

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра математического и компьютерного моделирования

Автоматизация процесса торговли в сфере инвестирования

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 442 группы
направление 09.03.03 — Прикладная информатика

механико-математического факультета

Даренского Романа Алексеевича

Научный руководитель
к.ф.-м.н., доцент

подпись, дата

А.М. Донник

Зав. кафедрой
д.ф.-м.н., профессор

подпись, дата

Л.Ю. Коссович

Введение. Сложно переоценить влияние информационных технологий на современное общество. Они проникли во многие сферы нашей жизни, включая экономику и инвестиции. В этой области различные информационные системы играют важную роль.

Раньше для успешной инвестиционной деятельности требовалось множество квалифицированных специалистов. Они должны были уметь анализировать рыночные тенденции, изучать финансовые отчёты компаний, принимать на основе этого инвестиционные решения и проводить операции с акциями вручную.

Сейчас, в эпоху информационных технологий, с широким распространением персональных компьютеров и смартфонов, все эти задачи может выполнять один человек с доступом в Интернет.

Инвестиции - это один из самых эффективных способов сохранить и приумножить капитал. При правильном управлении рисками, они позволяют не только сберечь вложенные средства, но и увеличить их за счёт получения дохода в виде дивидендов, процентов по займам или благодаря росту стоимости активов.

Автоматизация процесса торговли в сфере инвестирования открывает новые возможности для индивидуальных инвесторов. Она позволяет максимально использовать свои ресурсы и время для достижения финансовых целей, а также сокращает количество ошибок, обусловленных человеческим фактором.

Актуальность данной темы обусловлена объективностью и точностью работы автоматизированных алгоритмов, экономией времени, а также уменьшением человеческого фактора в инвестициях.

Целью работы является создание торгового робота - приложения, предназначенного для автоматического инвестирования в ценные бумаги и другие торгуемые на финансовой бирже активы через платформу брокера.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- проведение анализа предметной области;
- описание структуры торгового робота;
- разработка и тестирование торгового робота.

Структура работы. Основная часть состоит из 3 разделов:

- анализ предметной области;
- проектирование;
- реализация.

В первом разделе содержится основная информация об инвестициях.

Инвестирование - это размещение капитала с целью получения прибыли.

В рамках работы были выбраны инвестиции в финансовые активы.

Финансовые активы являются особым видом собственности, который приносит доход своему владельцу. Они включают в себя разнообразные инструменты, такие как наличные деньги, банковские депозиты, вклады, чеки, страховые полисы, инвестиции в ценные бумаги, коммерческие кредиты, пакеты акций, дающие право на контроль над другими предприятиями, а также доли или участие в других компаниях.

Биржа - это площадка, где можно купить и продать различные активы, включая ценные бумаги, товары, сырьё, драгоценные металлы и валюту. Она организует торги и зарабатывает на комиссиях от каждой сделки. Биржа выполняет несколько важных функций: создаёт условия для взаимодействия продавцов и покупателей, следит за соблюдением правил торгов и выполнением договоров, аккредитует участников торгов, хранит информацию и документы о сделках, а также предоставляет отчётность о сделках Центральному Банку.

Чтобы торговать на бирже инвестор, должен обратиться к лицензированному брокеру.

Брокер - компания, которая обеспечивает клиенту доступ на рынок ценных бумаг и от его имени будет торговать на бирже.

Были выдвинуты требования к будущему приложению:

- кроссплатформенность;
- простая и понятная структура;

- наличие возможности добавления модулей к готовой версии для расширения функционала;
- возможность отмены всех выставленных, но не исполненных торговых поручений;
- максимальная независимость от пользователя после запуска;
- сохранение информации о совершённых сделках;
- возможность запуска и работы на серверах, подключённых к интернету.

Второй раздел относится к выбору брокера, программного обеспечения для разработки приложения и к созданию его структуры.

Для работы был выбран брокер «Тинькофф Инвестиции», так как помимо связи с использованием REST API этот брокер предоставляет gRPC-клиент.

gRPC (Remote Procedure Calls) - это система удалённого вызова процедур с открытым исходным кодом, первоначально разработанная в Google в 2015 году.

Связь через gRPC быстрее, чем через REST API, за счёт разницы технического устройства данных систем.

Брокером предоставляются готовые gRPC-клиенты для различных языков программирования.

В качестве языка программирования для разработки приложения был выбран Python версии 3.10.0.

Для хранения данных о выполненных торговых операциях выбрана система управления базами данных SQLite.

При помощи python-библиотеки aiogram был создан telegram-бот.

Telegram Messenger, широко известный как Telegram, представляет собой облачный, кроссплатформенный, зашифрованный сервис обмена мгновенными сообщениями.

Созданный бот будет отправлять информацию об исполненных торговых поручениях в личные сообщения инвестора. Также через ввод специальной

команды в чат с ботом сделана возможность остановить работу приложения удалённо.

Чтобы иметь возможность запускать приложение на любой платформе без дополнительной настройки, было решено прибегнуть к контейнеризации приложения.

Контейнеризация приложений - это способ упаковки приложений и их зависимостей в стандартизированный контейнер. Такой подход упрощает процесс разработки программного обеспечения, позволяя разработчикам быстрее и надёжнее создавать, тестировать и развёртывать приложения по сравнению с традиционными методами виртуализации.

Для данной работы была использована одна из самых популярных систем контейнеризации - Docker.

Docker - это платформа и технология контейнеризации. Она позволяет создавать контейнеры и управлять ими для развёртывания и доставки кода на сервер. Чтобы использовать Docker, нужно установить кроссплатформенное приложение Docker Desktop. Его можно использовать бесплатно в коммерческих целях.

Структура торгового робота состоит из трёх главных модулей:

1. Стартовый модуль, который принимает входные данные и запускает соответствующую стратегию для выбранного финансового актива.
2. Стратегический модуль, который выполняет операции по генерации сигналов на покупку или продажу финансового актива на основе полученных от брокера биржевых данных, а также выдаёт брокеру торговые поручения на основе данных сигналов.
3. Статистический модуль, который сохраняет данные о торговом поручении, а также обновляет их статус при необходимости.

На рисунке 1 представлена структура приложения и основные виды взаимодействия его компонентов между собой и со внешними сервисами. Пунктиром указаны необязательные взаимодействия.

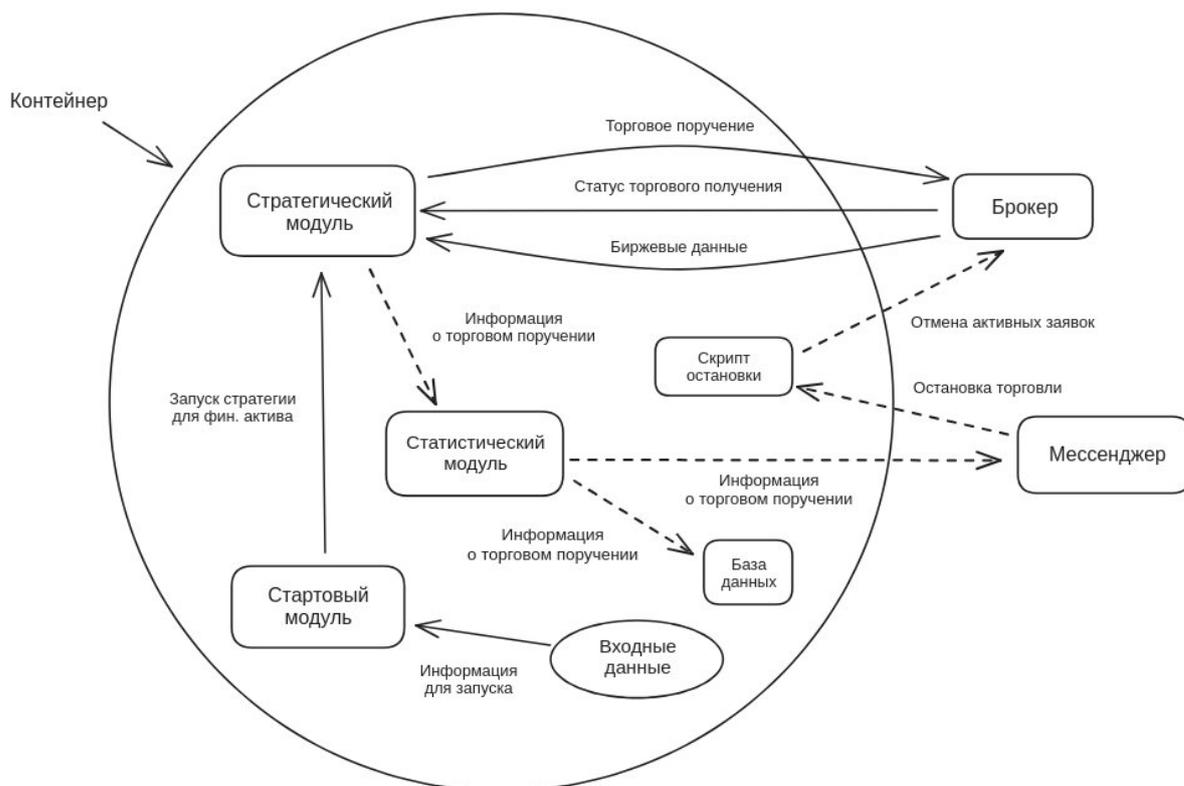


Рисунок 1 - Структура приложения

Третий раздел посвящён разработке приложения, его тестированию и контейнеризации.

Входные данные, поступающие на вход в стартовый модуль, содержат всю необходимую информацию для запуска приложения: токен доступа к брокеру, режим торговли (в сервисе «Песочница», где сделки не влияют на реальный брокерский счёт, или же на самом счёте), ID счёта, идентификатор финансового актива figi, название торговой стратегии вместе со всеми необходимыми для неё параметрами, а также название одного из обработчиков статистики (СУБД SQLite или telegram-бот) с необходимыми параметрами.

Стартовый модуль на основе входных данных для каждого финансового актива запускает соответствующую торговую стратегию вместе с выбранным обработчиком статистики. В качестве тестовой стратегии для приложения была реализована стратегия sma, основанная на скользящих средних.

Скользящее среднее – это инструмент технического анализа, который отображает линию на ценовом графике. Эта линия соединяет точки, рассчитанные как среднее значение цены за несколько последних периодов.

Основная задача этого индикатора — убрать краткосрочные колебания цены (так называемый «шум» на графиках котировок). По сути, скользящие средние работают как фильтр, который выделяет долгосрочные тенденции и убирает краткосрочные изменения.

При работе sma стратегии генерируются сигналы на покупку и продажу актива. Генерация происходит на основе скользящих средних за два периода времени разной длины. В рамках тестирования данного приложения будет использованы минутные промежутки.

Назовём скользящее среднее большего промежутка Long, а меньшего Short. Согласно стратегии, когда линия Short пересекает линию Long снизу вверх, то на бирже начинается тенденция по росту цены финансового актива. В этом случае мы имеем сигнал на покупку этого актива. Short пересекает Long сверху вниз - на бирже намечается тенденция на падение цены у актива. Следовательно, его надо продавать.

На рисунке 2 представлен график, иллюстрирующий на примере акций компании «Сбербанк» логику генерации сигналов на покупку и продажу. За промежуток времени взят день. Длина короткого скользящего среднего - 9 дней, длинного - 15 дней. Цены взяты с 18.10.23 по 06.02.24 включительно.



Рисунок 2 - Сигналы на покупку и продажу акций компании «Сбербанк»

Перед запуском стратегии проверяется, открыты ли на бирже в данный момент торги по выбранному финансовому активу. Если торги идут, то через gRPC происходит подписка на поток цен у брокера. Каждый раз при получении новой цены проверяются условия на покупку или продажу акции согласно торговой стратегии. Если необходимые условия выполнены, то брокеру посылается соответствующее торговое поручение. После исполнения поручения данные о нём поступают в статистический модуль.

В статистическом модуле в зависимости от выбранного обработчика данные о поручении либо записываются в базу данных, либо отправляются через telegram-бота в чат с инвестором.

На рисунке 3 представлен пример записи информации о выполненных торговых поручениях в базе данных.

	order_id	figi	direction	price	quantity	status
	Filter	Filter	Filter	Filter	Filter	Filter
1	52c7e7fd-e80e-43c1-a254-f9fd365ed555	BBG004730N88	ORDER_DIRECTION_BUY	3180.6	1	EXECUTION_REPORT_STATUS_FILL
2	c8b050f7-4f44-409e-860a-c04248a71c47	BBG004730N88	ORDER_DIRECTION_SELL	3181.3	1	EXECUTION_REPORT_STATUS_FILL
3	ca8c1313-95cb-4cca-ad07-762edc0146c8	BBG004730N88	ORDER_DIRECTION_BUY	3180.2	1	EXECUTION_REPORT_STATUS_FILL
4	8586e89f-52a6-42f8-8246-e8da0d143f73	BBG004730N88	ORDER_DIRECTION_SELL	3182.0	1	EXECUTION_REPORT_STATUS_FILL
5	0daec313-09d8-4ca0-8888-e632cbeb2fdc	BBG004730N88	ORDER_DIRECTION_BUY	3178.9	1	EXECUTION_REPORT_STATUS_FILL
6	2a298446-8b78-4ba8-bab4-17d4270f7d03	BBG004730N88	ORDER_DIRECTION_SELL	3180.2	1	EXECUTION_REPORT_STATUS_FILL

Рисунок 3 - Пример таблицы с данными о торговых поручениях

На рисунке 4 представлен пример информации о выполненных торговых поручениях в чате с telegram-ботом.

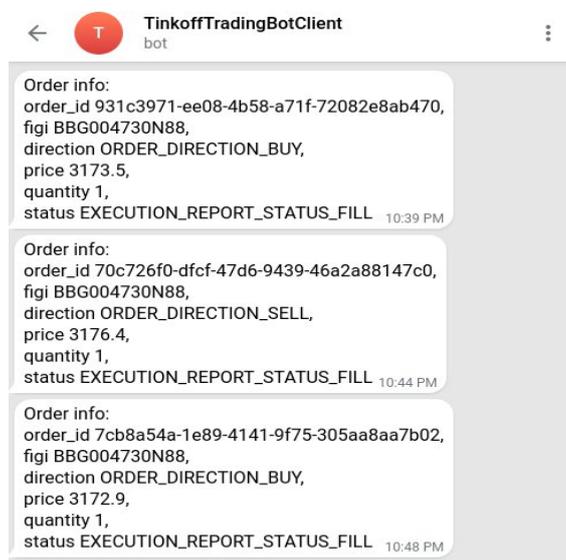


Рисунок 4 - Пример сообщений о торговых поручениях в чате с ботом

В процессе использования приложения может возникнуть необходимость остановки процесса торговли. Чтобы дать поручение брокеру на отмену всех невыполненных торговых поручений, был создан специальный скрипт.

В обработчике статистики TelegramHandler был реализован метод stop_trading. Когда пользователь отправляет в чат команду /stoptrading, метод выполняет скрипт остановки, отправляет в ответ сообщение об остановке торговли и останавливает работу приложения.

На рисунке 5 можно видеть сообщение об остановке торговли в чате.

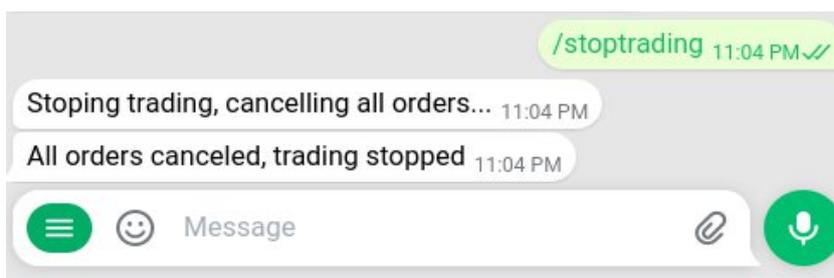


Рисунок 5 - Сообщение об остановке торговли в чате с ботом

Также был успешно создан Docker-образ, позволяющий выполнить контейнеризацию приложения с помощью Docker.

На рисунке 6 можно видеть, что запущенный контейнер теперь отображается в приложении Docker Desktop во вкладке Containers под названием trading-robot-container.

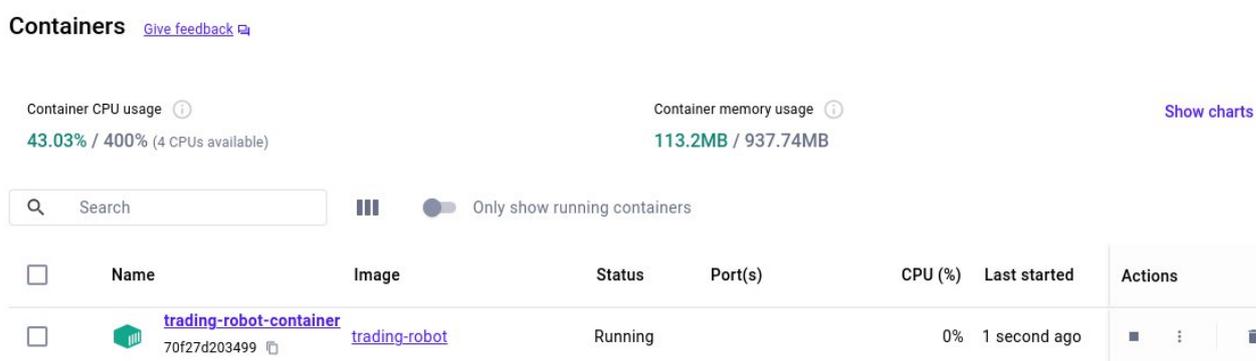


Рисунок 6 - Запущенный контейнер trading-robot-container

В приложениях представлены исходные программные коды реализации.

Заключение. В результате данной работы был разработан торговый робот для автоматического инвестирования через брокерскую платформу «Тинькофф Инвестиции».

Архитектура приложения спроектирована так, чтобы обеспечивать легкое добавление новых функций и возможностей. Это достигается за счет четкого разделения ответственности между различными компонентами системы, что позволяет избегать сложностей при интеграции с брокером и управлении данными о торговых операциях.

Для того чтобы приложение можно было запускать на разных устройствах и операционных системах, был создан Docker-образ. Он упрощает процесс установки и развёртывания приложения, так как не требует дополнительного программного обеспечения или настройки зависимостей.