, не предоставляют выход к торговой площадке. В этом случае, с их помощью реализуется аналитический модуль робота. В то же время, некоторые системы позволяют подключаться непосредственно к бирже, однако эта возможность используется только профессиональными участниками торгов. Далее будут рассмотрены некоторые программные продукты, относящиеся к этой группе.

1. StockSharp[13]. Данная платформа, реализованная на базе языка программирования C#, предназначена для разработки торговых роботов и других связанных биржей программ. S# предоставляет прямой доступ к торгам через биржевые шлюзы CGate и MICEX Bridge. Кроме того, StockSharp содержит более полутора десятков интерфейсов для брокерских систем, биржевых терминалов и аналитических программ. Еще одной особенностью платформы является возможность использовать при разработке МТС весь функционал .NET. В настоящее время, в состав платформы входит несколько элементов.

**Табл. 2. Элементы платформы S#.**

|  |  |
| --- | --- |
| Название | Описание |
| S#.Studio | Программная среда для разработки, управления и тестирования МТС. Обладает графическим интерфейсом. Имеется возможность создания собственных индикаторов и обмена данными с другими ИС. |
| S#.API | Библиотека под .NET для разработки МТС. |
| S#.Data | Утилита для хранения и обработки исторических данных торгов. |
| S#.Notify | Платный сервис для отправки уведомлений по SMS и электронной почте. |
| S#.MatLab, S#.Wealth-Lab | Интерфейсы для интеграции с соответствующими программными средствами. |

1. Wealth-Lab[14]. Этот программный комплекс является одним наиболее популярных средств для технического анализа рыночных данных, а также разработки и тестирования МТС. Отличительными чертой системы является её модульность, что позволяет увеличить функциональность за счет разработанных пользователями расширений. К таким дополнениям относятся технические индикаторы, готовые стратегии и подключаемые источники торговых данных.

Для разработки МТС Wealth-Lab предоставляет несколько вариантов. Первый из них – графический редактор, не требующий навыков программирования. Второй вариант заключается в использовании встроенного языка WealthScript. Наконец, последние версии программы совместимы с .NET, однако большинство российских систем заточено под работу с вышедшей более пяти лет назад четвертой версией программы, в которой эта возможность еще не реализована.

Процесс тестирования торговых систем можно проводить как на одном инструменте, так и на портфеле, а оценивать их эффективность – с помощью разнообразных числовых и графических индикаторов.

### Неспециализированные системы анализа рыночных данных

В некоторых случаях, с целью реализации или тестирования торгового алгоритма применяются неспециализированных программных средств. К их числу можно отнести MATLAB, Mathcad и MS Excel.

Преимуществом этих систем является наличие мощного математического и статистического аппарата и возможность применения собственных скриптовых языков.

Вместе с тем, подобные программы не заточены конкретно под биржевую торговлю, поэтому реализация элементов бизнес-логики, таких, как работа с клиентским портфелем или управление заявками, сопряжена с дополнительными трудностями.

## Способы прогнозирования цены

Технический анализ, помимо применения индикаторов, предлагает и иные способы прогнозирования цен: выявление типовых фигур на графиках с помощью уровней поддержки и сопротивления или японских свечей, выявление циклических и волновых зависимостей, например, с помощью теорий Фибоначчи и Эллиота и т.д. Однако, формализация их в виде алгоритма довольно сложна, поскольку требует создания сложной логики и поиска оптимальных значений для входных параметров. В связи с этим, в данном разделе работы будут рассматриваться наиболее распространенные методы анализа, относящиеся к категории индикаторов.

На практике применяется значительное число технических индикаторов, но, тем не менее, большинство из них можно отнести к одной из двух групп[16]:

1. Трендовые индикаторы.
2. Контртрендовые индикаторы (осцилляторы).

Существующие торговые ИС предоставляют, в среднем, несколько десятков встроенных методов анализа. В силу специфики выпускной работы, упор будет делаться на изучение небольшого числа индикаторов с целью исследования принципов их построения и применения.

### Трендовые индикаторы

К этой группе можно отнести методы, рассчитанные на определение наличия тенденции в ценовом движении. Вместе с тем, они генерируют сигналы с опозданием, поэтому непригодны для определения начала или окончания трендов.

Одним из индикаторов, относящихся к данной категории, является скользящее среднее (moving average). Метод основывается на группе одноименных функций, значения которых определяются путем усреднения нескольких последних значений некоторой исходной функции[17]. На практике используется несколько видов скользящих средних, к числу которых относятся:

* простое, значения которого являются средним арифметическим определенного числа закрывающих значений свечей на графике цены:

*где – скользящее среднее в момент t,*

*– период расчета MA.*

* взвешенное, при расчете которого значения цен берутся с весовыми коэффициентами, определяемыми с помощью арифметической прогрессии, причем вес значений убывает от последнего к первому:

*где – значение индикатора в момент времени t,*

*– число рассматриваемых свечей и вес последней цены закрытия,*

*а шаг прогрессии равен единице.*

* экспоненциальное, также учитывающее цены закрытия с весами, однако сами весовые коэффициенты убывают не линейно, а в геометрической прогрессии:

*где – экспоненциальное среднее в момент времени t,*

*– весовой коэффициент, .*

Использование скользящих средних позволяет отсечь случайные колебания цены и определить направление её движения. Вместе с тем, они неэффективны при боковом движении рынка.

Еще одним трендовым индикатором система Parabolic SAR. Для его расчета, в зависимости от направления движения цены, применяются две формулы[18]. При восходящем тренде используется формула:

,

*где – текущее значение индикатора,*

*– наивысшая цена на предыдущем периоде,*

*– значение индикатора для предыдущего периода,*

*- фактор ускорения.*

При падении цены, применяется формула, имеющая вид:

,

*где – наименьшая цена за прошлый период.*

Фактор ускорения рассчитывается следующим образом:

,

*где – минимальное значение, обычно* 0.02,

*– количество экстремальных значений цены, начиная с разворота индикатора,*

*– шаг изменения цены, обычно* 0.02*.*

Также фактор ускорения имеет максимально допустимое значение, как правило, равное 0.2. Значение параболика опережает движение цены до тех пор, пока продолжается тренд. В момент слома тенденции цена догоняет индикатор, и тот начинает рассчитываться по формуле для противоположного тренда. Таким образом, индикатор показывает момент, в который стоит закрывать открытые позиции. Как и скользящие средние, Parabolic SAR эффективен только при наличии восходящего или нисходящего трендов.



**Рис. 3. Графическое представление индикатора Parabolic SAR.**

К числу трендовых также можно отнести различные канальные индикаторы. Довольно широкое распространение получил анализ с помощью полос Боллинджера (Bollinger Bands). Индикатор состоит из трех линий[19]. В качестве центральной используется обычная скользящая средняя. Значения верхней и нижней линий рассчитываются, соответственно, как сумма и разность значения центральной и стандартного отклонения цены закрытия за период, соответствующий периоду скользящей средней.

Существуют различные варианты использования полос Боллинджера. С одной стороны, границы канала приближенно показывают пределы для возможного краткосрочного движения цены, а сама она практически всегда находится между ними. Поэтому при достижении границы цена быстро вернется обратно в канал. С другой стороны, пробой верхней или нижней линии может служить сигналом наличия тренда.



**Рис. 4. Графическое представление полос Боллинджера.**

### Осцилляторы

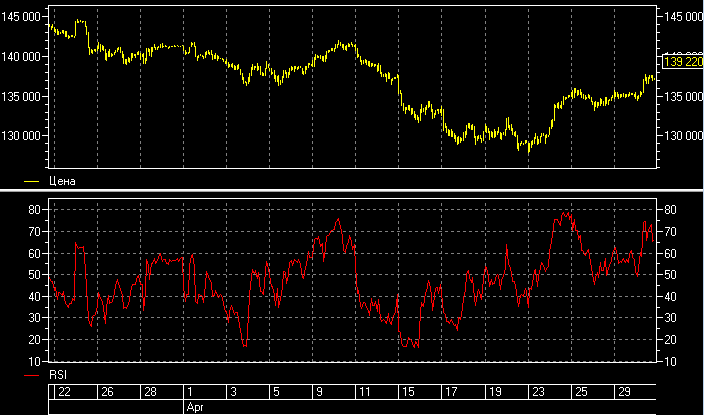
В противоположность трендовым, данный тип индикаторов используется для поиска момента окончания тренда с целью открытия позиции в противоположную сторону. Сигналы генерируются при сильной перепроданности или перекупленности рынка. Основанные на них торговые стратегии, как правило, дают лучший результат при боковом движении рынка.

Как следует из названия, особенностью осцилляторов является их колебание относительно определенного уровня и, как правило, в рамках заданных границ. Довольно часто применяется шкала от 0 до 100. Примером такого индекса является индекс относительной силы (RSI). Этот осциллятор рассчитывается по следующей формуле[20]:

*где – средний прирост за N периодов,*

*– среднее падение за N периодов.*

Значения *AG* и *AL* рассчитываются как среднее изменение цен для свечей, закрывшихся, соответственно, ростом и падением.



**Рис. 5. Графическое представление индикатора RSI.**

Значения, близкие к 50, указывают на нейтральные настроения на рынке. При нахождении индекса в пограничных зонах можно говорить о перепроданности или перекупленности рынка. Как правило, им соответствуют интервалы от 0 до 20 и от 80 до 100.

Еще один индикатор, являющийся осциллятором, основан на отслеживании схождении и расхождении скользящих средних и называется MACD. Это средство анализа включает в себя две линии[21]. Первая из них, линия MACD, считается как разность короткой и длинной скользящих средних, по умолчанию, с периодами 12 и 26. Вторая, сигнальная линия – скользящее среднее первой, период, по умолчанию, равен 9.



**Рис. 6. Графическое представление осциллятора MACD.**

При использовании MACD в торговой системе можно отслеживать:

* пересечение линии MACD с сигнальной: когда значение первой становится больше второго, возможно открытие длинной позиции, и наоборот;
* пересечение линией MACD нулевой отметки: смена знака индикатора на положительный служит сигналом к покупке, а на отрицательный – к продаже.

Недостатком индикатора является запаздывание сигналов, что делает его неэффективным при отсутствии сильного тренда.

### Арбитраж

Арбитражные торговые стратегии предполагают одновременное совершение двух или нескольких сделок, как на покупку, так и на продажу, с целью получения прибыли за счет разницы в цене. Объектом сделок выступают бумаги, стоимость которых сильно коррелирует. В результате, большую часть времени прибыль от одной операции эквивалентна потерям от другой. Поэтому арбитраж относят к рыночно-нейтральным стратегиям. Закрытие позиций осуществляется при наличии суммарной прибыли вследствие отклонения цен на активы.

Как правило, выделяют несколько видов арбитражных стратегий:

1. Торговля одинаковыми ценными бумагами на разных рынках.
2. Торговля производной ценной бумагой и ее базовым активом.
3. Торговля производными бумагами с разным сроком исполнения.

### Использование данных фундаментального анализа

Еще одним подходом для ведения алгоритмической торговли является отслеживание информации, представляющей интерес с точки зрения фундаментального анализа, такой, как макроэкономические показатели или статистические данные. Зачастую, их публикация оказывает сильное влияние на рынок.

Торговые алгоритмы могут либо получать соответствующие данные в момент их публикации и сравнивать с прогнозными, либо просто выставлять заявки в обе стороны в расчете на сильное изменение цены.

В то же время, не все фундаментальные факторы могут быть интерпретированы и, соответственно, учтены такими роботами, поэтому их применение также сопряжено с возможностью принятия ошибочных решений.

# Глава 2. Описание разработанного прототипа торговой системы

## Алгоритм

Поскольку эффективность торговой системы не является основным критерием достижения цели работы, выбор используемого индикатора не является принципиальным. Поэтому в качестве базового средства анализа будут применяться две экспоненциальные скользящие с различными периодами. Разрабатываемая система также рассчитана на открытие как длинных, так и коротких позиций.

Стандартным сигналом на покупку для такого метода анализа считается пересечение короткой скользящей средней длинной снизу вверх, тогда как на продажу – сверху вниз.



**Рис. 7. Сигналы на открытие позиций при пересечении скользящих средних.**

Недостатком данного индикатора является частая генерация ложных сигналов при боковом движении цены или краткосрочных коррекциях к основному тренду. С целью уменьшения числа убыточных сделок предполагается модифицировать условия генерации сигналов. Для появления сигнала необходимо не просто пересечение скользящих средних, но достижение их разности определенной величины. Иными словами, условие для открытия длинной позиции будет выглядеть так:

,

*где – значение длинной скользящей средней,*

*– значение короткой скользящей средней,*

*– требуемая разность величин скользящих средних,*

*– текущий период времени.*

Условие закрытия длинной позиции выглядит аналогично, однако для увеличения гибкости системы в неравенствах используется другой уровень разности:

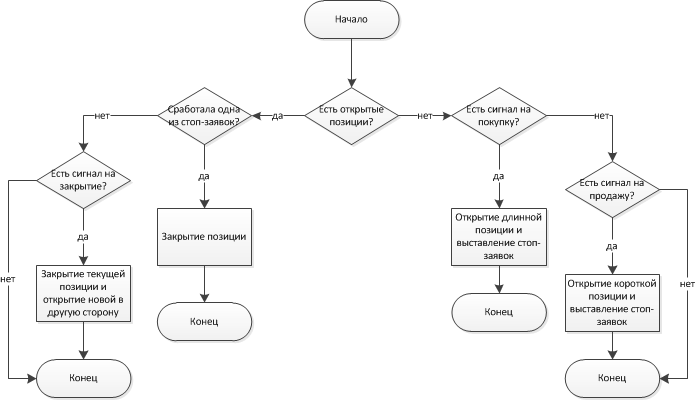
,

*где – требуемая для закрытия разность величин скользящих средних.*

Условия для открытия и закрытия коротких позиций выглядят идентично, с той лишь разницей, что значение длинной скользящей средней больше, чем у короткой.

Кроме того, после открытия позиций следует выставлять стоп-заявки, активирующиеся только в случае достижения ценой определенных уровней. Это, с одной стороны, позволит ограничить возможные убытки при неблагоприятных движениях рынка, а с другой – закрывать прибыльные позиции до перелома тренда и на более выгодных условиях.

Таким образом, общая логика робота выглядит следующим образом:



**Рис. 8. Логика торговой системы.**

## Программная платформа

В целом, каждая из предоставляемых брокерами информационных систем обладает необходимыми для разработки МТС возможностями – обменом данными с биржей и инструментарием для ведения автоматизированного анализа рынка с применением технических индикаторов. Использование дополнительных программ для технического анализа имеет свои преимущества при решении определенных задач, однако в данной работе оно представляется избыточным.

В качестве базиса для реализации торговой стратегии будет применяться программный комплекс Quik. К причинам выбора данной информационной системы можно отнести:

* наличие встроенных средств технического анализа;
* возможность работы с условными заявками нескольких типов;
* поддержка маржинальной торговли;
* возможность экспорта данных по DDE и ODBC с целью их дальнейшей обработки;
* наличие встроенного языка QPILE и поддержка Lua-скриптов.

При выборе языка программирования итоговое решение было сделано в пользу QPILE, поскольку в настоящее время библиотеки для Lua все еще находятся в состоянии разработки и не предоставляют некоторые функции. Кроме того, для QPILE существуют программные средства для написания исходного кода такие, как QPILE Master или плагин для Notepad++[22], а процесс отладки робота можно проводить в самом клиенте Quik.

## Язык программирования

Формальным предназначением QPILE является разработка программируемых таблиц, содержащих определенные пользователем данные, но, фактически, он предоставляет широкий спектр возможностей для создания торговых роботов[23]. С помощью языка возможна также обработка информации из системных таблиц и, как следствие, взаимодействие с графиками, выставление и контроль заявок, экспорт и импорт данных.

Структура программы, написанной на QPILE, включает три части: заголовок, хранящий название и основные параметры, тело, содержащее сам код, и описание столбцов программируемой таблицы. Также возможно подключение дополнительных файлов к основному.

Выполнение загруженных в Quik алгоритмов после их запуска происходит с определенной периодичностью, заданной пользователем, до отмены или приостановки работы. Таким образом, QPILE позволяет постоянно отслеживать динамику на рынке и своевременно выставлять заявки.

Исходный код разработанной торговой системы состоит из основной части и функций. При расчете алгоритма происходит выполнение логики, заложенной в основную часть, включая функции, которые в ней вызываются.

## Особенности реализации алгоритма

### Параметры

Параметры торговой системы задаются в основной части программы. Их можно разделить на следующие категории:

* информация о бумаге и клиенте (код клиента и его счета, коды бумаги, тэги графиков);
* параметры логики (таймфрейм, лимит открытых позиций, параметры условий открытия и закрытия позиций);
* параметры заявок (уровни тейк-профитов и стоп-лоссов и др.);
* вспомогательные глобальные переменные для хранения информации в течение всего расчета программы.

### Определение торгового времени

Для устранения возможных ошибок и неблагоприятных сделок торговля осуществляется только в определенный диапазон времени. В частности, пропускается небольшой объем времени после начала дневной и вечерней торговой сессии, что позволяет отсеять часть ложных сигналов, возникающих из-за разницы между уровнями цен на соседних свечах графика.

Аналогично, с целью избегания убытков, происходит закрытие позиций в конце дня.

### **Открытие** позиций

Логика робота исключает возможность повторного открытия позиции на одном таймфрейме в случае быстрого срабатывания условных заявок. Таким образом, уменьшается размер убытков при получении сигнала непосредственно перед отскоком цены в неблагоприятном направлении или сломом тренда.

### Добор до лимита

Алгоритм также реализует автоматическую закупку ценных бумаг до лимита, установленного пользователем. Это позволяет избежать ошибок в ситуациях, когда на рынке отсутствует достаточный объем встречных заявок.

### Разделение функций проверки сигналов

В торговой системе используются отдельные функции для определения точек входа и выхода, что дает дополнительную гибкость при ее настройке.

### Вывод данных

Данные о совершенных сделках выводятся в пользовательскую таблицу, что дает возможности анализа работы алгоритма и ведения статистики.

### Назначение функций

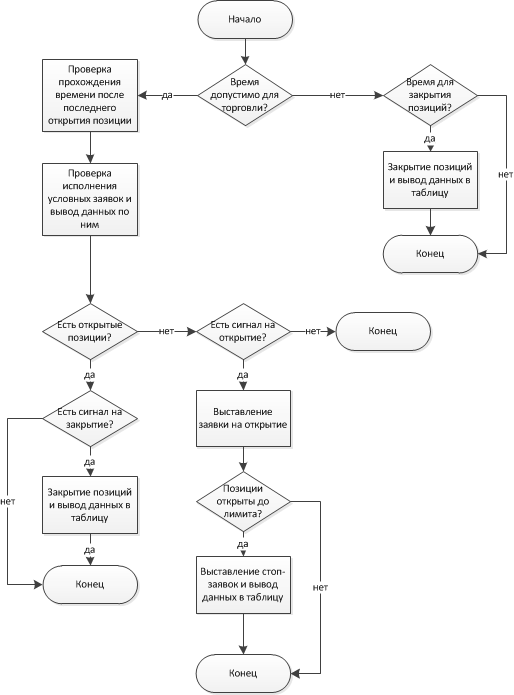
Основная функции робота располагаются в том же файле, что и его главная часть. Некоторые вспомогательные функции, не связанные с логикой МТС, вынесены в дополнительный файл lib.qpl. В целом, функции используются для:

* получения даты, времени;
* проверки допустимости проведения операций;
* получения значений из графиков и системных таблиц;
* выставления, контроля и снятия заявок;
* выполнения вспомогательных математических и иных операций;
* вывода данных в таблицу.

### Итоговый алгоритм торговой системы

Общая логика разработанного робота с привязкой к средствам реализации имеет несколько отличий по сравнению с описанной выше. Это связано, в частности, с автоматической обработкой условных заявок серверной частью Quik, а также c наличием указанных особенностей в реализации. Таким образом, торговый алгоритм работает по следующей схеме:

1. Проверка времени на допустимость ведения торговли.
   1. При прохождении проверки происходит переход к пункту 2.
   2. В конце дня производится закрытие текущих позиций.
   3. Если проверка не пройдена, никакие действия не предпринимаются.
2. Проверка окончания таймфрейма, на котором совершена последняя сделка во избежание повторного открытия позиции.
3. Проверка исполнения условных заявок и вывод статистики по ним.
4. Проверка на наличие текущих открытых позиций.
   1. Если позиции отсутствуют, происходит переход к пункту 5.
   2. Если позиции имеются, следующим выполняется пункт 6.
5. Проверка сигнала на открытие позиции.
   1. При отсутствии сигналов расчет программы завершается.
   2. При наличии сигналов происходит выставление заявки на открытие позиции.
   3. Если открыта максимально допустимая позиция, выставляются условные заявки на закрытие, и выводится статистика по последней(-им) сделке(-ам).
6. Проверка сигнала на закрытие. При наличии сигнала происходит закрытие позиций и вывод статистики.



**Рис. 9. Итоговый алгоритм торговой системы.**

# Глава 3. Оценка эффективности прототипа торговой системы

## Выбор способа тестирования робота

На практике применяется три подхода к проверке эффективности торгового алгоритма: на исторических данных (бэк-тестинг), на демонстрационном счете и в ходе реальных торгов. Второй и третий способы требуют значительного объема времени, а последний, кроме того, еще и сопряжен с возможностью несения убытков. Поэтому наиболее подходящей в рамках данной работы является проверка на истории торгов.

Проводится бэк-тестинг, как правило, с помощью аналитических программ, обладающих соответствующим функционалом (Wealth-Lab и пр.). Однако большинство подобного программного обеспечения является платным. К тому же, используя их, необходимо будет заново описывать логику алгоритма на другом языке программирования и определять механизм получения торговых данных.

По этим причинам, тестирование торговой системы будет проводиться с помощью собственного тестировщика, также базирующегося на Quik и QPILE. Это упростит процесс его разработки и эксплуатации за счет повторного использования части кода и наличия данных анализа.

## Алгоритм тестирования

В результате проверки требуется оценить эффективность торговой системы при тех или иных входных параметрах. Иными словами, необходимо сопоставить следующую информацию:

* входные данные МТС: уровни условных заявок тейк-профит и стоп-лосс, разности между скользящими средними при открытии и закрытии;
* результаты, показанные торговой системой: количество всех и убыточных сделок, конечная прибыль или убыток.

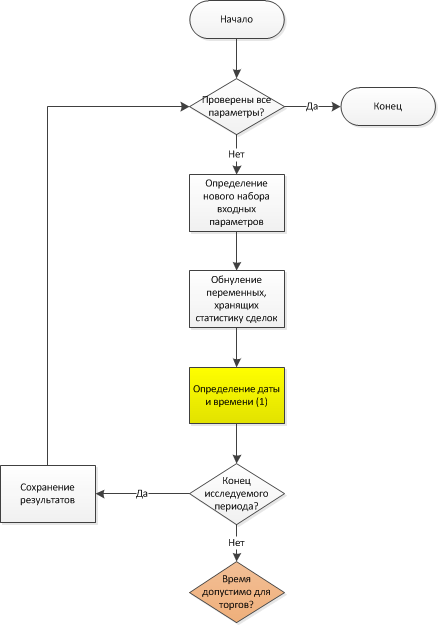
В ходе работы проверяющего алгоритма производится проверка всех комбинаций входных параметров. Для каждого из них указываются минимальная и максимальная величина проверяемого диапазона, а также шаг изменения при переборе значений. Итоговая прибыль рассчитывается из предположения, что ведется торговля одним лотом ценных бумаг.

При анализе рыночных данных тестировщик, фактически, выполняет те же действия, что и сам робот. Однако при обращении к графикам требуется получать не текущие данные, а последовательно перебирать их значения за определенный период времени. В связи с этим, алгоритм тестирования содержит измененный механизм получения даты и времени.

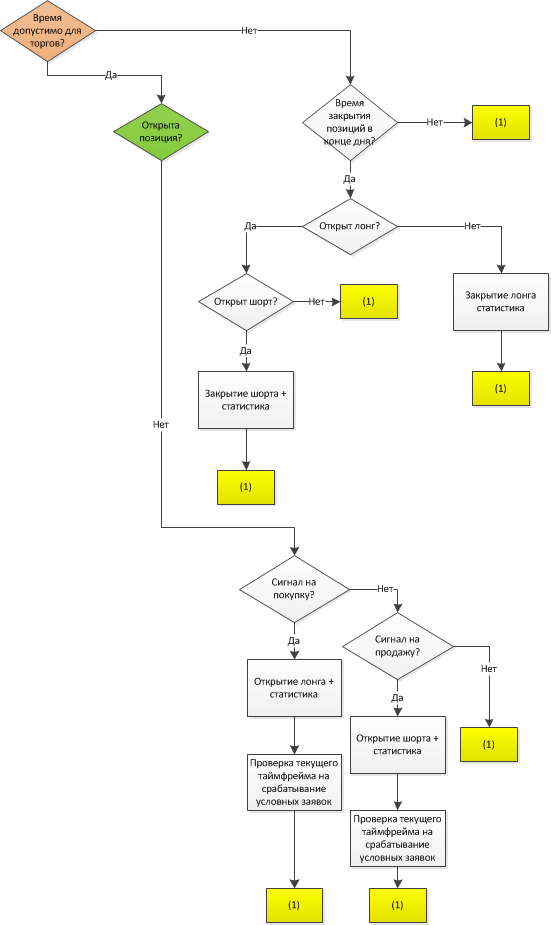
При получении сигналов на открытие позиций от модуля анализа данных тестировщик сохраняет цену и объем гипотетической сделки. В случае если в рассматриваемый момент времени имеется открытая позиция, проверяется, могла ли она закрыться. Причиной закрытия может выступать как получение сигнала, так и срабатывание одной из условных заявок. В момент выхода с рынка определяется успешность сделки и полученную в результате нее прибыль.

По завершению анализа требуемого периода времени необходимо сохранить результаты и продолжить выполнение алгоритма для других входных параметров.

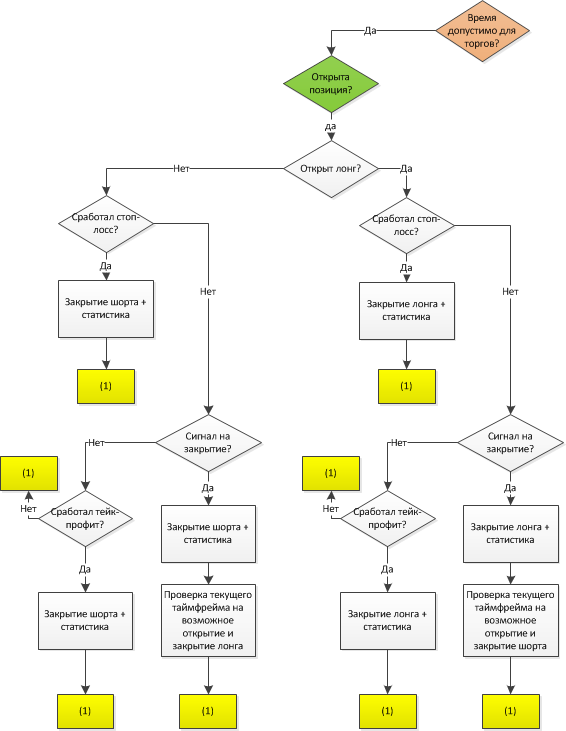
Общая схема алгоритма тестирования представлена ниже.



**Рис. 10. Алгоритм тестировщика. Определение параметров, даты и времени.**

****

**Рис. 11. Алгоритм тестировщика. Проверка на совершение сделок, ч. 1.**

****

**Рис. 12. Алгоритм тестировщика. Проверка на совершение сделок, ч. 2.**

В целом, разработанный метод тестирования позволяет оценить эффективность торговой системы, однако имеется ряд факторов, ограничивающих его точность. К их числу можно отнести следующее:

1. Не учитывается динамика цены внутри таймфрейма. Таким образом, невозможно определить точные время и цену, по которым совершаются сделки. Использование тиковых графиков невозможно в силу того, что тестирование предполагается проводить на бумагах срочного рынка, а минимально допустимый таймфрейм для них в Quik составляет 1 минуту, из-за чего уже теряется точность. Еще одним следствием этого ограничения является невозможность учета двух параметров для условных заявок: защитного спрэда и отступа от максимума (для тейк-профитов).
2. Открытие и закрытие позиций по сигналу производится по цене открытия свечи, следующей за той, которая закрылась с сигналом. Следствием этого является задержка при покупке и возможное игнорирование части ложных сигналов, возникающих на свечах с большими тенями, что завышает итоговую оценку.
3. Порядок проверки на закрытие позиции выглядит следующим образом: вначале проверяется срабатывание стоп-лосса, затем – закрытие по сигналу, после чего – исполнение тейк-профита. Эта очередность гарантирует, что закрытие произойдет с наихудшим из возможных вариантов, что в некоторой степени компенсирует вторую особенность тестировщика.

Для детального анализа совершенных сделок будет использоваться отдельная версия тестировщика. Главным отличием по сравнению с описанной выше являются проверка только одного набора значений и вывод данных по каждому закрытию позиций.

## Тестирование торговой системы

В качестве пробного испытания будет произведена проверка работы МТС на июньском фьючерсе на индекс РТС (RTS-6.13). Временной интервал тестирования – с 15.04.13 по 15.05.13, таймфрейм графика – 10 минут. Периоды скользящих средних составляют 5 и 20. Оцениваются следующие значения входных данных (указаны в ценовых пунктах):

* уровни тейк-профита и стоп-лосса – от 200 до 300, шаг – 50;
* разность между скользящими средними при проверке на открытие позиций – от 0 до 600, шаг – 200;
* разность при проверке на закрытие – 100.

Результаты тестов приведены в таблице ниже. Входные параметры и итоговая прибыль также указаны в ценовых пунктах.

**Табл. 3. Результаты тестирования торговой системы.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| № набора параметров | Тейк-профит | Стоп-лосс | Разность при открытии | Разность при закрытии | Количество открытых позиций | Количество убыточных позиций | Прибыль |
| 1 | 200 | 200 | 0 | 100 | 91 | 49 | -830 |
| 2 | 200 | 200 | 200 | 100 | 55 | 31 | -1400 |
| 3 | 200 | 200 | 400 | 100 | 30 | 15 | 0 |
| 4 | 200 | 200 | 600 | 100 | 18 | 13 | -1600 |
| 5 | 200 | 250 | 0 | 100 | 89 | 46 | -2230 |
| 6 | 200 | 250 | 200 | 100 | 55 | 28 | -1400 |
| 7 | 200 | 250 | 400 | 100 | 30 | 12 | 600 |
| 8 | 200 | 250 | 600 | 100 | 18 | 9 | -450 |
| 9 | 200 | 300 | 0 | 100 | 82 | 39 | -2390 |
| 10 | 200 | 300 | 200 | 100 | 55 | 26 | -1570 |
| 11 | 200 | 300 | 400 | 100 | 30 | 12 | 0 |
| 12 | 200 | 300 | 600 | 100 | 18 | 9 | -900 |
| 13 | 250 | 200 | 0 | 100 | 91 | 56 | -1880 |
| 14 | 250 | 200 | 200 | 100 | 55 | 31 | -200 |
| 15 | 250 | 200 | 400 | 100 | 30 | 17 | -150 |
| 16 | 250 | 200 | 600 | 100 | 18 | 13 | -1350 |
| 17 | 250 | 250 | 0 | 100 | 89 | 54 | -3690 |
| 18 | 250 | 250 | 200 | 100 | 54 | 28 | -300 |
| 19 | 250 | 250 | 400 | 100 | 30 | 14 | 500 |
| 20 | 250 | 250 | 600 | 100 | 18 | 9 | 0 |
| 21 | 250 | 300 | 0 | 100 | 81 | 46 | -3850 |
| 22 | 250 | 300 | 200 | 100 | 54 | 27 | -920 |
| 23 | 250 | 300 | 400 | 100 | 30 | 14 | -100 |
| 24 | 250 | 300 | 600 | 100 | 18 | 9 | -450 |
| 25 | 300 | 200 | 0 | 100 | 91 | 57 | -380 |
| 26 | 300 | 200 | 200 | 100 | 55 | 34 | 190 |
| 27 | 300 | 200 | 400 | 100 | 30 | 18 | 0 |
| 28 | 300 | 200 | 600 | 100 | 18 | 13 | -1100 |
| 29 | 300 | 250 | 0 | 100 | 89 | 55 | -2190 |
| 30 | 300 | 250 | 200 | 100 | 54 | 31 | 140 |
| 31 | 300 | 250 | 400 | 100 | 30 | 15 | 750 |
| 32 | 300 | 250 | 600 | 100 | 18 | 9 | 450 |
| 33 | 300 | 300 | 0 | 100 | 81 | 47 | -2350 |
| 34 | 300 | 300 | 200 | 100 | 54 | 30 | -350 |
| 35 | 300 | 300 | 400 | 100 | 30 | 15 | 100 |
| 36 | 300 | 300 | 600 | 100 | 18 | 9 | 0 |

Для большинства комбинаций входных параметров тестирование показало убыточность стратегии. Это можно объяснить малым таймфреймом графика, который, в совокупности с высокой ликвидностью инструмента, делает трудноотличимыми тренды в движении цены от краткосрочных импульсов.

В то же время, уровень разности между скользящими средними при проверке на открытие, равный 400, обеспечивает в большинстве случаев лучшие результаты, чем нулевой уровень, соответствующий сигналам при простом пересечении. Таким образом, можно говорить о том, что использование данной величины разности позволило бы уменьшить число убыточных сделок.

Изменение прибыли для наборов входных параметров {300; 300; 0; 100} и {300; 300; 400; 100} (33 и 36 запись табл. 3) представлено на графиках ниже. Для второго варианта параметров характерна меньшая просадка, а также примерно одинаковое время работы в плюс и в минус. С первым же набором входных данных торговая система не приносила прибыли практически на всем временном интервале.

**Рис. 13. Изменение прибыли при нулевой разности между скользящими средними.**

**рис. 14. Изменение разности при разности между скользящими средними в 400 пунктов.**

# Заключение

Разработанная механическая торговая система позволяет вести автоматизированную торговлю на фондовом и срочном рынках Московской биржи через систему брокерского обслуживания Quik. Возможности МТС позволяют проводить получение рыночных данных, их анализ и выставление торговых заявок с последующим управлением ими.

Тестирование торговой системы на исторических данных, однако, показало довольно низкую эффективность её применения на малых таймфреймах.

Использование двух дополнительных параметров во время проверки позволило получить лучшие результаты по сравнению с базовой торговой стратегией благодаря успешной фильтрации ложных сигналов. В то же время, принимая во внимание погрешность тестирующего алгоритма, можно утверждать, что вероятность работы МТС в убыток все равно остается высокой.

В целом, в рамках выпускной работы были решены все поставленные задачи. В первой части была рассмотрены применяемые для торговли информационные системы и способы прогнозирования цены. Далее была описана разработанная МТС. В заключительной части работы было произведено тестирование робота с различными вариантами входных параметров.

Варианты дальнейшей деятельности в области, связанной с тематикой и результатами ВКР, включают несколько потенциально интересных направлений:

* Продолжение исследования эффективности торговой стратегии с другими параметрами. К числу возможных изменений относятся, в частности, торгуемая ценная бумага и таймфрейм графика. Помимо этого, имеет смысл замена применяемого алгоритма анализа данных или добавление к нему дополнительных индикаторов с целью повышения точности анализа.
* Увеличение точности алгоритма тестирования. Использование минимально допустимого таймфрейма, в некоторой степени решает эту задачу. Однако при этом возникает проблема, заключающаяся в том, что Quik хранит соответствующие данные для меньшего временного периода. Обойти это ограничение возможно с помощью использования альтернативных источников данных.
* Повторная реализация торговой системы и алгоритма тестирования с помощью Lua. К преимуществам этого языка программирования по сравнению с QPILE можно отнести более высокую скорость работы и поддержку событийно-ориентированной модели программирования. Последнее упростит структуру кода и также положительно скажется на скорости выполнения алгоритмов.

# Библиографический список

1. ММВБ взялась за роботов. – Газета "Коммерсантъ", №129 (4429), 20.07.2010.
2. Сайт Московской биржи. – <http://rts.micex.ru/>
3. Варианты схем подключения к ПТК ASTS. – <http://fs.rts.micex.ru/f/123/select-soft-9.pdf>
4. Веб-сайт программного комплекса Quik. – <http://quik.ru>
5. Веб-сайт программного комплекса NetInvestor. – <http://netinvestor.ru>
6. Веб-сайт системы брокерского обслуживания TRANSAQ. – <http://www.transaq.ru/>
7. Веб-сайт торговой платформы MetaTrader 5. – <http://www.metatrader5.com/>
8. Торговая система NetInvestor Professional // Руководство пользователя NetInvestor Professional.
9. MetaTrader 5 // Веб сайт ОАО «Брокерский дом "ОТКРЫТИЕ"». – <http://www.open-broker.ru/ru/active-trading/trade-system/mt5/>
10. Заявки в ИТС «TRANSAQ» // Инвестиционная компания «Финам»: <http://finam.ru> – М.:2011. – <https://edox.finam.ru/info/faq/topic120.html>
11. Работа с ордерами: Stop Loss, Take Profit в MetaTrader // Брокерская компания «FinForce»: <http://www.finforce.ru/> – <http://www.finforce.ru/ru/trader/metatrader/documentations/114/>
12. Экспорт данных // Руководство пользователя NetInvestor Professional.
13. Веб-сайт платформы для торговых роботов StockSharp. – <http://stocksharp.com/>
14. Веб-сайт платформы для технического анализа Wealth-Lab. – <http://www.wealth-lab.com/>
15. Wealth-Lab // Робострой: <http://robostroy.ru/> – <http://robostroy.ru/faq/software/wealthlab.aspx>
16. А. А. Куликов. Форекс для начинающих. Справочник биржевого спекулянта. 2-е изд. – СПб.: Питер, 2006. – 384 с.:ил.
17. Moving Averages: Introduction // Investopedia: <http://investopedia.com/> – <http://www.investopedia.com/university/movingaverage/>
18. Индикатор Parabolic SAR, часть 1 // Журнал для трейдеров The Ignat Post: <http://theignatpost.ru/> – <http://theignatpost.ru/magazine/index.php?mlid=3366>
19. Bollinger Bands // StockCharts.com - ChartSchool – <http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:bollinger_bands>
20. Справочная система TRANSAQ. – <http://www.transaq.ru/cl_files/manual_transaq_v498_2010.pdf>
21. Moving Average Convergence-Divergence (MACD) // StockCharts.com - ChartSchool – <http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:moving_average_conve>
22. Редактор QPILE – Notepad++ // Механические торговые системы на QPILE: <http://selftrade.ru> – <http://www.selftrade.ru/automatization/redaktor-qpile-notepad/>
23. Алгоритмический язык QPILE // Руководство пользователя Quik версии 6.3.

# Приложение 1. Исходный код прототипа торговой системы

## Главный файл

PORTFOLIO\_EX EMA ED 06.02.13;

DESCRIPTION EMA ED;

CLIENTS\_LIST ALL\_CLIENTS;

FIRMS\_LIST ALL\_FIRMS;

INCLUDE lib.qpl;

PROGRAM

'Параметры программы

OpenFlag = 1

ClassCode = "SPBFUT" 'класс бумаги

SecCode = "EDH3" 'код бумаги

Account = "A901CIS" 'счет

ClientCode = "16228" 'код клиента

Limit = 6 'допустимое количество лотов

price\_tag = "ED\_PRICE"

lema\_tag = "LONG\_ED" 'тэг длинной EMA

sema\_tag = "SHORT\_ED" 'тэг короткой EMA

timeframe = 10 'таймфрейм

openDiff = 0.000 'lEMA + openDiff < sEMA => long

closeDiff = 0.000

'параметры заявок

tp\_offset = 0.0015

sl\_offset = 0.001

max\_offset = 0.0005

def\_offset = 0.0005

'настройки

GetNextTFLag = 20 'дополнительное время сна для защиты от рассинхронизации

'служебные переменные

new\_global("TransID", 1) 'номер сделки (для отправки заявок)

new\_global("StopOrderID", "0") 'номер последней сделки (для таблицы всех сделок)

new\_global("TradePrice", 0) 'цена совершения сделок

new\_global("SleepFlag", 0) 'флаг бездействия до конца фрейма

new\_global("EndSleepTime", "") 'время окончания сна

'Основная часть программы

Check = CheckTime()

BoughtCount = GetBoughtCount()

if Check == "T" and CheckVolume() <> 0 'если время торговать и в течение посл. периода были сделки

CheckEndSleep() 'проверка окончнания ожидания следующего фрейма (обнуление SleepFlag)

CheckStopOrder() 'проверка состояния стоп-заявки (выставление, обнуление StopOrderID)

if (BoughtCount == 0) and (OpenFlag == 1) 'если нет открытых позиций

if (CheckOpenSignal() == "B") and (SleepFlag == 0) 'если сигнал на покупку

Price = GetPrice("B") + def\_offset

SendOrder("B", Price, Limit - BoughtCount)

else

if (CheckOpenSignal() == "S") and (SleepFlag == 0) 'если сигнал на продажу

Price = GetPrice("S") - def\_offset

SendOrder("S", Price, Limit + BoughtCount)

end if

end if

else

if BoughtCount > 0 'если стоим в лонге

if (CheckCloseSignal() == "S") and (SleepFlag == 0) 'если сигнал продавать

Price = GetPrice("S") - def\_offset

SendOrder("S", Price, BoughtCount)

ClearStopOrder()

end if

else 'если стоим в шорте

if (CheckCloseSignal() == "B") and (SleepFlag == 0) 'если сигнал на покупку

Price = GetPrice("B") + def\_offset

SendOrder("B", Price, BoughtCount)

ClearStopOrder()

end if

end if

end if

else

if Check == "S" 'если время закрывать позицию (конец дня)

if StopOrderID <> "0"

ClearStopOrder()

end if

if BoughtCount > 0 'если открыт лонг

Price = GetPrice("S") - def\_offset

SendOrder("S", Price, BoughtCount)

else

if BoughtCount < 0 'если шорт

Price = GetPrice("B") + def\_offset

SendOrder("B", Price, -1\*BoughtCount)

end if

end if

end if

end if

'Конец основной части

'проверка времени; возвращает "Т" (trade) в период с 10:30 до 23:40,

'"S" (sell) в период с 23:40 по 23:45 и "N" (no trade) в остальное время

func CheckTime()

time = GetTime()

hour = substr(time, 0, 2)+0

min = substr(time, 2, 2)+0

if (hour == 10 and min >= 30) or (hour > 10 and hour < 18) or (hour == 19 and min >= 10) or (hour == 18 and min < 43) or (hour == 19 and min >= 20) or (hour > 19 and hour < 23) or (hour == 23 and min < 48)

result = "T"

else

if (hour == 18 and min >= 43) or (hour == 23 and min >= 48)

result = "S"

else

result = "N"

end if

end if

end func

'Проверка на наличие сделок за текущий период

func CheckVolume()

time\_curr = GetTime()

date = GetDate()

if (0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time\_curr),"LINES"),0),"VOLUME")) == 0

result = 0

else

result = 1

end if

end func

'возвращение количества купленных лотов инструмента

func GetBoughtCount()

result = 0

for i from 0 to get\_number\_of("FUTURES\_CLIENT\_HOLDINGS")

if get\_value (get\_item ("FUTURES\_CLIENT\_HOLDINGS", i), "SECCODE")== SecCode

result = get\_value(get\_item("FUTURES\_CLIENT\_HOLDINGS",i), "TOTAL\_NET")+0

end if

end for

end func

'Проверка сигналов на открытие; "B" - покупать, "S" - продавать "N" - отсутствие сигнала

func CheckOpenSignal()

date = GetDate()

price\_curr = 0

price\_prev = 0

lema\_curr = 0

lema\_prev = 0

sema\_curr = 0

sema\_prev = 0

time\_curr = GetTime()

'проверка сигнала без лага

price\_curr = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time\_curr),"LINES"),0),"CLOSE")

lema\_curr = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(lema\_tag, date, time\_curr),"LINES"),0),"CLOSE")

sema\_curr = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(sema\_tag, date, time\_curr),"LINES"),0),"CLOSE")

if price\_curr == 0 'выход из функции без сигнала, если 20 предыдущих фреймов пусты

result = "N"

return

end if

time\_prev = time\_curr 'определение начального значения time\_prev

for i from 1 to 20

if substr(time\_prev, 2, 2)+0 < timeframe

time\_prev = time\_prev - 10000 + (60-timeframe)\*100 + ""

else

time\_prev = time\_prev - timeframe\*100 + ""

end if

if (0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time\_prev),"LINES"),0),"VOLUME")) > 0 'если проверяемый фрейм непустой

price\_prev = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time\_curr),"LINES"),0),"CLOSE")

lema\_prev = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(lema\_tag, date, time\_prev),"LINES"),0),"CLOSE")

sema\_prev = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(sema\_tag, date, time\_prev),"LINES"),0),"CLOSE")

i = 20

end if

end for

if price\_prev == 0 'выход из функции, если среди 20 фреймов до time\_curr нет ни одного непустого

result = "N"

return

end if

if (lema\_curr + openDiff < sema\_curr) and (lema\_prev + openDiff >= sema\_prev)

result = "B"

else

if (lema\_curr - openDiff > sema\_curr) and (lema\_prev - openDiff <= sema\_prev)

result = "S"

else

result = "N"

end if

end if

end func

'Проверка сигналов на закрытие; "B" - покупать, "S" - продавать "N" - отсутствие сигнала

func CheckCloseSignal()

date = GetDate()

price\_curr = 0

price\_prev = 0

lema\_curr = 0

lema\_prev = 0

sema\_curr = 0

sema\_prev = 0

time\_curr = GetTime()

'проверка сигнала без лага

price\_curr = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time\_curr),"LINES"),0),"CLOSE")

lema\_curr = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(lema\_tag, date, time\_curr),"LINES"),0),"CLOSE")

sema\_curr = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(sema\_tag, date, time\_curr),"LINES"),0),"CLOSE")

if price\_curr == 0 'выход из функции без сигнала, если 20 предыдущих фреймов пусты

result = "N"

return

end if

time\_prev = time\_curr 'определение начального значения time\_prev

for i from 1 to 20

if substr(time\_prev, 2, 2)+0 < timeframe

time\_prev = time\_prev - 10000 + (60-timeframe)\*100 + ""

else

time\_prev = time\_prev - timeframe\*100 + ""

end if

if (0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time\_prev),"LINES"),0),"VOLUME")) > 0 'если проверяемый фрейм непустой

price\_prev = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time\_curr),"LINES"),0),"CLOSE")

lema\_prev = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(lema\_tag, date, time\_prev),"LINES"),0),"CLOSE")

sema\_prev = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(sema\_tag, date, time\_prev),"LINES"),0),"CLOSE")

i = 20

end if

end for

if price\_prev == 0 'выход из функции, если среди 20 фреймов до time\_curr нет ни одного непустого

result = "N"

return

end if

if (lema\_curr + closeDiff >= sema\_curr) and (lema\_prev + closeDiff < sema\_prev)

result = "S"

else

if (lema\_curr - closeDiff <= sema\_curr) and (lema\_prev - closeDiff > sema\_prev)

result = "B"

else

result = "N"

end if

end if

end func

'проверка на завершение ожидания следующего фрейма

func CheckEndSleep()

if (SleepFlag == 1) and (GetTime() > EndSleepTime)

SleepFlag = 0

EndSleepTime = ""

message("Sleep ended", 1) 'debug

end if

end func

'функция проверки состояния и перевыставления стоп-заявки

func CheckStopOrder()

'проверка наличия стопа

if GetBoughtCount() == Limit and StopOrderID == "0"

SendStopOrder("S")

return

else

if GetBoughtCount() == -1\*Limit and StopOrderID == "0"

SendStopOrder("B")

return

end if

end if

'проверка исполнения стопа

lastStop = GetLastStopOrder(SecCode)

if StopOrderID <> "0" and get\_value(lastStop, "NUMBER") <> "" and get\_value(lastStop, "STATUS") <> "ACTIVE"

StopOrderID = "0"

TransactionOutput()

end if

end func

'Отправка заявки

func SendOrder (Operation, Price, Quantity)

trans\_params = ""

trans\_params = set\_value (trans\_params, "TRANS\_ID", TransID)

trans\_params = set\_value (trans\_params, "ACTION", "NEW\_ORDER")

trans\_params = set\_value (trans\_params, "CLIENT\_CODE", ClientCode)

trans\_params = set\_value (trans\_params, "OPERATION", Operation)

trans\_params = set\_value (trans\_params, "TYPE", "L")

trans\_params = set\_value (trans\_params, "CLASSCODE", ClassCode)

trans\_params = set\_value (trans\_params, "SECCODE", SecCode)

trans\_params = set\_value (trans\_params, "PRICE", Price)

trans\_params = set\_value (trans\_params, "QUANTITY", Quantity)

trans\_params = set\_value (trans\_params, "ACCOUNT", Account)

trans\_params = set\_value (trans\_params, "EXECUTION\_CONDITION", "KILL\_BALANCE")

trans\_result = send\_transaction(5, trans\_params)

if get\_value (trans\_result, "RESULT\_EX")&"" == "3"

Pause(2000)

TransID = TransID + 1

for i from 1 to 20

lastTrade = GetLastTrade(SecCode)

if get\_value(lastTrade, "NUMBER")&"" <> ""

TransactionOutput()

i = 20

else

Pause(200)

end if

end for

end if

end func

'Создание стоп-заявки

func SendStopOrder(Operation)

Price = get\_value(get\_item("TRADES", get\_number\_of("TRADES")+0), "PRICE")+0

tp = 0 'цена активации тп

sl\_act = 0 'цена активации сл

sl\_exec = 0 'цена исполнения сл

if Operation == "B"

tp = Price - tp\_offset

sl\_act = Price + sl\_offset

sl\_exec = sl\_act + def\_offset

end if

if Operation == "S"

tp = Price + tp\_offset

sl\_act = Price - sl\_offset

sl\_exec = sl\_act - def\_offset

end if

if GetBoughtCount() == Limit or GetBoughtCount() == -1\* Limit

trans\_params = ""

trans\_params = set\_value(trans\_params, "ACCOUNT", Account)

trans\_params = set\_value(trans\_params, "CLASSCODE", ClassCode)

trans\_params = set\_value(trans\_params, "SECCODE", SecCode)

trans\_params = set\_value(trans\_params, "TRANS\_ID", TransID)

trans\_params = set\_value(trans\_params, "ACTION", "NEW\_STOP\_ORDER")

trans\_params = set\_value(trans\_params, "STOP\_ORDER\_KIND", "TAKE\_PROFIT\_AND\_STOP\_LIMIT\_ORDER")

trans\_params = set\_value(trans\_params, "OPERATION", Operation)

trans\_params = set\_value(trans\_params, "QUANTITY", Limit)

trans\_params = set\_value(trans\_params, "STOPPRICE", tp) 'цена активации ТП

trans\_params = set\_value(trans\_params, "OFFSET", max\_offset) 'отступ от максимума

trans\_params = set\_value(trans\_params, "OFFSET\_UNITS", "PRICE\_UNITS")

trans\_params = set\_value(trans\_params, "SPREAD", def\_offset) 'защитный спрэд

trans\_params = set\_value(trans\_params, "SPREAD\_UNITS", "PRICE\_UNITS")

trans\_params = set\_value(trans\_params, "STOPPRICE2", sl\_act) 'цена активации СЛ

trans\_params = set\_value(trans\_params, "PRICE", sl\_exec) 'цена исполнения СЛ

trans\_params = set\_value(trans\_params, "EXPIRY\_DATE", "GTC") 'до отмены

trans\_result = send\_transaction(5, trans\_params)

if get\_value(trans\_result, "RESULT\_EX")&"" == "3"

Pause(2000)

TransID = TransID + 1

for i from 1 to 20

lastStop = GetLastStopOrder(SecCode)

message("lastStop = "&get\_value(lastStop, "NUMBER")&"", 1) 'dbg

if get\_value(lastStop, "NUMBER")&"" <> ""

StopOrderID = get\_value(lastStop, "NUMBER")&""

SleepFlag = 1 'вырубаем бота до следующего фрейма

message("SleepFlag = 1", 1) 'dbg

EndSleepTime = GetNextFTWithLag(timeframe, GetNextTFLag)

message("EST = "&EndSleepTime, 1) 'dbg

i = 20

else

Pause(200)

end if

end for

end if

end if

end func

'Снятие активной стоп-заявки

func ClearStopOrder()

trans\_params = ""

trans\_params = set\_value(trans\_params, "TRANS\_ID", TransID)

trans\_params = set\_value(trans\_params, "ACTION", "KILL\_STOP\_ORDER")

trans\_params = set\_value(trans\_params, "STOP\_ORDER\_KEY", StopOrderID)

trans\_params = set\_value(trans\_params, "CLASSCODE", ClassCode)

trans\_result = send\_transaction(5, trans\_params)

if get\_value(trans\_result, "RESULT\_EX")&"" == "3"

Pause(2000)

TransID = TransID + 1

for i from 1 to 20

lastStop = GetLastStopOrder(SecCode)

if get\_value(lastStop, "STATUS") == "KILLED"

StopOrderID = "0"

i = 20

else

Pause(200)

end if

end for

end if

end func

'Вывод данных об операциях

func TransactionOutput()

lastTrade = GetLastTrade(SecCode)

Output = create\_map()

Output = set\_value(Output, "Time", get\_value(lastTrade, "TIME")&"")

Output = set\_value(Output, "Operation", get\_value(lastTrade, "OPERATION")&"")

Output = set\_value(Output, "Price", get\_value(lastTrade, "PRICE"))

Output = set\_value(Output, "Quantity", get\_value(lastTrade, "QUANTITY"))

add\_item(TransID, Output)

end func

'получение лучшей цены из стакана

func GetPrice(operation)

m = GET\_QUOTES\_II\_LEVEL\_DATA(ClassCode, SecCode)

bidCount = get\_value(m, "BID\_COUNT")

offerCount = get\_value(m, "OFFER\_COUNT")

bidColumn = get\_value(m, "BID")

offerColumn = get\_value(m, "OFFER")

q1 = get\_collection\_item (offerColumn, 0)

q2 = get\_collection\_item (bidColumn, bidCount-1)

if operation == "B"

price = 0+get\_value(q1, "PRICE")

end if

if operation == "S"

price = 0+get\_value(q2, "PRICE")

end if

result = price

end func

END\_PROGRAM

PARAMETER Time;

PARAMETER\_TITLE Время сделки;

PARAMETER\_DESCRIPTION Время выполнения сделки;

PARAMETER\_TYPE STRING (10);

END

PARAMETER Operation;

PARAMETER\_TITLE Тип операции;

PARAMETER\_DESCRIPTION Тип операции (покупка или продажа);

PARAMETER\_TYPE STRING (15);

END

PARAMETER Price;

PARAMETER\_TITLE Цена;

PARAMETER\_DESCRIPTION Цена совершения сделки;

PARAMETER\_TYPE NUMERIC (10,5);

END

PARAMETER Quantity;

PARAMETER\_TITLE Количество;

PARAMETER\_DESCRIPTION Количество лотов, участвовавших в сделке;

PARAMETER\_TYPE NUMERIC (10,5);

END

END\_PORTFOLIO\_EX

## Вспомогательный файл lib.qpl

'==========================================

'============== Дата и время ==============

'==========================================

'серверная дата

func GetDate()

server = get\_info\_param("TRADEDATE")

result = substr(server,6,4) & substr(server,3,2) & substr(server,0,2)

end func

'серверное время

func GetTime()

server = ""

server = get\_info\_param("SERVERTIME")

if len(server) > 0

result = substr(server,0,2) & substr(server,3,2) & substr(server,6,7)

else

result = get\_value(get\_datetime(),"HOUR")\*10000 + \_

get\_value(get\_datetime(),"MIN")\*100 + get\_value(get\_datetime(),"SEC") + ""

end if

end func

'функция паузы, на входе - время паузы в мс

func Pause(pause\_time)

pst = get\_datetime()

time = GetTime()

firstTime = 0 + substr(time,0,2)\*3600000 + substr(time,2,2)\*60000 + substr(time,4,2)\*1000 + get\_value(pst, "MILLISEC")

for pst\_flag from 0 to 1

pst = get\_datetime()

time = GetTime()

secondTime = 0 + substr(time,0,2)\*3600000 + substr(time,2,2)\*60000 + substr(time,4,2)\*1000 + get\_value(pst, "MILLISEC")

if secondTime - firstTime <= pause\_time

pst\_flag = pst\_flag - 1

end if

end for

end func

'ожидание следующего таймфрейма

func Sleep()

time = GetTime()

min = substr(time, 2, 2)

sec = substr(time, 4, 2)

endMin = (floor(min/timeframe) + 1) \* timeframe 'минута начала следующего фрейма

sleepTime = ((endMin - min) \* 60 - sec + 15) \* 1000 '15 - произвольное число для того, чтобы сократить шанс попасть на тот же фрейм

Pause(sleepTime)

end func

'получение времени следующего фрейма (тф <= 60)

func GetNextFrameTime(timeframe)

time = GetTime()

min = substr(time, 2, 2)

endMin = (floor(min/timeframe + 1) \* timeframe

if endMin < 10

result = substr(time, 0, 2)&"0"&endMin&"00"

else

if (endMin >= 10) and (endMin < 60)

result = substr(time, 0, 2)&endMin&"00"

else

endMin = endMin - 60

if endMin < 10 'делаем двузначный endMin

endMin = "0"&endMin

end if

result = (substr(time, 0, 2) + 1)&endMin&"00"

end if

end if

end func

'получение времени следующего фрейма с лагом (в секундах)

func GetNextFTWithLag(timeframe, lag)

a = GetNextFrameTime(timeframe) + lag

result = a&""

end func

'============================================

'============= Заявки и сделки ==============

'============================================

'получение последней стоп-заявки по инструменту

func GetLastStopOrder(SecCode)

n = get\_number\_of("STOP\_ORDERS") + 0

result = 0

for i from 1 to n

trade = get\_item("STOP\_ORDERS", i)

if get\_value(trade, "SECCODE") == SecCode

result = trade

end if

end for

end func

'получение последней сделки по инструменту

func GetLastTrade(SecCode)

n = get\_number\_of("TRADES") + 0

result = 0

for i from 1 to n

trade = get\_item("TRADES", i)

if get\_value(trade, "SECCODE") == SecCode

result = trade

end if

end for

end func

'============================================

'============= Математические ===============

'============================================

'остаток от деления

func mod (a1, a2)

if (0 + a2 = 0)

result = 0

else

cel = floor((a1/a2))

result = a1 - (a2 \* cel)

end if

end func

# Приложение 2. Исходный код программы-тестировщика

PORTFOLIO\_EX EMA RTS test w/ cycles;

DESCRIPTION EMA RTS test w/ cycles;

CLIENTS\_LIST ALL\_CLIENTS;

FIRMS\_LIST ALL\_FIRMS;

PROGRAM

new\_global("start\_time", 100000) 'время начала работы робота

new\_global("start\_date", 20130415)'начальная дата

new\_global("date", 0) 'текущая проверяемая дата

new\_global("time", 0) 'текущее проверяемое время

new\_global("dateFin", 20130515) 'последнее проверяемое число

new\_global("timeFin", 234000) 'время окончания проверки

new\_global("timeframe", 10) 'таймфрейм, <= 60 min

new\_global("price\_tag", "RTS\_PRICE") 'тэги

new\_global("lema\_tag", "RTS\_LONG")

new\_global("sema\_tag", "RTS\_SHORT")

new\_global("id", 0) 'id записи в таблицу

new\_global("limit", 1) 'лимит на покупку

new\_global("bought", 0) 'количество купленных лотов

new\_global("profit", 0) 'итоговая прибыль/убыток

new\_global("count", 0) 'число сделок

new\_global("fail\_count", 0) 'число убыточных сделок

tp\_start = 200 'диапазон и шаг изменения тейк-профита

tp\_step = 50

tp\_fin = 300

sl\_start = 200 'диапазон и шаг изменения стоп-лосса

sl\_step = 50

sl\_fin = 300

od\_start = 0 'диапазон и шаг изменения разности между скользящими средними

od\_step = 200 'при открытии

od\_fin = 600

cd\_start = 100 'диапазон и шаг изменения разности между скользящими средними

cd\_step = 50 'при закрытии

cd\_fin = 100

price = 0 'цена открытия позиции

'main

for tp\_offset from tp\_start to tp\_fin

for sl\_offset from sl\_start to sl\_fin

for openDiff from od\_start to od\_fin

for closeDiff from cd\_start to cd\_fin

date = start\_date 'обнуление переменных перед началом прогонки по истории

time = start\_time

profit = 0

count = 0

fail\_count = 0

for i from 0 to 1 'прогонка по истории

if date > dateFin or (date == dateFin and time >= timeFin + timeframe\*100)

Output(tp\_offset, sl\_offset, openDiff, closeDiff)

break

else

if time >= 235000 'смена дня

ChangeDate()

time = 095900

end if

ChangeTime()

bot\_Main()

i = -1

end if

end for

closeDiff = closeDiff - 1 + cd\_step

end for

openDiff = openDiff - 1 + od\_step

end for

sl\_offset = sl\_offset - 1 + sl\_step

end for

tp\_offset = tp\_offset - 1 + tp\_step

end for

'main end

'обновление даты

func ChangeDate()

time = 100000 'сброс времени на начало дня

month = substr(date + "", 4, 2)

day = substr(date + "", 6, 2)

'конец месяцев с 31 днем, кроме декабря

if (month == "01" or month == "03" or month == "05" or month == "07" or month == "10") and day == "31"

date = date + 100 - 30

else

'конец месяцев с 30 днями

if (month == "04" or month == "06" or month == "09" or month == "11") and day == "30"

date = date + 100 - 29

else

'конец декабря

if month == "12" and day == "31"

date = date + 10000 - 1100 - 30

else

'конец февраля

if month == "02"

year = substr(date + "", 0, 4) + 0

if mod(year, 4) == 0 and day == "29" 'високосный год

date = date + 100 - 28

else

if mod(year, 4) <> 0 and day == "28" 'невисокосный год

date = date + 100 - 27

end if

end if

else 'конец обычного дня

date = date + 1

end if

end if

end if

end if

'проверка на наличие торгов

if CheckTrade() == "NoTrade"

ChangeDate()

end if

end func

'функция смены времени

func ChangeTime()

minutes = substr(time + "", 2, 2) + 0

if minutes + timeframe >= 60 'конец часа

time = time + 10000 + (timeframe - 60)\*100

else

time = time + timeframe \* 100

end if

end func

'остаток от деления

func mod (a1, a2)

if (0 + a2 = 0)

result = 0

else

cel = floor((a1/a2))

result = a1 - (a2 \* cel)

end if

end func

'проверка день на наличие торгов; проверяется объем за первые 10 фреймов

func CheckTrade()

t = 100000

volume = 0

for i from 1 to 10

volume = volume + get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, t),"LINES"),0),"VOLUME")

if volume > 0

result = "Trade"

return

end if

if substr(t + "", 2, 2) + timeframe > 60 'конец часа

t = t + 10000 + (timeframe - 60)\*100

else

t = t + timeframe \* 100

end if

end for

result = "NoTrade"

end func

func Output(tp\_offset, sl\_offset, openDiff, closeDiff)

id = id + 1

Output = create\_map()

Output = set\_value(Output, "TP", tp\_offset)

Output = set\_value(Output, "SL", sl\_offset)

Output = set\_value(Output, "OD", openDiff)

Output = set\_value(Output, "СD", closeDiff)

Output = set\_value(Output, "Count", count)

Output = set\_value(Output, "FailCount", fail\_count)

Output = set\_value(Output, "Profit", profit)

add\_item(id, Output)

end func

'=========================== bot logic ========================================

'логика бота

func bot\_Main()

Check = bot\_CheckTime()

if Check == "T" and bot\_CheckVolume() <> 0

if bought == 0 'если нет позиций

if bot\_CheckOpenSignal() == "B" 'если сработал сигнал на покупку

price = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"OPEN") 'HIGH

bought = limit

profit = profit - bought \* price

'проверка текущего фрейма

p = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"LOW")

if p <= price - sl\_offset 'если сработал сл

price = price - sl\_offset

bought = 0

profit = profit + limit \* price

fail\_count = fail\_count + 1

count = count + 1

else

p = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"HIGH")

if p >= price + tp\_offset 'если сработал тп

price = price + tp\_offset

bought = 0

profit = profit + limit \* price

count = count + 1

end if

end if

else

if bot\_CheckOpenSignal() == "S" 'если сработал сигнал на продажу

price = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"OPEN") 'LOW

bought = -1 \* limit

profit = profit + limit \* price

'проверка текущего фрейма

p = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"HIGH")

if p >= price + sl\_offset 'если сработал сл

price = price + sl\_offset

bought = 0

profit = profit - limit \* price

fail\_count = fail\_count + 1

count = count + 1

else

p = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"LOW") 'проверка текущего фрейма

if p <= price - tp\_offset 'если сработал тп

price = price - tp\_offset

bought = 0

profit = profit - limit \* price

count = count + 1

end if

end if

end if

end if

else '============ конец случая без позиций

if bought > 0 'если стоим в лонге

price\_curr = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"LOW")

if price\_curr <= price - sl\_offset 'если сработал сл

price = price - sl\_offset

bought = 0

profit = profit + limit \* price

fail\_count = fail\_count + 1

count = count + 1

else

if bot\_CheckCloseSignal() == "S" 'если сигнал на закрытие позиции

price = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"OPEN")

bought = 0

profit = profit + limit \* price

fail\_count = fail\_count + 1

count = count + 1

'еще раз проверяем фрейм на случай, если можно открыть шорт

if bot\_CheckOpenSignal() == "S" 'если сработал сигнал на продажу

price = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"OPEN") 'LOW

bought = -1 \* limit

profit = profit + limit \* price

'проверка текущего фрейма

p = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"HIGH")

if p >= price + sl\_offset 'если сработал сл

price = price + sl\_offset

bought = 0

profit = profit - limit \* price

fail\_count = fail\_count + 1

count = count + 1

else

p = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"LOW") 'проверка текущего фрейма

if p <= price - tp\_offset 'если сработал тп

price = price - tp\_offset

bought = 0

profit = profit - limit \* price

count = count + 1

end if

end if

end if

else

price\_curr = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"HIGH")

if price\_curr >= price + tp\_offset 'если сработал тп

price = price + tp\_offset

bought = 0

profit = profit + limit \* price

count = count + 1

end if

end if

end if '================ конец случая с лонгом

else 'если стоим в шорте

price\_curr = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"HIGH")

if price\_curr >= price + sl\_offset 'если сработал сл

price = price + sl\_offset

bought = 0

profit = profit - limit \* price

fail\_count = fail\_count + 1

count = count + 1

else

if bot\_CheckCloseSignal() == "B" 'если сигнал на закрытие позиции

price = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"OPEN")

bought = 0

profit = profit - limit \* price

fail\_count = fail\_count + 1

count = count + 1

'еще раз проверяем фрейм на случай, если можно открыть шорт

if bot\_CheckOpenSignal() == "B" 'если сработал сигнал на продажу

price = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"OPEN") 'HIGH

bought = limit

profit = profit - limit \* price

'проверка текущего фрейма

p = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"LOW")

if p <= price + sl\_offset 'если сработал сл

price = price - sl\_offset

bought = 0

profit = profit + limit \* price

fail\_count = fail\_count + 1

count = count + 1

else

p = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"HIGH") 'проверка текущего фрейма

if p >= price + tp\_offset 'если сработал тп

price = price + tp\_offset

bought = 0

profit = profit + limit \* price

count = count + 1

end if

end if

end if

else

price\_curr = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"LOW")

if price\_curr <= price - tp\_offset 'если сработал тп

price = price - tp\_offset

bought = 0

profit = profit - limit \* price

count = count + 1

end if

end if

end if

end if

end if

else

if Check == "S" 'если сигнал закрывать позиции

if bought > 0 'если открыт лонг

price = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"OPEN") 'LOW

bought = 0

profit = profit + limit \* price

fail\_count = fail\_count + 1

count = count + 1

else

if bought < 0 'если открыт шорт

price = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"OPEN") 'HIGH

bought = 0

profit = profit - limit \* price

fail\_count = fail\_count + 1

count = count + 1

end if

end if

end if

end if

end func

'проверка времени [10:30; 23:40] => T, [23:40; 23:50] => S, [23:50; 10:30] => N

func bot\_CheckTime()

if time >= 103000 and time < 234000

result = "T"

else

if time >= 234000 and time < 235000

result = "S"

else

result = "N"

end if

end if

end func

'Проверка на наличие сделок за текущий период

func bot\_CheckVolume()

if (0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time),"LINES"),0),"VOLUME")) == 0

result = 0

else

result = 1

end if

end func

'Проверка сигналов на открытие позиций; "B" - покупать, "S" - продавать "N" - отсутствие сигнала

func bot\_CheckOpenSignal()

price\_curr = 0

price\_prev = 0

lema\_curr = 0

lema\_prev = 0

sema\_curr = 0

sema\_prev = 0

time\_curr = time

'один цикл "фор" для time\_curr = лаг в 1 фрейм

for i from 1 to 20 'определяем фрейм для time\_curr

if substr(time\_curr, 2, 2)+0 < timeframe 'определение времени таймфрейма

time\_curr = time\_curr - 10000 + (60-timeframe)\*100 + ""

else

time\_curr = time\_curr - timeframe\*100 + ""

end if

if (0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time\_curr),"LINES"),0),"VOLUME")) > 0 'если проверяемый фрейм непустой

price\_curr = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time\_curr),"LINES"),0),"CLOSE")

lema\_curr = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(lema\_tag, date, time\_curr),"LINES"),0),"CLOSE")

sema\_curr = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(sema\_tag, date, time\_curr),"LINES"),0),"CLOSE")

i = 20

end if

end for

if price\_curr == 0 'выход из функции без сигнала, если 20 предыдущих фреймов пусты

result = "N"

return

end if

time\_prev = time\_curr 'определение начального значения time\_prev

for i from 1 to 20

if substr(time\_prev, 2, 2)+0 < timeframe

time\_prev = time\_prev - 10000 + (60-timeframe)\*100 + ""

else

time\_prev = time\_prev - timeframe\*100 + ""

end if

if (0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time\_prev),"LINES"),0),"VOLUME")) > 0 'если проверяемый фрейм непустой

price\_prev = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time\_prev),"LINES"),0),"CLOSE")

lema\_prev = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(lema\_tag, date, time\_prev),"LINES"),0),"CLOSE")

sema\_prev = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(sema\_tag, date, time\_prev),"LINES"),0),"CLOSE")

i = 20

end if

end for

if price\_prev == 0 'выход из функции, если среди 20 фреймов до time\_curr нет ни одного непустого

result = "N"

return

end if

if (lema\_curr + openDiff < sema\_curr) and (lema\_prev + openDiff >= sema\_prev)

result = "B"

else

if (lema\_curr - openDiff > sema\_curr) and (lema\_prev - openDiff <= sema\_prev)

result = "S"

else

result = "N"

end if

end if

end func

'Проверка сигналов на закрытие позиций; "C" - закрывать, "N" - отсутствие сигнала

func bot\_CheckCloseSignal()

price\_curr = 0

price\_prev = 0

lema\_curr = 0

lema\_prev = 0

sema\_curr = 0

sema\_prev = 0

time\_curr = time

'один цикл "фор" для time\_curr = лаг в 1 фрейм

for i from 1 to 20 'определяем фрейм для time\_curr

if substr(time\_curr, 2, 2)+0 < timeframe 'определение времени таймфрейма

time\_curr = time\_curr - 10000 + (60-timeframe)\*100 + ""

else

time\_curr = time\_curr - timeframe\*100 + ""

end if

if (0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time\_curr),"LINES"),0),"VOLUME")) > 0 'если проверяемый фрейм непустой

price\_curr = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time\_curr),"LINES"),0),"CLOSE")

lema\_curr = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(lema\_tag, date, time\_curr),"LINES"),0),"CLOSE")

sema\_curr = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(sema\_tag, date, time\_curr),"LINES"),0),"CLOSE")

i = 20

end if

end for

if price\_curr == 0 'выход из функции без сигнала, если 20 предыдущих фреймов пусты

result = "N"

return

end if

time\_prev = time\_curr 'определение начального значения time\_prev

for i from 1 to 20

if substr(time\_prev, 2, 2)+0 < timeframe

time\_prev = time\_prev - 10000 + (60-timeframe)\*100 + ""

else

time\_prev = time\_prev - timeframe\*100 + ""

end if

if (0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time\_prev),"LINES"),0),"VOLUME")) > 0 'если проверяемый фрейм непустой

price\_prev = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(price\_tag, date, time\_prev),"LINES"),0),"CLOSE")

lema\_prev = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(lema\_tag, date, time\_prev),"LINES"),0),"CLOSE")

sema\_prev = 0+get\_value(get\_collection\_item(get\_value(get\_candle\_ex(sema\_tag, date, time\_prev),"LINES"),0),"CLOSE")

i = 20

end if

end for

if price\_prev == 0 'выход из функции, если среди 20 фреймов до time\_curr нет ни одного непустого

result = "N"

return

end if

if (lema\_curr + closeDiff >= sema\_curr) and (lema\_prev + closeDiff < sema\_prev)

result = "S"

else

if (lema\_curr - closeDiff <= sema\_curr) and (lema\_prev - closeDiff > sema\_prev)

result = "B"

else

result = "N"

end if

end if

end func

END\_PROGRAM

PARAMETER TP;

PARAMETER\_TITLE Тейк-профит;

PARAMETER\_DESCRIPTION Тейк-профит;

PARAMETER\_TYPE Numeric (12,5);

END

PARAMETER SL;

PARAMETER\_TITLE Стоп-лосс;

PARAMETER\_DESCRIPTION Стоп-лосс;

PARAMETER\_TYPE Numeric (12,5);

END

PARAMETER OD;

PARAMETER\_TITLE Разность при открытии;

PARAMETER\_DESCRIPTION Интервал при открытии;

PARAMETER\_TYPE Numeric (12,5);

END

PARAMETER СD;

PARAMETER\_TITLE Разность при закрытии;

PARAMETER\_DESCRIPTION Интервал при закрытии;

PARAMETER\_TYPE Numeric (12,5);

END

PARAMETER Count;

PARAMETER\_TITLE Количество открытых позиций;

PARAMETER\_DESCRIPTION Количество открытых позиций;

PARAMETER\_TYPE NUMERIC (12,0);

END

PARAMETER FailCount;

PARAMETER\_TITLE Количество неудачных открытий;

PARAMETER\_DESCRIPTION Количество неудачных открытий;

PARAMETER\_TYPE NUMERIC (12,0);

END

PARAMETER Profit;

PARAMETER\_TITLE Прибыль;

PARAMETER\_DESCRIPTION Прибыль;

PARAMETER\_TYPE NUMERIC (12,5);

END

END\_PORTFOLIO\_EX

## Пояснение к коду

В ходе выполнения ВКР было создано 2 версии тестировщика. Первая, представленная в приложении, производит перебор входных параметров и сбор статистики по работе МТС. Вторая выводит детальную историю операций и отличается только отсутствием перебора параметров и другими данными, получаемыми на выходе.