Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

Институт Вычислительной математики и информационных технологий

ОТЧЕТ

по эксплуатационной (производственной) практике

Обучающийся Фамилия Имя Отчество \_гр.09-132\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(ФИО студента) (Группа) (Подпись)

Научный руководитель: доцент кафедры САИТ,

канд. физ.-мат. наук Фамилия И.О.

(Подпись)

Руководитель практики от кафедры:

cт.преподаватель КСАИТ Фамилия И.О. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Подпись)

Оценка за практику \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Подпись)

Дата сдачи отчета \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Казань – 2025

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 3](#_Toc196041921)

[1. Развертывание системы 4](#_Toc196041922)

[2. Тестирование и эксплуатация системных модулей 6](#_Toc196041923)

[2.1. Модуль портфолио 6](#_Toc196041924)

[2.2. Модуль инструментов 8](#_Toc196041925)

[2.3. Модуль уведомлений 10](#_Toc196041926)

[2.4. Модуль состояния рынка 12](#_Toc196041927)

[2.5. Модуль настройки сигналов 13](#_Toc196041928)

[2.6. Модуль торгового робота 14](#_Toc196041929)

[2.7. Модуль дивидендов 16](#_Toc196041930)

[2.8. Модуль долгосрочных и среднесрочных сигналов 17](#_Toc196041931)

[2.9. Модуль базы знаний 18](#_Toc196041932)

[2.10. Модуль статистики 19](#_Toc196041933)

[ЗАКЛЮЧЕНИЕ 22](#_Toc196041934)

[СПИСОК ЛИТЕРАТУРЫ 24](#_Toc196041935)

[ПРИЛОЖЕНИЯ 25](#_Toc196041936)

[Приложение 1. Модуль удаления инструмента 25](#_Toc196041937)

[Приложение 2. Код построения графика для сигнала SMA 27](#_Toc196041938)

ВВЕДЕНИЕ

Производственная практика проходила на кафедре системного анализа и информационных технологий Института вычислительной математики и информационных технологий КФУ с 14 апреля 2025 года по 05 мая 2025 года.

В ходе предыдущей практики была разработана информационная система для управления инвестиционным портфелем с использованием алгоритмического трейдинга. Данная система представляет собой комплексное решение, включающее в себя Telegram-бота, backend-сервер на FastAPI, интеграцию с API брокера Т-Инвестиции, а также модульную архитектуру с базой данных для хранения пользовательских настроек, торговых сигналов и статистики операций. Основными возможностями системы являются отображение состояния портфеля, управление инструментами, настройка уведомлений, реализация торговых стратегий, получение статистических данных и графиков, а также доступ к справочной информации.

**Цель данной эксплуатационной практики** заключается в проверке работоспособности и эксплуатационной пригодности разработанной системы для управления инвестиционным портфелем. В рамках практики будет осуществлено развертывание системы, её эксплуатация и проверка функциональных модулей, а также сравнение полученных данных с официальными источниками. Дополнительно будет произведена настройка параметров и стратегий, моделирование пользовательских сценариев и наблюдение за поведением системы в условиях, приближенных к реальной эксплуатации.

Для достижения поставленной цели были определены следующие задачи:

1. развертывание системы в тестовой среде,
2. проведение ручного тестирования функциональных компонентов,
3. проверка работоспособности системы в условиях эксплуатации,
4. проверка поведения системы при различных входных параметрах,
5. проверка корректности работы команд и сравнение данных с внешними источниками.
6. Развертывание системы

Развертывание системы возможно двумя способами: путем непосредственного запуска программы или с использованием контейнеризации через Docker. Оба варианта позволяют обеспечить работоспособность системы, но контейнеризация предоставляет дополнительное удобство в управлении зависимостями и изоляции окружения.

Перед тем, как осуществлять запуск системы, необходимо прописать значения для токенов счетов Т-Инвестиций, идентификатор чата в Telegram, а также токен бота.

Запуск системы осуществляется через основной исполняемый файл, логирование действий осуществляется с использованием встроенного модуля logging [1]. В данном файле выполняются несколько ключевых операций:

* настройка окружения и подключение всех необходимых зависимостей и конфигурационных файлов, а также загрузка переменных окружения;
* инициализация бота и API-клиента;
* инициализация и запуск сервера FastAPI. Данный сервер выполняется в отдельном потоке, чтобы не блокировать работу основного процесса;
* настройка базы данных и планировщиков задач;
* определение команд и кнопок, с помощью которых пользователь будет взаимодействовать с ботом;
* запуск основного процесса, в котором бот начинает работу в режиме polling, непрерывно ожидая сообщения от пользователя.

Пример терминала после успешного запуска бота с настроенными планировщиками уведомлений и неактивной стратегией можно увидеть на рисунке 1.

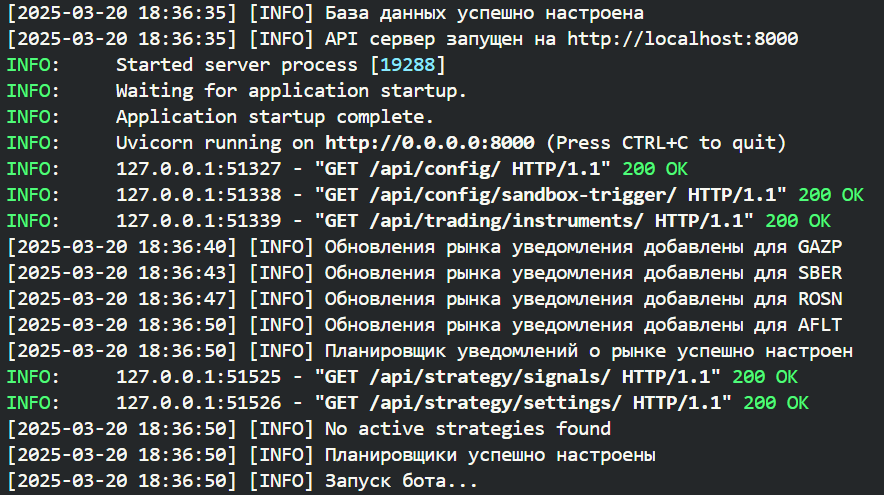


Рисунок - Состояние терминала после успешного запуска приложения

1. Тестирование и эксплуатация системных модулей

Для проверки работоспособности приложения и его эксплуатационной пригодности будет проводиться ручное тестирование каждого отдельного модуля с подробным описанием действий. Также, где это возможно, будет происходить сравнение непосредственно с официальным сайтом брокера и данными, которые на нем отображаются.

* 1. Модуль портфолио

Функция получения портфолио выполняется при нажатии на кнопку «Получить портфолио». После нажатия на данную кнопку система отправляет запрос брокеру, после чего полученные данные форматируются в удобный для чтения вид и предоставляются пользователю. Пример полученного ответа можно увидеть на рисунке 2 и на рисунке 3.

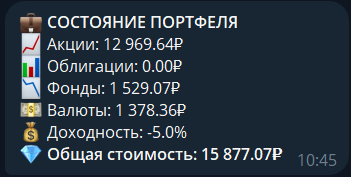


Рисунок - Пример ответа о состоянии портфеля пользователя

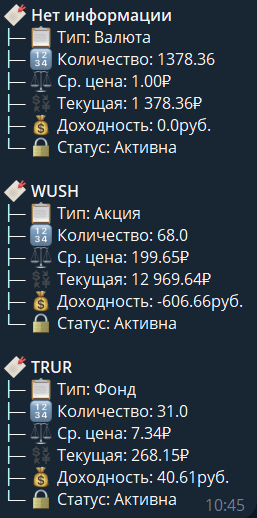


Рисунок - Пример ответа о позициях в портфеле пользователя

Из полученных ответов пользователь может извлечь главную информацию о состоянии своего портфеля и активах, находящихся в нем.

Для портфеля пользователь может узнать информацию о количестве различных валютных инструментов, выраженную в рублях, а также доходность и общую стоимость активов.

Для каждого инструмента пользователь получает более детальную информацию, которая раскрывает следующие параметры инструмента:

* тип инструмента,
* количество в портфеле,
* средняя цена,
* цена на момент отправки отчета,
* доходность на момент отправки отчета,
* статус инструмента.

Все приходит в удобном для пользователя виде, с добавлением эмодзи для лучшей концентрации на отдельных пунктах. Можно заметить, что у некоторых инструментов отсутствует название и оно заменено на «Нет информации». Это происходит в виду того, что у некоторых биржевых инструментов, таких как валюты и фонды, может отсутствовать поле name, что учитывается приложением.

Для сравнения рассмотрим скриншот из приложения брокера, который представлен на рисунке 4.

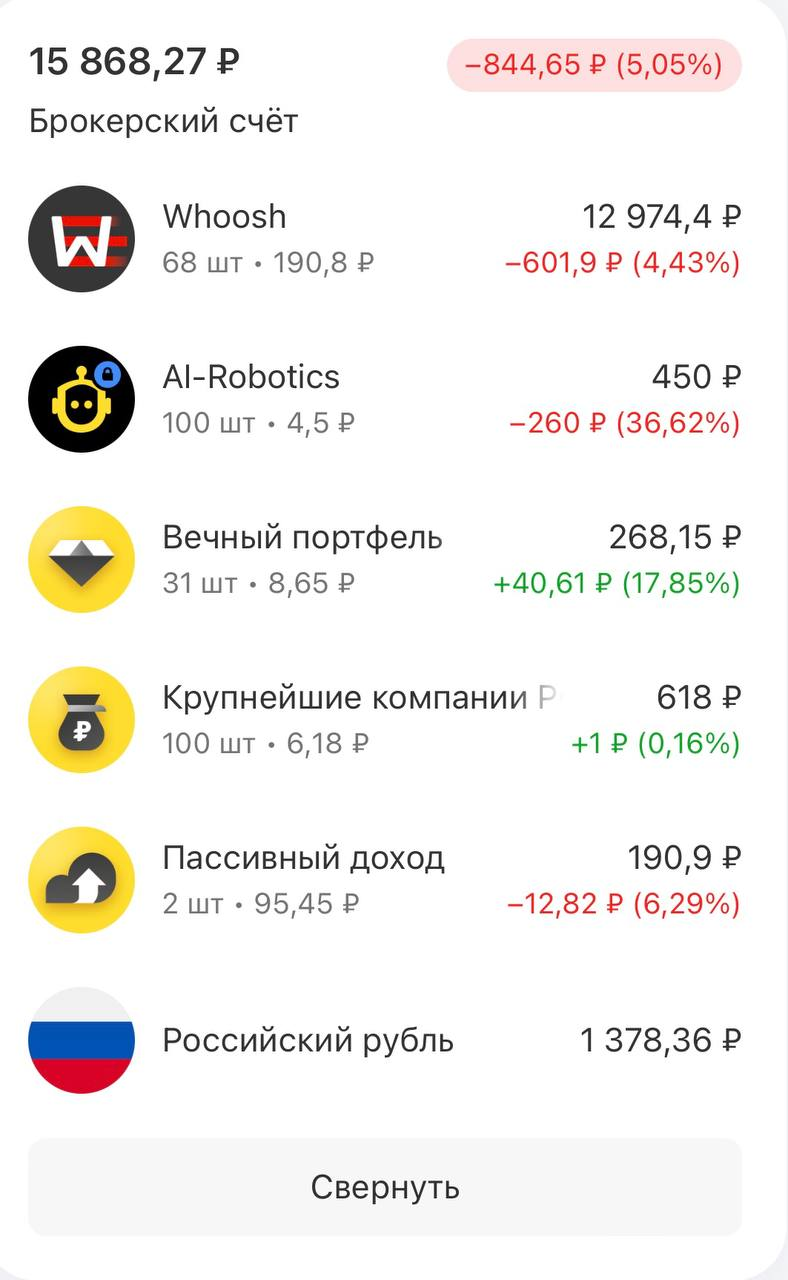


Рисунок - Скриншот из приложения брокера

Как можно заметить, данные в отчете и на скриншоте различаются с небольшой погрешностью. Это происходит в связи с постоянным обновлением данных в приложении брокера, в то время как данные в отчете актуальны на момент отправки отчета.

* 1. Модуль инструментов

Модуль инструментов включает в себя функциональности добавления, отображения и удаления. Визуальное представление управления инструментами пользователя можно увидеть на рисунке 5.

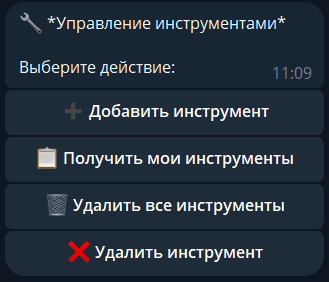


Рисунок - Управление инструментами пользователя

Необходимо опробовать и протестировать каждую функциональность по-отдельности.

При нажатии на кнопку «Добавить инструмент» система предложит пользователю ввести тикер инструмента, а если данный инструмент уже существует в базе данных, пользователю придет соответствующее уведомление, как показано на рисунке 6.



Рисунок - Уведомление о существовании инструмента

Случай с некорректно введенным тикером также обрабатывается, это можно увидеть на рисунке 7.

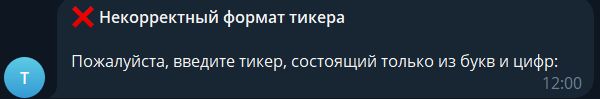


Рисунок - Обработка некорректно введенного тикера

При вводе корректного тикера, если он отсутствует в базе данных, инструмент успешно добавится в систему и пользователь получит соответствующее уведомление.

При нажатии на кнопку «Получить мои инструменты» пользователю придет список его инструментов, который будет включать тикер каждого инструмента и его специальный идентификатор. Данный список можно увидеть на рисунке 8. При отсутствии у пользователя добавленных инструментов, ему придет соответствующее уведомление.



Рисунок - Список инструментов пользователя

При нажатии на кнопку «Удалить инструмент» пользователь получит список его инструментов в виде активных кнопок. Данный список представлен на рисунке 9.



Рисунок - Список инструментов для удаления

После выбора соответствующего инструмента он удалится из базы данных и пользователь получит соответствующее уведомление. Программный код модуля для удаления инструмента представлен в [приложении 1](#Приложение_1).

При нажатии на кнопку «Удалить все инструменты» пользователю придет уведомление об удалении всех его добавленных инструментов из базы данных, что можно увидеть на рисунке 10.

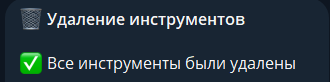


Рисунок - Уведомление об удалении всех инструментов

* 1. Модуль уведомлений

Модуль уведомлений представлен в виде четырех функциональных команд, которые можно увидеть на рисунке 11.

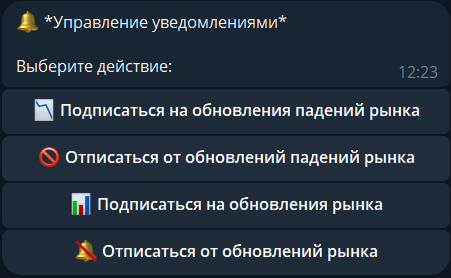


Рисунок - Модуль управления уведомлениями

При выборе команды «Подписаться на обновления падений рынка» или команды «Подписаться на обновления рынка» система предложит пользователю выбрать интервал для получения уведомлений, как показано на рисунке 12.

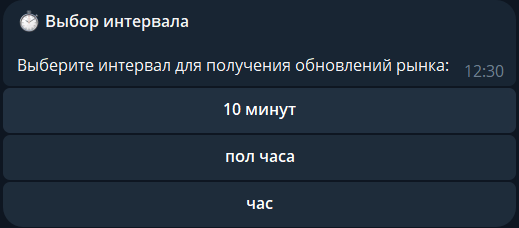


Рисунок - Выбор интервала уведомлений

После выбора интервала уведомлений система настраивает планировщики для каждого инструмента и присылает уведомление об успешной подписке, что можно увидеть на рисунке 13 и на рисунке 14.

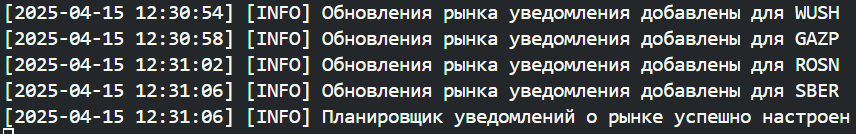


Рисунок - Логирование планировщиков уведомлений



Рисунок - Успешная подписка на уведомления

По прошествии указанного срока пользователь получит уведомления по каждой бумаге. Пример уведомления представлен на рисунке 15.

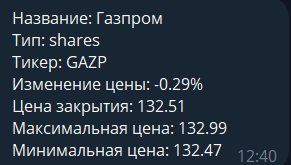


Рисунок - Пример уведомления по изменению цены инструмента

Для каждого инструмента система специальным запросом получает ценовые свечи за выбранный интервал, после чего определяет изменение цены и предоставляет пользователю в удобном для восприятия виде.

Если сравнивать информацию из уведомления с терминалом, который представлен на рисунке 16, можно увидеть тенденцию на падение ценной бумаги. Если говорить о процентном изменении цены, присутствует небольшая погрешность. Это происходит, так как в терминале данные обновляются каждые несколько секунд, а данные из уведомления актуальны на момент получения.

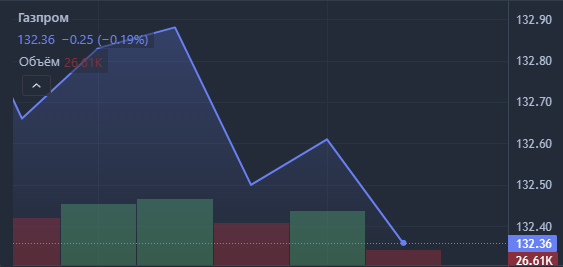


Рисунок - Состояние терминала по ценной бумаге

При выборе пользователем команды «Отписаться от уведомлений» он будет отписан от соответствующих уведомлений и получит об этом предупреждение.

* 1. Модуль состояния рынка

Данный модуль предлагает пользователю три команды, которые представлены на рисунке 17.

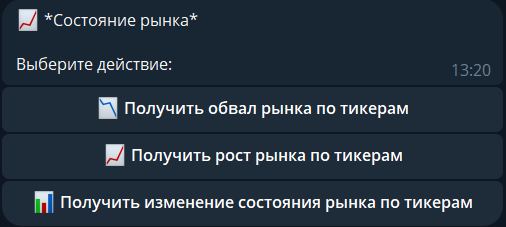


Рисунок - Доступные пользователю команды

При выборе любой команды в данном модуле будет действовать примерно тот же механизм, что и у уведомлений. Для выбранного пользователем временного интервала система будет получать ценовые свечи и вычислять изменение цены бумаги, после чего, в зависимости от выбранного условия изменения цены, присылать соответствующие уведомления, что можно увидеть на рисунке 18.

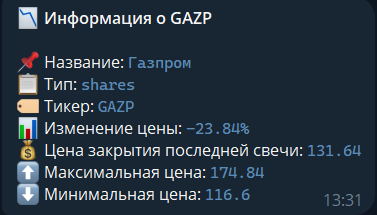


Рисунок - Уведомление по ценной бумаге за выбранный период

Стоит уточнить, что корректно сравнить изменение цен в уведомлении и в терминале брокера не получится, так как система высчитывает изменение цены в интервале от месяца назад до текущего дня, в то время как в терминале информация предоставляется в интервале от начала прошлого месяца до начала текущего месяца.

* 1. Модуль настройки сигналов

Модуль настройки сигналов предоставляет пользователю настроить любой сигнал, который предусмотрен системой. Для любого сигнала пользователь может по своему желанию указать абсолютно любые значения для каждого параметра. При нажатии на кнопку «Настройка сигналов» пользователь получит список сигналов в виде функциональных кнопок, которые представлены на рисунке 19.

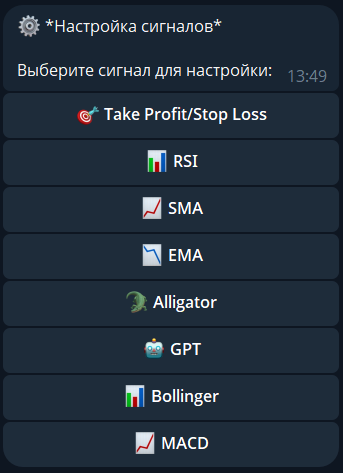


Рисунок - Список сигналов для настройки

В качестве примера будет рассмотрена настройка сигнала SMA [2]. При выборе соответствующего сигнала пользователю будет предложено ввести значение первого параметра, а после и всех остальных параметров.

После ввода всех параметров, система отправит пользователю соответствующее уведомление и обновит соответствующие параметры сигнала в базе данных, что можно увидеть на рисунке 20.

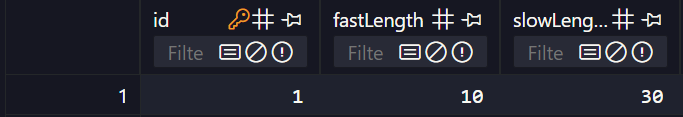


Рисунок - Параметры сигнала SMA в базе данных

* 1. Модуль торгового робота

Модуль торгового робота предоставляет пользователю функционал для торговли на бирже. Доступный пользователю набор команд данного модуля представлен на рисунке 21.

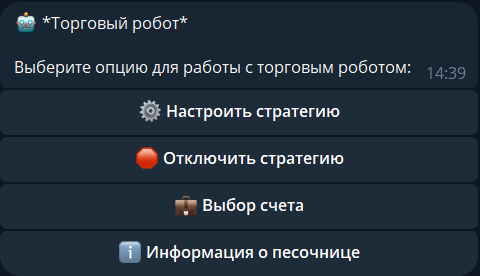


Рисунок - Команды модуля торгового робота

При выборе команды «Настроить стратегию» пользователь перейдет в режим настройки стратегии. Первым делом пользователю придет список функциональных кнопок с выбором соответствующих сигналов, который представлен на рисунке 22. При выборе конкретного сигнала он добавляется во временный список хранения сигналов. Если же был выбран сигнал, который еще не настроен, пользователю отправится соответствующее предупреждение.

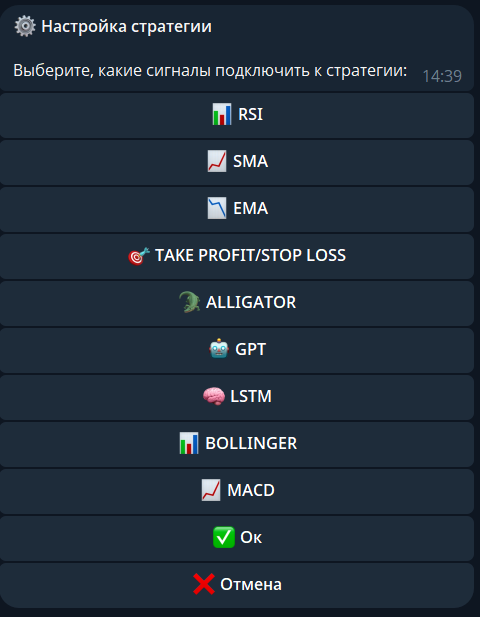


Рисунок - Выбор сигналов для стратегии

После выбора всех нужных сигналов пользователю будет предложено выбрать интервал срабатывания стратегии.

Далее пользователь выбирает режим торговли и количество бумаг, которые торговый робот будет продавать или покупать за один раз, а также логический оператор срабатывания стратегии, который означает, что стратегия может сработать либо при уведомлениях на покупку или продажу одновременно всех сигналов, либо же любого сигнала из выбранных.

После выбора оператора срабатывания стратегия обновляется и запускается, срабатывая через выбранный временной интервал.

При успешном срабатывании стратегии в терминале отобразится соответствующая запись, а пользователю отправится соответствующее уведомление, пример которого можно увидеть на рисунке 23.

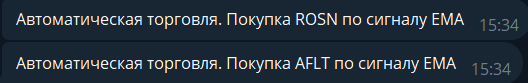


Рисунок - Уведомления об успешной покупке ценных бумаг

Чтобы убедиться в корректности операции, можно зайти в приложение брокера и посмотреть последние выполненные операции, которые представлены на рисунке 24.

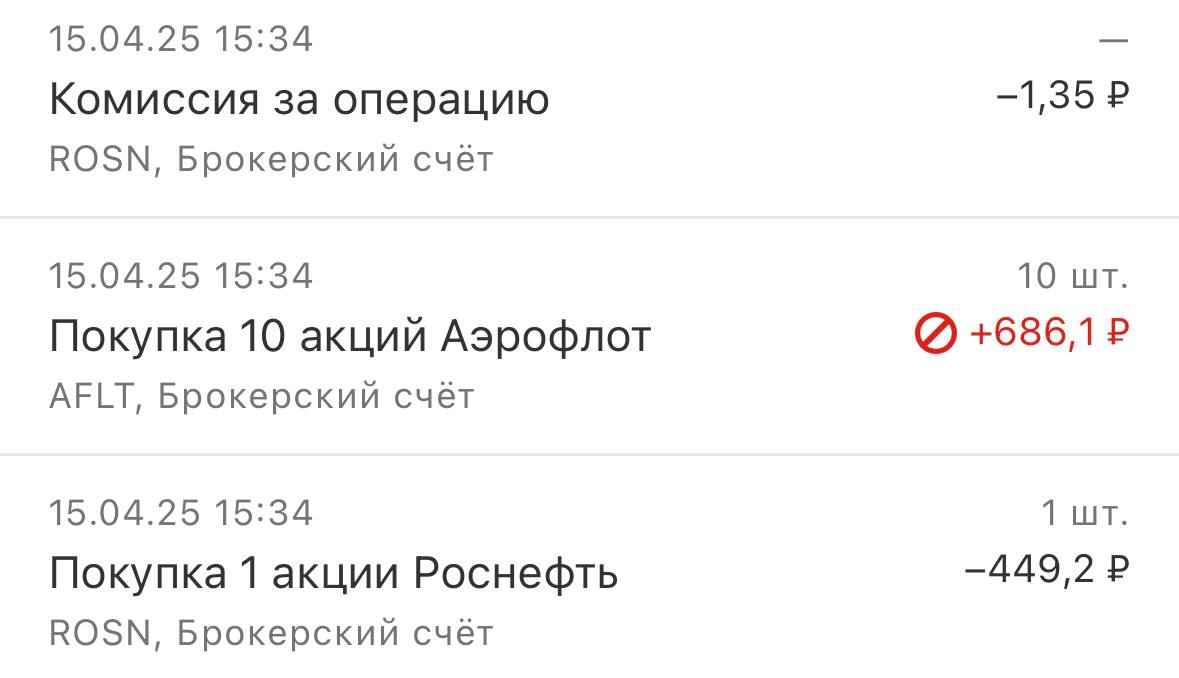


Рисунок - Выполненные операции по покупке ценных бумаг

При выборе команды «Отключить стратегию» все настройки и планировщик стратегии сбрасываются, что можно увидеть на рисунке 25.

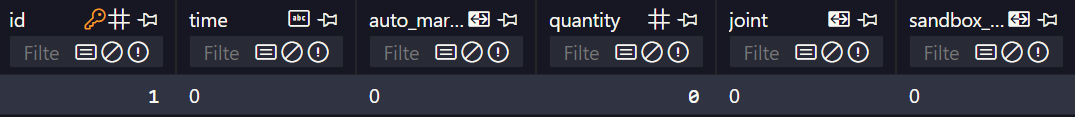


Рисунок - Сброшенные настройки стратегии

При выборе команды «Выбор счета» пользователю будет дана возможность выбора из боевого счета и песочницы. При выборе песочницы стратегия пользователя останется такой же, но торги будут вестись уже на отдельном счете с ненастоящей валютой. Если же пользователь выберет боевой счет, торговля будет осуществляться на его основном счете.

При выборе команды «Информация о песочнице» пользователь получит окно взаимодействия с песочницей, которое представлено на рисунке 26.

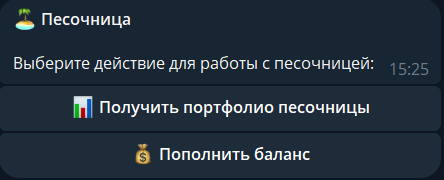


Рисунок - Окно взаимодействия с песочницей

При выборе команды «Пополнить баланс» пользователь сможет ввести значение в рублях, после чего баланс песочницы будет увеличен на это самое значение. Если же пользователь выберет команду «Получить портфолио песочницы», то получит полный отчет по активам в песочнице, идентичный отчету по основному портфелю.

* 1. Модуль дивидендов

Данный модуль представляет из себя функциональность одной команды – получения информации о предстоящих дивидендах по выбранным пользователем инструментам. При выборе команды «Дивиденды» пользователю будет предложено выбрать период, за который он хочет посмотреть наличие дивидендов по ценным бумагам.

Период в данном случае валидируется и не может быть меньше 1 и больше 365 дней. После ввода периода пользователю придет уведомление о дивидендах по его ценным бумагам за данный период. Пример уведомления можно увидеть на рисунке 27.

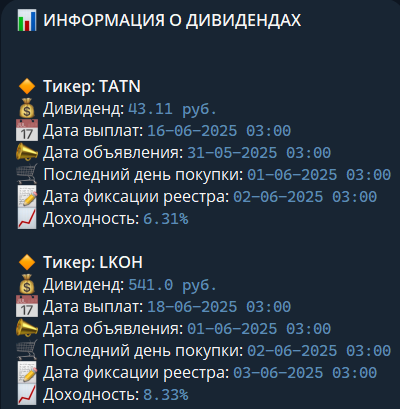


Рисунок - Уведомление о дивидендах

* 1. Модуль долгосрочных и среднесрочных сигналов

Модуль долгосрочных и среднесрочных сигналов предлагает пользователю визуальное представление по конкретным сигналам. При выборе команды по долгосрочным и среднесрочным сигналам пользователю присылается список функциональных кнопок с перечислением сигналов, представленный на рисунке 28.

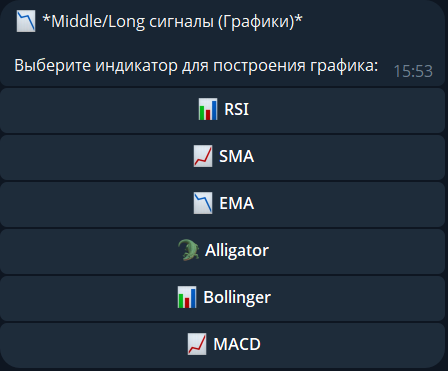


Рисунок - Список доступных сигналов

Если какой-то сигнал не был настроен, то при его выборе пользователю отправится соответствующее уведомление.

В качестве примера будет рассмотрен сигнал SMA. После выбора соответствующего сигнала пользователю будет предложен выбор из временных периодов, после чего пользователю будет представлен список функциональных кнопок, состоящий из перечисления его ценных бумаг.

После выбора пользователем нужной ему ценной бумаги, будет построен график по выбранному сигналу, отражающий временной промежуток и все параметры сигнала, а также ценовые свечи инструмента. Пример такого графика представлен на рисунке 29. Программный код построения графика представлен в [приложении 2](#Приложение_2).

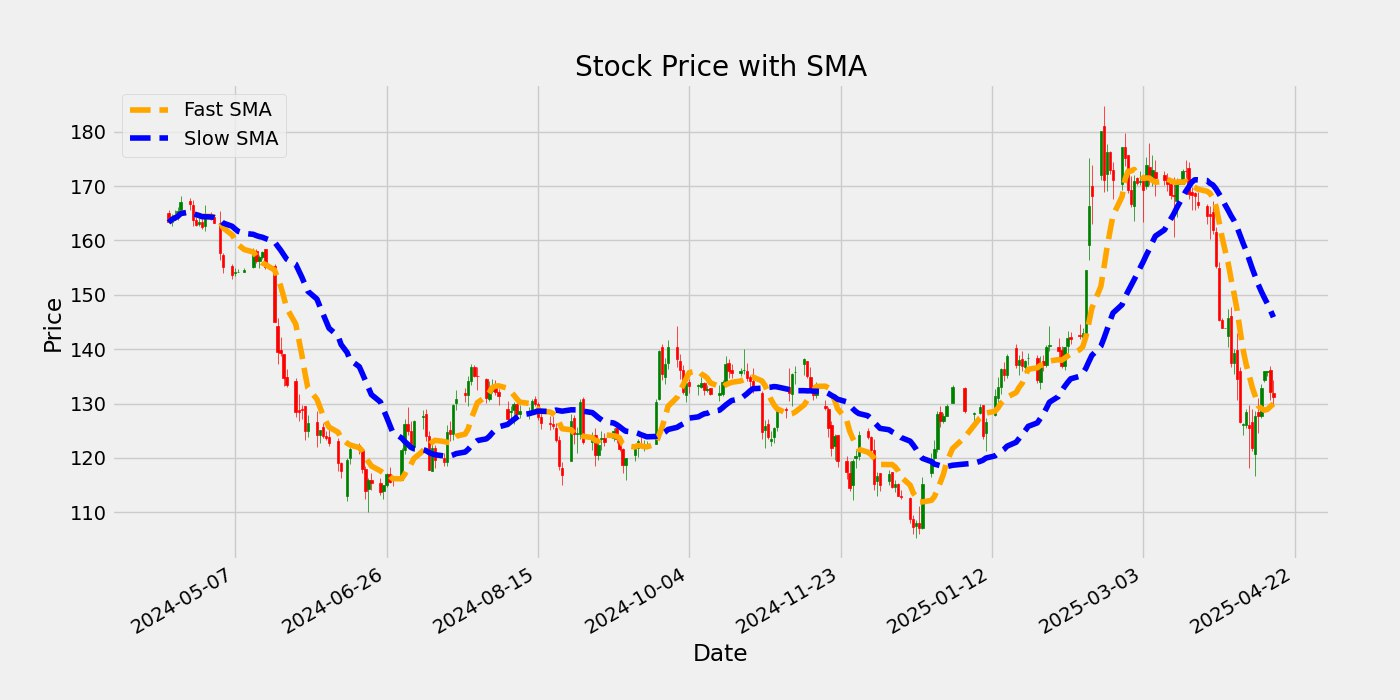


Рисунок - График сигнала SMA

* 1. Модуль базы знаний

Модуль базы знаний предоставляет пользователю набор информации по всем модулям системы. При выборе команды «База знаний» пользователь получает уведомление с функциональными кнопками, которое представлено на рисунке 30.

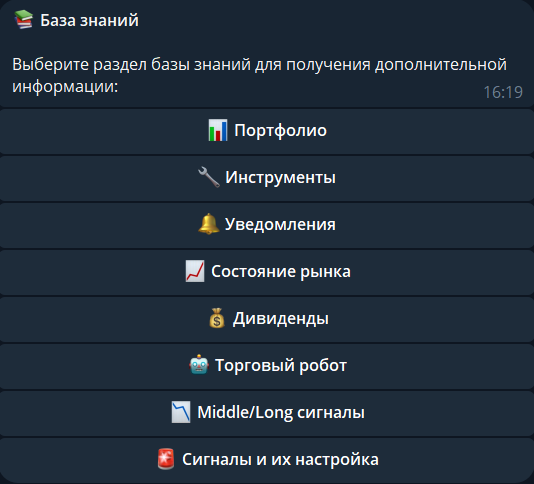


Рисунок - База знаний

При выборе необходимой пользователю команды он получит информацию по интересующему его модулю. Для примера будет рассмотрена команда «Инструменты». При выборе данной команды пользователь получит уведомление, представленное на рисунке 31.

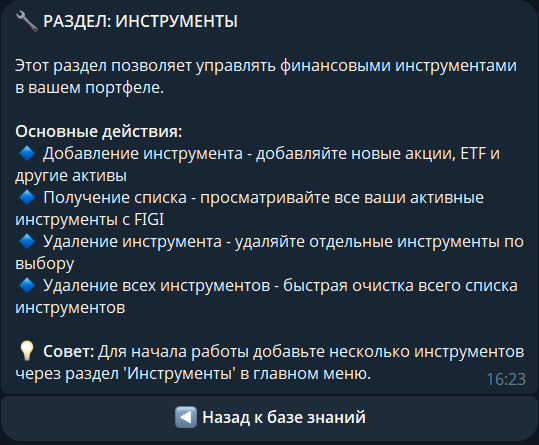


Рисунок - Раздел инструментов

В данном разделе пользователь может найти общую информацию по соответствующему модулю, обзор выполняемых модулем команд, а также некоторые советы по работе данного модуля внутри системы.

* 1. Модуль статистики

Данный модулю предоставляет пользователю статистику покупок и продаж, совершенных через торгового робота. При выборе команды «Статистика» пользователю будет предложено либо выбрать интервал получения статистики, либо же получить общую статистику за все время.

После выбора получения интервала статистики пользователю будет отправлено наглядное представление покупок и продаж в виде различных графиков, которые будут представлены в ответе в различных видах. Для построения графиков используется библиотека matplotlib [3].

На первом типе графиков отражена статистика покупок или продаж по тикерам. Пример такого графика можно увидеть на рисунке 32.

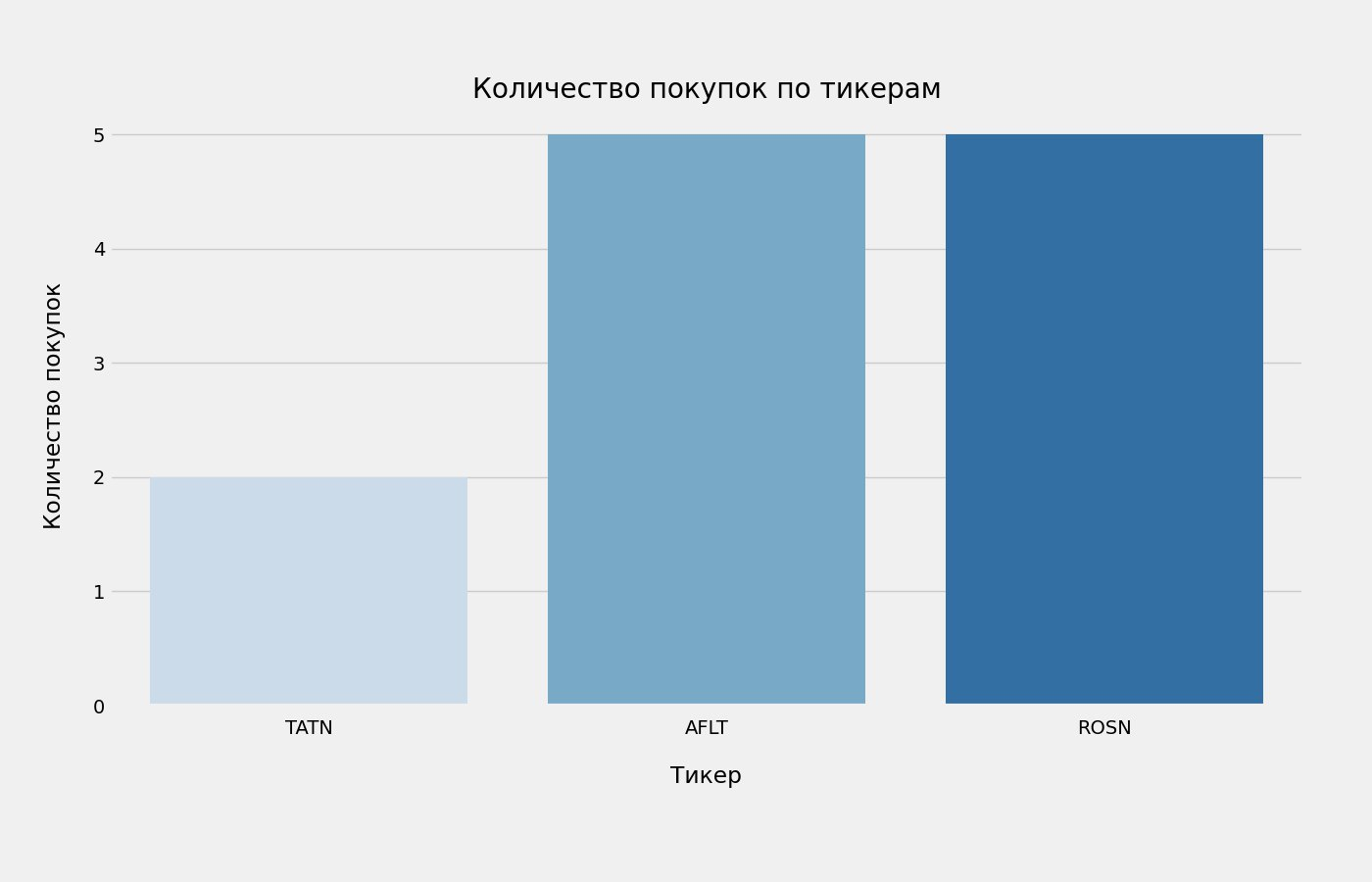


Рисунок - График количества покупок по тикерам

Второй тип графиков отражает количество покупок или продаж по сигналам. Пример такого графика можно увидеть на рисунке 33.

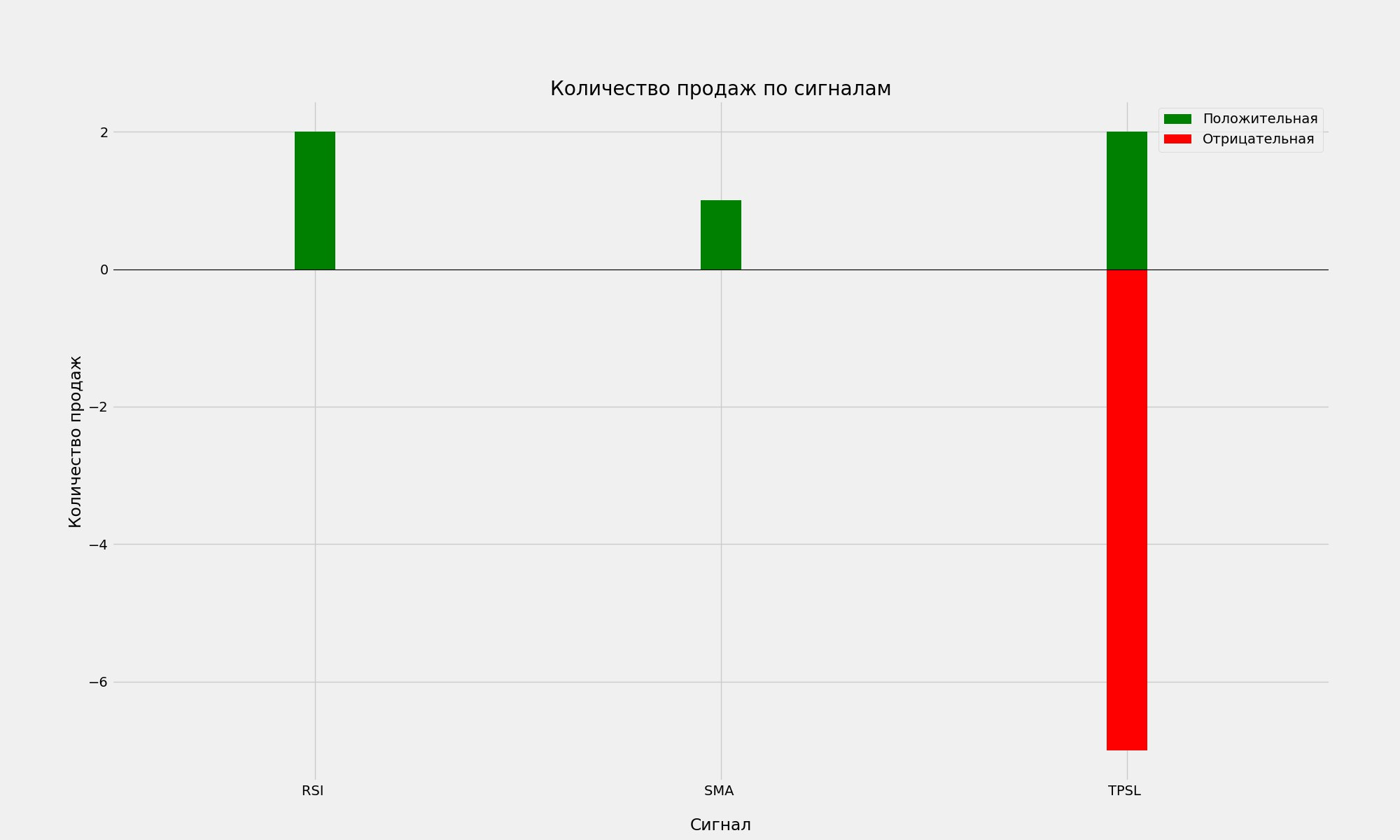


Рисунок - График количества продаж по сигналам

Следующий тип графиков отражает количество покупок или продаж по дням. Пример такого графика представлен на рисунке 34.

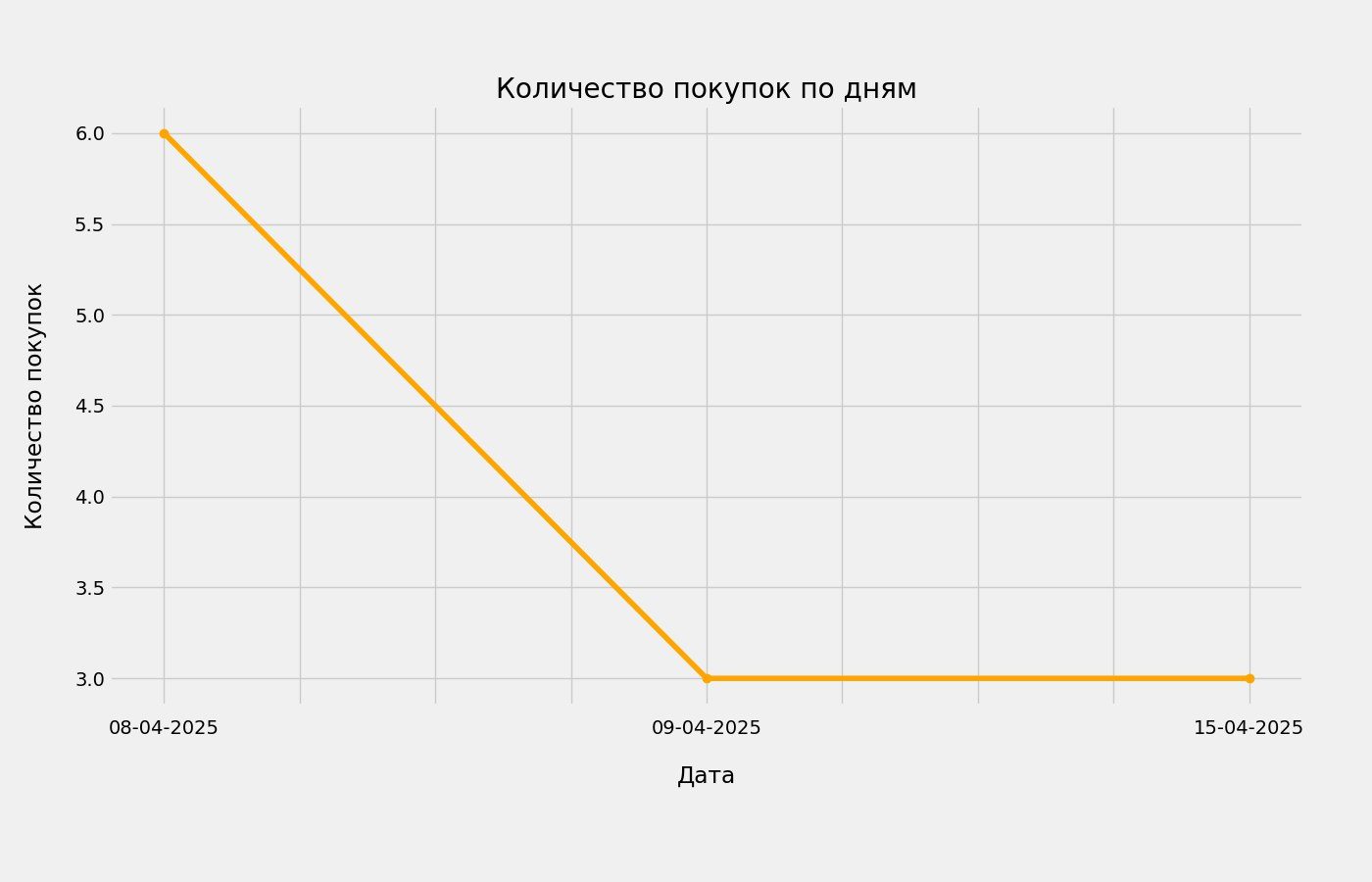


Рисунок - График количества покупок по дням

Последний график – это график соотношения положительной и отрицательной маржи. Пример такого графика представлен на рисунке 35.



Рисунок - График соотношения маржи

ЗАКЛЮЧЕНИЕ

В ходе эксплуатационной практики была выполнена проверка работоспособности информационной системы для автоматизированного управления инвестициями. Практика включала в себя развертывание приложения, настройку конфигурационных параметров, а также пошаговое ручное тестирование всех ключевых компонентов системы [4]. Проект имеет модульную архитектуру, что упростило тестирование и сопровождение, а сама структура кода организована в соответствии с рекомендациями по построению архитектуры Python-приложений [5]. Проверка осуществлялась вручную по каждому функциональному модулю. Для каждого из них были смоделированы сценарии взаимодействия с пользователем, а также зафиксированы полученные данные и реакция системы.

В рамках практики была подтверждена стабильная работа программного обеспечения при типовых пользовательских действиях. Система корректно взаимодействует с API брокера, обрабатывает команды, предоставляет актуальные данные и демонстрирует правильную работу стратегий в условиях, приближенных к реальным. В некоторых модулях внимание уделялось сопоставлению данных с официальным приложением брокера для оценки точности получаемой информации. Практика позволила закрепить навыки тестирования, настройки окружения, анализа поведения системы, а также выработать рекомендации по возможному улучшению взаимодействия с пользователем.

За период эксплуатационной практики были приобретены следующие компетенции (таблица 1):

Таблица - Приобретенные компетенции

|  |  |  |
| --- | --- | --- |
| Компетенция | Расшифровка компетенции | Описание приобретенных знаний, умений и навыков |
| УК-1 | Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач | Сравнивал данные, полученные через систему, с данными официального брокера. Оценивал корректность работы функций и отклонения в расчетах |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| Компетенция | Расшифровка компетенции | Описание приобретенных знаний, умений и навыков |
| ПК-1 | Проверка работоспособности и рефакторинг кода программного обеспечения | Проводил ручное тестирование всех модулей системы, выявлял ошибки отображения, некорректные состояния и обрабатывал пограничные случаи |
| ПК-2 | Интеграция программных модулей и компонент и верификация выпусков программного продукта | Проверял согласованную работу всех компонентов системы |
| ПК-3 | Разработка требований и проектирование программного обеспечения | На основе результатов тестирования сформулировал предложения по улучшению логики работы команд и взаимодействия с пользователем |
| ПК-4 | Оценка и выбор варианта архитектуры программного средства | Запускал систему через основной исполняемый файл, анализировал взаимодействие компонентов, оценивал удобство сопровождения и тестирования выбранной архитектурной схемы |

Практика позволила протестировать созданную ранее систему и получить практический опыт эксплуатации программного продукта в условиях, приближенных к реальному применению. Работа с функциональными модулями, ручное тестирование и наблюдение за логикой обработки пользовательских сценариев углубили понимание внутреннего устройства системы, её архитектуры и взаимосвязей между компонентами. Результаты практики подтверждают, что система способна выполнять заявленные функции и может быть использована для автоматизации инвестиционной деятельности.

СПИСОК ЛИТЕРАТУРЫ

1. Python Logging Module Documentation [Электронный ресурс]. – 2025. — URL: <https://docs.python.org/3/library/logging.html> (дата обращения 14.04.2025).
2. Investopedia. Technical Indicators: SMA, RSI, MACD [Электронный ресурс]. – 2023. — URL: https://www.sqlalchemy.org/ (дата обращения 20.04.2025).
3. Matplotlib documentation [Электронный ресурс]. – 2024. — URL: <https://matplotlib.org/stable/index.html> (дата обращения 22.04.2025).
4. Руководство по ручному тестированию программного обеспечения [Электронный ресурс]. – 2024. — URL: https://sky.pro/wiki/profession/osnovy-ruchnogo-testirovaniya/ (дата обращения 27.04.2025).
5. Руководство по организации архитектуры Python-приложений [Электронный ресурс]. – 2024. — URL: https://realpython.com/python-application-layouts/ (дата обращения 30.04.2025).

ПРИЛОЖЕНИЯ

Приложение 1. Модуль удаления инструмента

@bot.callback\_query\_handler(func=lambda call: call.data == 'delete\_instrument')

def delete\_instrument\_handler(call):

"""

Обработчик для удаления инструмента.

Отображает список доступных инструментов для удаления.

"""

chat\_id = call.message.chat.id

try:

# Отправляем сообщение о начале обработки

send\_or\_edit\_message(chat\_id, '\*Обработка запроса\*\n\nПолучаем список инструментов...')

# Получаем список всех инструментов через API-клиент

instruments = instruments\_client.get\_all\_instruments()

if not instruments:

send\_or\_edit\_message(chat\_id, '\*Удаление инструмента\*\n\nУ вас нет активных инструментов')

else:

inline\_keyboard = types.InlineKeyboardMarkup()

for instrument in instruments:

ticker = instrument.get('ticker')

button = types.InlineKeyboardButton(text=f" {ticker}", callback\_data=f'ticker\_{ticker}')

inline\_keyboard.add(button)

send\_or\_edit\_message(

chat\_id,

'\*Удаление инструмента\*\n\nВыберите инструмент для удаления:',

reply\_markup=inline\_keyboard

)

except Exception as e:

send\_or\_edit\_message(chat\_id, f'\*Ошибка при получении списка инструментов\*\n\n`{str(e)}`')

@bot.callback\_query\_handler(func=lambda call: call.data.startswith('ticker\_'))

def delete\_ticker\_step(call):

"""

Обработчик для удаления выбранного инструмента.

Удаляет инструмент по выбранному тикеру.

"""

chat\_id = call.message.chat.id

ticker = call.data.replace('ticker\_', '')

try:

# Отправляем сообщение о начале обработки

send\_or\_edit\_message(chat\_id, f'\*Обработка запроса\*\n\nУдаляем инструмент `{ticker}`...')

# Удаляем инструмент через API-клиент

instruments\_client.delete\_instrument(ticker)

send\_or\_edit\_message(chat\_id, f'\*Успешно\*\n\nИнструмент `{ticker}` успешно удален')

except Exception as e:

send\_or\_edit\_message(chat\_id, f'\*Ошибка при удалении инструмента\*\n\n`{str(e)}`')

Приложение 2. Код построения графика для сигнала SMA

def plot\_sma(chat\_id, df):

"""

Функция для построения свечного графика цены и сигналов SMA, и отправки его в Telegram.

:param chat\_id: Идентификатор чата в Telegram, куда нужно отправить график.

:param df: DataFrame с колонками 'time', 'open', 'high', 'low', 'close' для построения свечного графика цены.

"""

fast\_sma = store.fast\_sma

slow\_sma = store.slow\_sma

# Конвертация столбца 'time' в формат для matplotlib

df['time'] = pd.to\_datetime(df['time'])

df['time'] = df['time'].map(mdates.date2num)

fig, ax = plt.subplots(figsize=(14, 7))

ax.set\_title('Stock Price with SMA')

ax.set\_xlabel('Date')

ax.set\_ylabel('Price')

# Создаем список данных для candlestick\_ohlc

ohlc = df[['time', 'open', 'high', 'low', 'close']].values

candlestick\_ohlc(ax, ohlc, width=0.8, colorup='green', colordown='red')

# Добавление графиков SMA

ax.plot(df['time'], fast\_sma, label='Fast SMA', color='orange', linestyle='--')

ax.plot(df['time'], slow\_sma, label='Slow SMA', color='blue', linestyle='--')

# Форматирование дат на оси X

ax.xaxis.set\_major\_formatter(mdates.DateFormatter('%Y-%m-%d'))

fig.autofmt\_xdate()

# Добавление легенды

ax.legend(loc='upper left')

# Сохранение графика во временный файл

file\_path = 'sma\_candlestick\_chart.png'

plt.savefig(file\_path)

plt.close(fig)

with open(file\_path, 'rb') as photo:

bot.send\_photo(chat\_id, photo)