**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Доцент департамента  программной инженерии  факультета компьютерных наук  Изображение выглядит как текст  Автоматически созданное описаниеканд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н. С. Белова 12 мая 2022 г. | УТВЕРЖДАЮ  Академический руководитель  образовательной программы  «Программная инженерия»  Профессор департамента программной инженерии, канд.техн.наук \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В.Шилов  12 мая 2022 г. |

Факультет компьютерных наук  
Департамент программной инженерии

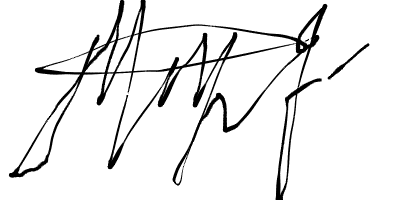
|  |  |
| --- | --- |
| *Подп. и дата* |  |
| *Инв. № дубл.* |  |
| *Взам. инв. №* |  |
| *Подп. и дата* |  |
| *Инв. № подл* |  |

**БИРЖЕВОЙ БОТ**

**Техническое задание**

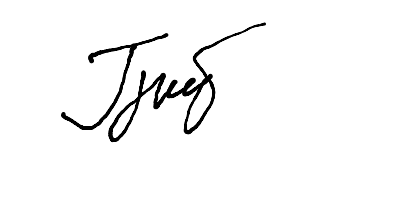
**ЛИСТ УТВЕРЖДЕНИЯ**

RU.17701729.05.01-01 ТЗ 01-1-ЛУ

**Исполнители:**Студент группы БПИ208  
\_\_\_\_\_\_\_\_\_\_ /Армаш В.В. /  
12 мая 2022 г

**Изображение выглядит как текст

Автоматически созданное описание**Студент группы БПИ201  
\_\_\_\_\_\_\_\_\_\_ /Ищенко А.С. /  
12 мая 2022 г

Студент группы БПМИ208  
\_\_\_\_\_\_\_\_\_\_ /Глинник А.М. /  
12 мая 2022 г

**Москва 2022**

УТВЕРЖДЕН

RU.17701729.05.01 -01 ТЗ 01-1-ЛУ

|  |  |
| --- | --- |
| *Подп. и дата* |  |
| *Инв. № дубл.* |  |
| *Взам. инв. №* |  |
| *Подп. и дата* |  |
| *Инв. № подл* |  |

**БИРЖЕВОЙ БОТ**

**Техническое задание**

RU.17701729.05.01 -01 ТЗ 01-1

**Листов 34**

**Москва 2022**

## АННОТАЦИЯ

Техническое задание – это основной документ, оговаривающий набор требований и порядок создания программного продукта, в соответствии с которым производится разработка программы, ее тестирование и приемка.

Настоящее Техническое задание на разработку мобильного приложения «Биржевой бот» содержит следующие разделы: «Введение», «Основание для разработки», «Назначение разработки», «Требования к программе», «Требования к программным документам», «Технико-экономические показатели», «Стадии и этапы разработки», «Порядок контроля и приемки» и приложения [1].

В разделе «Введение» указано наименование и краткая характеристика области применения программы.

В разделе «Основания для разработки» указан документ, на основании которого ведется разработка и наименование темы разработки.

В разделе «Назначение разработки» указано функциональное и эксплуатационное назначение программного продукта.

Раздел «Требования к программе» содержит основные требования к функциональным характеристикам, к надежности, к условиям эксплуатации, к составу и параметрам технических средств, к информационной и программной совместимости, к маркировке и упаковке, к транспортировке и хранению, а также специальные требования.

Раздел «Требования к программным документам» содержит предварительный состав программной документации и специальные требования к ней.

Раздел «Технико-экономические показатели» содержит ориентировочную экономическую эффективность, предполагаемую годовую потребность, экономические преимущества разработки программы.

Раздел «Стадии и этапы разработки» содержит стадии разработки, этапы и содержание работ.

В разделе «Порядок контроля и приемки» указаны общие требования к приемке работы.

Настоящий документ разработан в соответствии с требованиями:

1. ГОСТ 19.101-77 Виды программ и программных документов [1];
2. ГОСТ 19.102-77 Стадии разработки [2];
3. ГОСТ 19.103-77 Обозначения программ и программных документов [3];
4. ГОСТ 19.104-78 Основные надписи [4];
5. ГОСТ 19.105-78 Общие требования к программным документам [5];
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом [6];
7. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению [7].

Изменения к данному Техническому заданию оформляются согласно   
ГОСТ 19.603-78 [8], ГОСТ 19.604-78 [9].

Перед прочтением данного документа рекомендуется ознакомиться с терминологией, приведенной в соответствующем разделе настоящего технического задания.

СОДЕРЖАНИЕ

[АННОТАЦИЯ 2](#_Toc96891996)

[ТЕРМИНОЛОГИЯ 6](#_Toc96891997)

[1. ВВЕДЕНИЕ 8](#_Toc96891998)

[1.1. Наименование программы 8](#_Toc96891999)

[1.2. Краткая характеристика и область назначения 8](#_Toc96892000)

[2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ 9](#_Toc96892001)

[2.1. Документы, на основании которых ведется разработка 9](#_Toc96892002)

[2.2. Наименование темы разработки 9](#_Toc96892003)

[3. НАЗНАЧЕНИЕ РАЗРАБОТКИ 10](#_Toc96892004)

[3.1. Функциональное назначение 10](#_Toc96892005)

[3.2. Эксплуатационное назначение 10](#_Toc96892006)

[4. ТРЕБОВАНИЯ К ПРОГРАММЕ 11](#_Toc96892007)

[4.1. Требования к функциональным характеристикам 11](#_Toc96892008)

[4.1.1. Требования к составу выполняемых функций программы 11](#_Toc96892009)

[4.1.1.1. Функциональные требования к API для получения информации о ценах акций 11](#_Toc96892010)

[4.1.1.2. Функциональные требования к API-модулю, эмулирующему работу торгового API 12](#_Toc96892011)

[4.1.1.3. Функциональные требования к модели для прогнозирования цен на акции 12](#_Toc96892012)

[4.1.1.4. Функциональные требования к модулю принятия решений: 13](#_Toc96892013)

[4.1.1.5 Функциональные требования к основному модулю: 13](#_Toc96892014)

[4.1.1.6 Функциональные требования к интерфейсному модулю: 13](#_Toc96892015)

[4.1.2. Организация хранения данных 14](#_Toc96892016)

[4.1.3. Организация входных данных 14](#_Toc96892017)

[4.1.4. Организация выходных данных 14](#_Toc96892018)

[4.2. Требования к временным характеристикам 15](#_Toc96892019)

[4.3. Требования к интерфейсу 15](#_Toc96892020)

[4.4. Требования к надежности 15](#_Toc96892021)

[4.4.1. Требования к обеспечению надежного (устойчивого) функционирования программы 15](#_Toc96892022)

[4.4.2. Требования к корректности данных 15](#_Toc96892023)

[4.4.3. Отказы из-за некорректных действий оператора 15](#_Toc96892024)

[4.5. Условия эксплуатации 15](#_Toc96892025)

[4.5.1. Климатические условия эксплуатации 15](#_Toc96892026)

[4.6. Требования к составу и параметрам технических средств 16](#_Toc96892027)

[4.7. Требования к информационной и программной совместимости 16](#_Toc96892028)

[4.8. Требования к транспортировке и хранению 16](#_Toc96892029)

[4.8.1. Требования к транспортировке и хранению программных документов, предоставленных в электронном виде 16](#_Toc96892030)

[4.8.2. Требования к хранению и транспортировке программных документов, предоставляемых в печатном виде 16](#_Toc96892031)

[4.8.3. Требования к транспортировке и хранению программного продукта 17](#_Toc96892032)

[5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ 18](#_Toc96892033)

[5.1. Предварительный состав программной документации 18](#_Toc96892034)

[5.2. Специальные требования к программной документации 18](#_Toc96892035)

[6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 19](#_Toc96892036)

[6.1. Ориентировочная экономическая эффективность 19](#_Toc96892037)

[6.2. Предполагаемая потребность 19](#_Toc96892038)

[6.3. Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами 19](#_Toc96892039)

[7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ 21](#_Toc96892040)

[8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ 24](#_Toc96892041)

[8.1. Виды испытаний 24](#_Toc96892042)

[8.2. Общие требования к приёмке работы 24](#_Toc96892043)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 25](#_Toc96892044)

[ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 27](#_Toc96892045)

## ТЕРМИНОЛОГИЯ

* 1. API-модуль - программный интерфейс, посредством которого можно обратиться к данным. В данном случае— модуль, отвечающий за взаимодействие с API торговой площадки (биржи), осуществляющий периодические запросы к оному с целью получения актуальной информации о состоянии интересующих позиций.
  2. Агрегация запроса, предагрегация данных — процесс осуществления действий с данными, приводящий их к состоянию, необходимому для работы программы. Пример: перевод времени из человекочитаемого формата в миллисекунды с момента EPOCH для корректности сравнения с другими временами и датами внутри программы.
  3. Актор (здесь и далее в рамках проекта) - автономный асинхронный модуль, способный принимать сообщения и отвечать на них. Обладает базовым перечнем доступов и функций. Подробнее про работу с акторами можно прочитать в документации Thespian[13] [14].
  4. Акция (share, stock) - вид ценных бумаг.
  5. Визуализация — представление информации пользователю в читаемом виде, это может быть как выводимое на экран значение, так и график.
  6. Виртуальная транзакция — транзакция, не производящая действительных действий на бирже, но меняющая состояние виртуального счёта и баланса акций системы, и т.д.
  7. Виртуальный (здесь и далее в рамках проекта) — имитирующий работу с подлинными сущностями, действующий в рамках тех же правил и ограничений, имеющихся с настоящими объектами. То есть, виртуальный счёт — счёт в широком понимании этого слова, но никак не привязанный к живым деньгам пользователя и имитирующий работу оного.
  8. Восстановление после сбоя — автоматическое возобновление работы программы в случае критической ошибки, предполагает полное соответствие состояния работы программы и значений её данных оным до сбоя, т.е. отсутствие потери данных и т.п.
  9. Измерение эффективности модуля: среднее время обработки одной акции (время от получения запроса до выдачи результата)
  10. Лог — краткий отчёт унифицированного вида о действиях и состоянии системы в данный момент времени.
  11. Логгер — система непрерывного автоматического создания логов.
  12. Модель акторов — модель параллельных вычислений, строящаяся вокруг понятия актора (англ. actor «актёр; действующий субъект»), считающегося универсальным примитивом параллельного исполнения. Актор в данной модели взаимодействует путём обмена сообщениями с другими акторами, и каждый в ответ на получаемые сообщения может принимать локальные решения, создавать новые акторы, посылать свои сообщения, устанавливать, как следует реагировать на последующие сообщения.
  13. Модуль — независимая единица исполнения внутри программы (отдельный поток/процесс), действующий в соответствии с определённым стандартом входных и выходных данных, с возможностью полноценного, в том числе нагрузочного тестирования в отрыве от остальной программы, взаимозаменяемый с любыми другими модулями, действующими в рамках аналогичных стандартов ввода/вывода.
  14. Наблюдаемые позиции (акции) - перечень позиций (акций), добавленных пользователем для наблюдения/отслеживания и совершения программой операций купли/продажи.
  15. Позиция (также встречается под обозначениями „акция”, „метрика”) — интересующая нас единица торговли, в отношении которой проводится работа системы в том или ином виде. В зависимости от контекста может подразумевать как наименование позиции, так и конкретную единицу с таким названием (одну акцию, например), а также временной ряд с информацией о динамике изменения параметров данной позиции (количество лотов, текущая цена, и т.п.)
  16. Полный перечень акций – перечень акций, получаемый с помощью выбранных API
  17. Свободные денежные средства - те, которые остаются у владельца после уплаты всех текущих обязательств.
  18. Тик (англ. tick) — единица времени.
  19. Токен (в контексте данного проекта) – набор символов, в котором зашифрована информация о владельце, правах доступ и прочей необходимой для авторизации в TINKOFF INVEST API [11] информации.
  20. Торговый API — предоставляемый биржей API для осуществления операций торговли на ней, а также получения актуальной информации о статусе имеющихся позиций на ней.
  21. Транзакция —процедура разового обмена информацией/деньгами с обратной связью, подтверждающей успешное её проведение. В рамках проекта мы рассматриваем транзакции, удовлетворяющие требованиям ACID.
  22. Трейдинг - совершение торговых операций на биржевом рынке с целью получения прибыли.
  23. Фондовая биржа - площадка для торговли ценными бумагами.
  24. FIGI, figi, фиджи (англ.: Financial Instrument Global Identifier) — глобальный идентификатор финансового инструмента.
  25. Open-source – проекты с открытым исходным кодом. Разрабатываются преимущественно на некоммерческих началах. Часто проекты публикуются именно так, чтобы появилась возможность улучшить качество продукта посредством его доработки и изучения совместными усилиями.

# ВВЕДЕНИЕ

# Наименование программы

Наименование программы – «Биржевой бот».

Наименование программы на английском языке – «Stock bot».

# Краткая характеристика и область назначения

«Биржевой бот» - автоматизированная распределённая (модульная, многопоточная) система на базе машинного обучения для (виртуальной) торговли на бирже и составления кратковременных прогнозов поведения позиций.

Принято считать, что для получения прибыли на бирже нужно много знаний, интуиция опытного трейдера или проприетарные биржевые роботы с многомиллионными бюджетами разработки. Проект призван развеять это предубеждение и показать, что даже самостоятельно разработанные автономные системы могут генерировать приятную прибыль при намного меньшем пороге входа.

# ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

# Документы, на основании которых ведется разработка

Учебный план подготовки бакалавров по направлению 09.03.04 «Программная инженерия», 01.03.02 «Прикладная математика и информатика» НИУ ВШЭ и утвержденная академическим руководителем тема выбранного курсового проекта.

# Наименование темы разработки

Наименование программы – «Биржевой бот».

Условное обозначение темы разработки – «Stock bot».

Программа выполняется в рамках темы курсовой работы в соответствии с учебным планом подготовки бакалавров по направлениям 09.03.04 «Программная инженерия», 01.03.02 «Прикладная математика и информатика» Национального исследовательского университета "Высшая школа экономики", факультет компьютерных наук.

# НАЗНАЧЕНИЕ РАЗРАБОТКИ

# Функциональное назначение

«Биржевой бот» представляет собой программу(бота), который в реальном времени совершает виртуальные операции торговли на бирже и логирует происходящее.

Программа реализует набор функций, связанных с виртуальной торговлей: операции с денежными средствами на виртуальном счету, операции с акциями, предоставление информации о текущем размере средств, визуализация текущей работы системы и графиков цен\прогноза в реальном времени.

# Эксплуатационное назначение

Программа ориентирована на рынок ценных бумаг. «Биржевой бот» предоставляет пользователям возможность самостоятельно задавать список наблюдаемых акций для торговли, видеть графики состояния счёта и показатели прибыли, добавлять и изымать деньги с виртуального счета.

Целевой аудиторией проекта являются люди, которые непосредственно связаны с инвестициями – проект потенциально может развиться в «боевого» бота, позволяющего людям автоматизировать торговлю на рынках ценных бумаг, криптовалют и т.п. логируемых позиций.

# ТРЕБОВАНИЯ К ПРОГРАММЕ

# Требования к функциональным характеристикам

# Требования к составу выполняемых функций программы

Программа состоит из 2 частей (6 модулей).

Общие функциональные требования к программе:

1. Произведение виртуальной торговли для выбранных акций
2. Добавление и удаление наблюдаемых акций для торговли ими
3. Добавление и изъятие денег с виртуального счёта
4. Предоставление информации о текущем размере средств (общее количество, вложенные и свободные) – отображение в интерфейсе программы
5. Предоставление информации о прибыли за выбранный период времени – отображение в интерфейсе программы
6. Визуализация состояния счета в реальном времени - график, обновляющийся каждую секунду, а также отображение текущего
7. Визуализация текущей работы системы и графика цен/прогноза цен в реальном времени
8. Эффективность при работе с большим количеством наблюдаемых позиций (не более трёх секунд на получение и обработку текущих цен, построение прогнозов и принятие решения без учёта времени ответа сервера для получения цен)

Core-часть: API для получения информации о ценах акций; API-модуль, эмулирующий работу торгового API; Модель для прогнозирования цен на акции в будущем; Модуль принятия решений; Основной модуль.

UI-часть: Интерфейсный модуль.

# Функциональные требования к API для получения информации о ценах акций

1. Налаженное взаимодействие с выбранным API. Выбор пал на официальный клиент Tinkoff Invest [12]. Мотивация выбора описана в приложении 4.
2. Обработка и предоставление полученной информации о позициях(акциях) в установленных форматах (подробнее о форматах и запрашиваемой информации в развернутом списке требований, предоставленном в приложении 3)
3. Эффективность при работе с большим количеством наблюдаемых акций (агрегация ответа сервера не более секунды)
4. Передача информации о наличии неполадок на стороне сервиса, предоставляющего информацию о ценах (передача сообщений основному модулю о наличии неполадок)
5. Проверка входных данных об акциях, поступающих от основного модуля, на корректность и существование (т. е. решение ситуаций, когда поступает запрос на несуществующие позиции\ запрос на продажу\покупку невозможного количества акций - считаем невозможными те данные, которые не кратны лоту или <0).
6. Восстановление после сбоя (см.п.4.4. «Требования к надежности»)

# Функциональные требования к API-модулю, эмулирующему работу торгового API

В этом модуле логируется запись транзакций.

1. Совершение виртуальных операций купли/продажи на бирже в соответствии с принятым основным модулем решением (в рамках проекта совершение виртуальной операции мы осуществляем через логирование транзакций)
2. Предоставление информации о совершенных операциях в формате [время- позиция - количество единиц - тип операции(купля/продажа) - цена за единицу]
3. Запись истории транзакций в файл [статус транзакции: success\failure, информация об операции – см.п.2 текущего списка].
4. Обработка некорректных\неактуальных данных – передача оповещения основному модулю, продолжение работы программы (за счет игнорирования данных, признанных некорректными).  
   Статус транзакции логируется как failure.

# 4.1.1.3. Функциональные требования к модели для прогнозирования цен на акции

1. Построение прогнозов цен для наблюдаемых позиций в соответствии с последними полученными данными о текущих ценах
2. Оценка точности предоставляемых прогнозов
3. Хранение модели прогнозирования для каждой наблюдаемой позиции
4. Предоставление основному модулю программы прогнозов и оценки их точности (в виде [<название>, <цена>, <timestamp>, <среднеквадратичное отклонения для данной позиции>])
5. Эффективность (не более секунды на построение прогноза по позиции, не более двух часов на обучение модели)

# 4.1.1.4. Функциональные требования к модулю принятия решений:

1. Поиск и интеграция актор-системы с открытым исходным кодом и свободной лицензией, либо написание собственной в соответствии с установленными критериями (асинхронность, скорость, кроссплатформенность, устойчивость к падениям отдельных акторов и автоматический их перезапуск, возможность низкоуровневого логирования сообщений, возможность развёртывания на нескольких устройствах с сопутствующим общением модулей посредством встроенного либо внешнего http/grpc-клиента)
2. Поиск и интеграция тактового генератора и механизма реализации отложенных событий с открытым исходным кодом и свободной лицензией, либо написание собственного в соответствии с установленными критериями (асинхронность, нетребовательность в ресурсах процессора, обеспечение точности тактов/реализации событий с погрешностью не более секунды)
3. Выбор формата и способа взаимодействия модулей программы между собой.

В первую очередь необходимо реализовать базовый актор, бесшовно интегрирующийся с имеющейся актор-системой (об этом ниже, в требованиях к «основному модулю»), создать формат сообщений и наладить взаимодействие данного модуля с другими модулями настоящей программы.

Основные модели поведения модуля (неразрывны с другими модулями программы):

1. Получение прогнозов от модуля прогнозирования в определённом формате (временной ряд вида «время — предполагаемая цена») с последующей оценкой рисков и вероятностей ошибок (линейная регрессия вероятности по времени, квадратичная по цене)
2. Построение торговой стратегии по текущей позиции в соответствии с полученными прогнозами и рисками, а также комиссией по итогам п.1
3. В соответствии со стратегией из п.3 создание отложенных событий и отправка оных в соответствующую часть основного модуля, посредством которых осуществляется непосредственное управление торговым модулем (эмулятором торгового API) и следствием исполнения которых является совершение операций купли-продажи.
4. (Опционально) в зависимости от задержки исполнения модуля прогнозирования прямой запрос к модулю API о текущем состоянии наблюдаемой позиции с целью исключения действий по устаревшей стратегии и прогнозам.

Обязательна проверка поступающих запросов на корректность, т.е. на актуальность и соответствие детерминированному формату сообщений, а также автоматический перезапуск модуля в случае его падения по каким-либо обстоятельствам (осуществляется актор-системой)

# Функциональные требования к основному модулю:

Основной модуль должен содержать в себе актор-систему и осуществлять управление оной. Для данного проекта выбрана актор-система Thespian [13] [14], поскольку та отвечает всем обязательным требованиям и является удобной в использовании. Дополнительно необходимо наличие тактового генератора и механизма реализации отложенных событий. Было принято решение сделать оные на базе минималистичной библиотеки schedule, так как её функционала достаточно для данного проекта.

Базовые механизмы и действия данного модуля (неразрывны с другими модулями программы):

1. Получение запроса от интерфейсного модуля на добавление/удаление наблюдаемых акций
2. В соответствии с п.1 создание и инициализация цепочки исполнения для каждой конкретной акции, состоящей из модуля прогнозирования и модуля принятия решений
3. Исполнение отложенных действий, т.е. отправка отложенных сообщений в необходимый момент времени тем акторам, которым они предназначались изначально
4. Осуществление передачи сообщений между модулями посредством актор-системы
5. Создание, управление, выключение и удаление акторов посредством актор-системы
6. Содержание информации о балансе и наблюдаемых позициях, а также предоставление данной информации по запросу от интерфейсного модуля, а также периодическая актуализация информации о балансе средств посредством модуля API
7. Исполнение всех операций, затребованных интерфейсным модулем, либо переадресация их ответственным за это модулям.

# Функциональные требования к интерфейсному модулю:

1. Наличие пользовательского интерфейса, см. «приложение 1»
2. Предоставление пользователю возможности добавления и удаления наблюдаемых позиций (кнопки с окном для ввода названий)
3. Предоставление пользователю возможности добавления и изъятия средств с виртуального счёта (кнопки с окном для ввода суммы)
4. Отображение текущего количества средств на счёте
5. Отображение информации о прибыли (по умолчанию в процентах за последнюю неделю, пользователь может задать свой период)
6. Визуализация состояния счета (график количества средств на счёте за неделю и за день, пользователь может задать свои периоды)
7. Вывод на экран всех сообщений об ошибках, получаемых от основного модуля
8. Эффективность при работе с большим количеством наблюдаемых позиций (все данные, выводящиеся на экран, должны обновляться не дольше секунды)
9. Устойчивость к некорректным входным данным (например, при вводе некорректной суммы для добавления или изъятия средств) и вывод соответствующего сообщения на экран
10. Осуществление записи текущего состояния в файл и его последующее восстановления в случае сбоя

# Организация хранения данных

Хранение данных каждый из перечисленных в п.4.1.1. модулей обеспечивает самостоятельно.

Организация хранения данных для каждого модуля описана в технических заданиях, касающихся конкретных модулей.

# Организация входных данных

Формат организации входных данных определяется каждым модулем самостоятельно и описывается соответственно в пункте «Организация входных данных» в отдельных технических заданиях. Предварительный формат входных данных кратко описан для каждого модуля в п.4.1.1.

# Организация выходных данных

Выходные данные системы описаны в пункте 4.1.1.6. («Функциональные требования к интерфейсному модулю»).

При условии, что входные данные были некорректными выходными данными будет являться сообщение об ошибке. Корректность данных, формат организации выходных данных определяется каждым модулем самостоятельно и описывается соответственно в пунктах «Требования к корректности данных», «Организация выходных данных» в отдельных технических заданиях.

# Требования к временным характеристикам

Скорость работы системы зависит от скорости интернет-подключения и скорости работы алгоритма. Скорость работы алгоритма зависит от объема поступающих данных.

Эффективной работа системы будет считаться, если агрегация ответа сервера длится не более секунды.

# Требования к интерфейсу

Интерфейс программы должен быть реализован на русском языке.

Требования к интерфейсному модулю описаны в пункте 4.1.1.6.

Все исключительные ситуации, которые могут возникнуть в программе, должны быть обработаны так, чтобы было интуитивно понятно, в чем заключалась проблема.

# Требования к надежности

# 4.4.1. Требования к обеспечению надежного (устойчивого) функционирования программы

При любых действиях пользователя и при любых входных данных (т. е. при условии работоспособности устройства и отсутствия попыток со стороны пользователя и/или ОС принудительно завершить программный процесс), система не должна завершаться из-за ошибки. Все исключения, возникающие в процессе работы программы, должны быть обработаны и не должны прерывать процесс.

# 4.4.2. Требования к корректности данных

Программа должна обеспечивать проверку корректности входных данных.

Корректность входных данных определяется для каждого модуля отдельно, она описывается в соответствующем техническом задании и в п.4.1.2. («Организация входных данных»)

# 4.4.3. Отказы из-за некорректных действий оператора

Отказ может возникнуть при условии повреждения исходного кода и\или при условии попытки принудительно завершить программный процесс.

# Условия эксплуатации

## 4.5.1. Климатические условия эксплуатации

Климатические условия эксплуатации должны удовлетворять требованиям, предъявляемым к техническому устройству, используемому пользователем.

Требования к эксплуатации технического устройства предъявляются производителем технического устройства.

# Требования к составу и параметрам технических средств

Запуск и стабильная работа должна быть обеспечены при:

1. Unix-система
2. 8+ ядер/16+ потоков
3. 8+ Гб ОЗУ
4. 4+ Гб ПЗУ
5. Доступ к интернету.

# Требования к информационной и программной совместимости

Для функционирования системы требуются следующие программные совместимости:

1. Unix-система
2. Предустановленный Python версии 3.9 или выше
3. Доступ к жесткому диску
4. Доступ к интернету.

# Требования к транспортировке и хранению

# Требования к транспортировке и хранению программных документов, предоставленных в электронном виде

Программные документы в электронном виде загружаются в систему систему Smart LMS (Smart Learning Management System). Требования к хранению и транспортировке не предъявляются.

Программа распространяется в виде электронного пакета, содержащего программную документацию, приложение (исполняемые файлы и прочие необходимые для работы файлы).

# Требования к хранению и транспортировке программных документов, предоставляемых в печатном виде

Программные документы, предоставляемые, в печатном виде, должны соответствовать общим правилам учета и хранения программных документов, предусмотренных стандартами ЕСПД (Единой системы программной документации) и соответствовать требованиям ГОСТ 19.602–78[9]

# Требования к транспортировке и хранению программного продукта

Программное изделие может храниться и транспортироваться на USB-носителе или в облачном хранилище.

# ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

# Предварительный состав программной документации

1. «Биржевой бот». Техническое задание (ГОСТ 19.201-78);
2. «Биржевой бот». Программа и методика испытаний (ГОСТ 19.301-78);
3. «Биржевой бот». Текст программы (ГОСТ 19.401-78);
4. «Биржевой бот». Пояснительная записка (ГОСТ 19.404-79);
5. «Биржевой бот». Руководство оператора (ГОСТ 19.505-79);

# Специальные требования к программной документации

1. Документы к программе должны быть выполнены в соответствии с ГОСТ 19.106-78 и ГОСТами к каждому виду документа (см. п. 5.1.);
2. Пояснительная записка должна быть загружена в систему Антиплагиат через LMS «НИУ ВШЭ». Лист, подтверждающий загрузку пояснительной записки, сдается в учебный офис вместе со всеми материалами не позже, чем за день до защиты курсовой работы;
3. Вся документация также воспроизводится в печатном виде, она должна быть подписана академическим руководителем образовательной программы 09.03.04 «Программная инженерия», руководителем разработки и исполнителем перед сдачей курсовой работы в учебный офис не позже одного дня до защиты;
4. Документация также сдается в электронном виде в формате .pdf или .docx. проект программы в архиве формата .zip или .rar;
5. Все документы перед защитой курсовой работы должны быть загружены в информационно-образовательную среду НИУ ВШЭ систему Smart LMS (Smart Learning Management System), в личном кабинете дисциплина - «Курсовой проект, 2 курс ПИ», одним архивом.

# ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

# Ориентировочная экономическая эффективность

В рамках данной работы оценка экономической эффективности не предусмотрена.

Если бот будет генерировать разумную прибыль, но не превосходить буквально существующие аналоги - проект можно считать успешным.

# Предполагаемая потребность

Низкий порог входа для использования программы позволит пользователям без детального изучения сферы трейдинга попробовать себя в роли инвесторов.

Опытные трейдеры используют роботов для торговли на бирже в стабильные периоды, когда можно подобрать подходящие настройки с использованием методов финансового анализа.

# Экономические преимущества разработки по сравнению с отечественными и зарубежными аналогами

Преимущества [19] бота это:

* автоматизация процессов биржевой торговли: не нужно безостановочно отслеживать графики и индикаторы
* пассивный доход: правильно настроенный бот регулярно приносит прибыль
* масштабируемость: программа охватывает гораздо больше торговых пар, чем трейдеры
* строгое следование заданному алгоритму: модуль не подвержен сомнениям и панике, он безошибочно выполнит требуемую последовательность действий
* мгновенная реакция на поступающие сигналы: меньше шансов пропустить выгодную сделку.

Недостатки торгового бота [20]:

* сложность: на создание и тестирование алгоритма уходят месяцы
* технический анализ: бот не способен учитывать и оценивать влияние важных новостей на рынок
* быстрое устаревание: если количество убыточных сделок стало чрезмерным - выбранная стратегия торговли уже не работает.

В приложении 2 приведена сравнительная таблица аналогов проекта.

Рассмотрим решения на рынке, имеющие аналогичную направленность [21]:

1. **Tickeron.**  Позволяет пользователям самостоятельно настраивать используемые нейронные сети, что при должном изучении предоставляемой программой информации может привести к хорошим показателям прибыли. Из недостатков можно отметить то, что возможности программы ограничиваются отправкой пользователям уведомлений о купле/продаже позиций, что делает затруднительным получение прибыли на краткосрочных периодах, так как требует своевременного вмешательства пользователя.
2. **Tradetron.** Платформа для алгоритмической торговли, кроме использования предустановленных алгоритмов, как и Tickeron предоставляет возможность создания собственных стратегий, а также обмена стратегиями между пользователями. В отличие от Tickeron, Tradetron является платформой для автоматизированной торговли и уведомляет пользователя уже о произведённых операциях.
3. **Streak.** Предоставляет стратегии для краткосрочной и долгосрочной торговли, основанные на математическом и техническом анализе. Аналогично решениям выше, позволяет пользователю создавать собственные стратегии. Особое внимание разработчики уделили быстрому тестированию стратегий на реальных данных в прошлом (backtesting). Платформа позволяет производить виртуальную торговлю или реальную, как с помощью уведомлений, подобно Tickeron, так и полностью автоматизированную, подобно Tradetron.

Все рассмотренные выше решения предоставляют множество стратегий с последующей возможной настройкой пользователем. В рамках проекта мы сами берём ответственность за принятые ботом решения по каждой конкретной позиции с учётом её специфики.

Нужно уточнить, что сравнивать «Биржевого бота» с существующими промышленными роботами бессмысленно, т. к. они зачастую имеют миллионные бюджеты разработки в долларах и крайне высокий порог входа для использования. Тема является довольно сложной, а экономика переживает не лучшие свои времена - сравнивать текущую статистику разработанного в краткие сроки бота с открытой (чаще всего нерелевантной) статистикой открытых ботов также неразумно.

Безусловным преимуществом нашей разработки является низкий порог входа для пользователей.

# СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Стадии и этапы разработки были выявлены с учетом ГОСТ 19.102-77 [2]:

|  |  |  |  |
| --- | --- | --- | --- |
| Стадии разработки | Этапы работ | Содержание работ | Временной период и исполнители |
| 1. Техническое задание | Подготовительные работы | * Постановка задачи. * Сбор исходных теоретических материалов. * Обоснование возможности решения поставленной задачи. * Определение структуры входных и выходных данных. * Предварительный выбор методов решения задач. | 01.12.2021-31.12.2021  Армаш В.В.  Ищенко А. С.  Глинник А.М. |
| Разработка и утверждение технического задания | * Определение требований к программе. * Определение требований к техническим средствам. * Определение стадий. Этапов и сроков разработки программы и документации на неё. * Согласование и утверждение технического задания. | 01.01.2022 - 28.02.2022  Армаш В.В.  Ищенко А. С.  Глинник А.М. |
| 2. Рабочий проект | Разработка core- части приложения(«бэкенд») (15.02.22 – 29.04. 22) | 1. API для получения информации о ценах акций 2. Блок виртуальной торговли (эмулятор торгового API) 3. Модель для прогнозирования цен на акции 4. Модуль принятия решений 5. Основной модуль | 28.02.2022 - 30.04.2022  1,2 – Армаш В.В.  3 – Глинник А.М.  4,5 – Ищенко А.С. |
| Разработка UI-части программы(«фронтенд»)  (10.03.22 – 29.04.22) | Интерфейсный модуль | 28.02.2022 - 30.04.2022  Глинник А.М. |
| Разработка программной документации | Разработка программных документов в соответствии с требованиями ГОСТ 19.101-77 | 25.03.2022 - 30.04.2022  Армаш В.В. |
| Испытания программы | * Разработка, согласование и утверждение порядка и методики испытаний. * Корректировка программы и программной документации по результатам испытаний. | 30.04.2022 - 09.05.2022  Армаш В.В.  Ищенко А. С.  Глинник А.М. |
| 3. Внедрение | Подготовка и передача программы | Подготовка и передача программы и программной документации для сопровождения. | с 10.05.2022  Армаш В.В.  Ищенко А. С.  Глинник А.М. |

Программа и документация к ней разрабатываются к утвержденным срокам сдачи курсового проекта (13.05.2022).

# ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

# Виды испытаний

Производится проверка корректного выполнения программой заложенных в нее функций, то есть осуществляется функциональное тестирование программы. Также осуществляется визуальная проверка интерфейса программы на соответствие с характеристиками интерфейсного модуля и приложением в содержании технического задания. Функциональное тестирование осуществляется в соответствии с документом «Биржевой бот». Программа и методика испытаний (ГОСТ 19.301-79), в котором указывают:

1. Перечень функций программы, выделенных в программе для испытаний, и перечень требований, которым должны соответствовать эти функции (со ссылкой на пункт 4.1.1. настоящего технического задания);
2. Перечень необходимой документации и требования к ней (со ссылкой на пункт 5 настоящего технического задания);
3. Методы испытаний и обработки информации;
4. Технические средства и порядок проведения испытаний;

Защита выполненного проекта осуществляется комиссии, состоящей из преподавателей департамента программной инженерии, в утверждённые приказом декана факультета компьютерных наук сроки.

# 8.2. Общие требования к приёмке работы

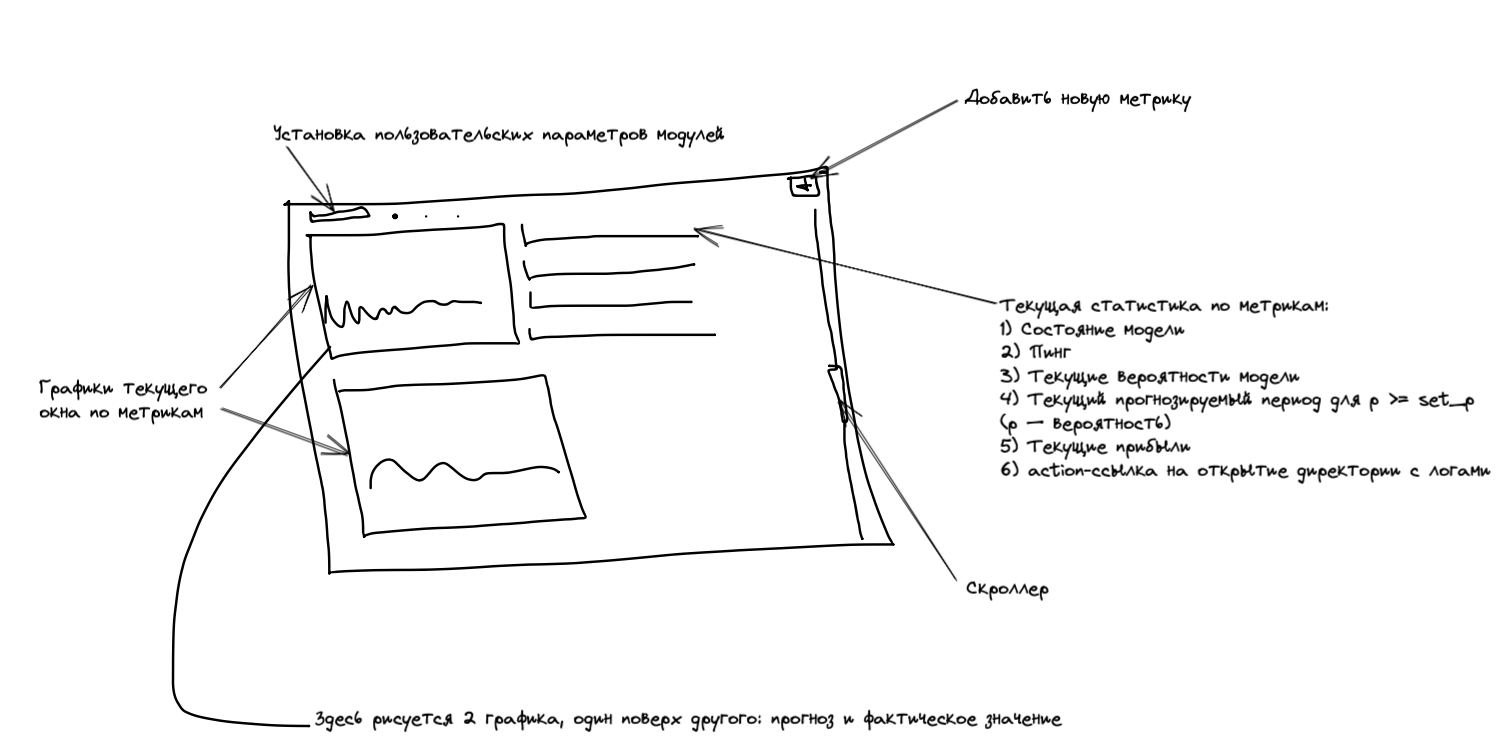
Прием программного продукта происходит при полной работоспособности программы при различных входных данных, при выполнении указанных в пункте 4.1.1. настоящего документа функций, при выполнении требований указанных в пункте 4.2. настоящего документа и при наличии полной документации к программе, указанной в пункте 5.1., выполненной в соответствии со специальными требования указанными в пункте 5.2. настоящего технического задания.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 19.101-77 Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603-78 Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. Tinkoff Invest Api Token// [Электронный ресурс] - Режим доступа: свободный, URL: <https://tinkoff.github.io/investAPI/token/> (5 мая 2022)
11. Тинькофф инвестиции // [Электронный ресурс] - Режим доступа: свободный, URL: <https://www.tinkoff.ru/invest/> (5 мая 2022)
12. Тинькофф клиент API // [Электронный ресурс] - Режим доступа: свободный, URL: <https://tinkoff.github.io/invest-python/> (5 мая 2022)
13. Thespian Actors User`s Guide// [Электронный ресурс] - Режим доступа: свободный, URL: <https://thespianpy.com/doc/using.html> (5 мая 2022)
14. Thespian Actors Library// [Электронный ресурс] - Режим доступа: свободный, URL: <https://github.com/kquick/Thespian>(5 мая 2022)
15. Finage.co.uk - предоставляет информацию в реальном времени для всех бирж в Соединенных Штатах и для внебиржевых рынков //[Электронный ресурс] - Режим доступа: свободный, URL: <http://finage.co.uk/> (5 мая 2022)
16. **Yahoo Finance API** // [Электронный ресурс] - Режим доступа: свободный, URL: [https://www.yahoofinanceapi.com](https://www.yahoofinanceapi.com/) (5 мая 2022)
17. Alpha Vantage API // [Электронный ресурс] - Режим доступа: свободный, URL:  
    <https://www.alphavantage.co/documentation/> (5 мая 2022)
18. EXANTE API // [Электронный ресурс] - Режим доступа: свободный, URL:  
    <https://exante.eu/ru/technology/> (5 мая 2022)
19. Альфа-Банк, торговые роботы // [Электронный ресурс] - Режим доступа: свободный, URL: <https://alfabank.ru/make-money/investments/torgovye-roboty/> (5 дек. 2021)
20. Обзор ботов для торговли криптовалютой // [Электронный ресурс] - Режим доступа: свободный, URL: <https://crypto.ru/boty-dlya-torgovli-kriptovalyutoy/> (5 дек. 2021)
21. Best Stock Trading Bots // [Электронный ресурс] - Режим доступа: свободный, URL: <https://sourceforge.net/software/stock-trading-bots/> (5 дек. 2021)
22. Trade Ideas // [Электронный ресурс] - Режим доступа: свободный, URL: <https://sourceforge.net/software/product/Trade-Ideas/> (5 дек. 2021)
23. Stoic AI // [Электронный ресурс] - Режим доступа: свободный, URL: <https://sourceforge.net/software/product/Stoic/> (5 дек. 2021)

**ПРИЛОЖЕНИЕ 1**

Общая концепция интерфейса программы



*Рис. - Схема интерфейса.*

**ПРИЛОЖЕНИЕ 2**

# СРАВНИТЕЛЬНАЯ ТАБЛИЦА АНАЛОГОВ ПРОЕКТА «БИРЖЕВОЙ БОТ»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Tickeron | Trade Ideas[22] | Stoic AI[23] | Streak | **Биржевой бот** |
| Наличие открытого исходного кода | **-** | **-** | **-** | **-** | **+** |
| Модульность программы | **-** | **-** | **-** | **-** | **+** |
| Заявленное использование актор-системы как модели параллельных вычислений | **-** | **-** | **-** | **-** | **+** |
| Кроссплатформенность | **+** | **+** | **+** | **+** | **-\*\*** |
| Бесплатная подписка | **-** | **-** | **+** | **-** | **+** |
| Бесплатный пробный период | **-** | **+** | **+** | **+** | **+** |
| Операции с криптовалютой(криптотрейдинг) | **+** | **-** | **+** | **+** | **-\*** |
| Отображение (в виде временного ряла) детального прогноза real-time | **-** | **-** | **-** | **-** | **+** |
| Возможность виртуальной торговли | **-** | **-** | **-** | **+** | **+** |

\*Благодаря модульности программы может быть добавлен соответствующий модуль  
\*\*Может быть собрано пользователем на любой системе на свой страх и риск

**ПРИЛОЖЕНИЕ 3**

# Развернутый список требований к API для получения информации о ценах акций

Использование токена для работы с Tinkoff API [12]. Токен генерируется в личном кабинете сайта Тинькофф Инвестиции [11], срок жизни токена – 3 месяца с даты последнего использования. В проекте используется readonly token[10] — он предназначен только для получения различной информации, такой как состояние портфеля, расписание торгов различных торговых площадок, текущие котировки, исторические данные и т.п., а выставление торговых поручений невозможно.

1. Предоставление модулям программы полного перечня акций.  
   Предоставляется: время запроса – общее для всех акций, figi\имя\цена\валюта\статус покупки\статус продажи\лот – для каждой акции.
2. Предоставление перечня акций, доступных к покупке.  
   Предоставляется: время запроса – общее для всех акций, figi\имя\цена\валюта\статус покупки\статус продажи\лот – для каждой акции. Отличие от предыдущего пункта - статус покупки у акций в предоставляемом перечне всегда true.
3. Предоставление перечня акций, доступных к продаже.  
   Предоставляется: время запроса – общее для всех акций, figi\имя\цена\валюта\статус покупки\статус продажи\лот – для каждой акции. Отличие от предыдущего пункта - статус продажи у акций в предоставляемом перечне всегда true.
4. Предоставление цены для конкретной акции за произвольный период в прошлом.  
   Произвольный период и шаг подаются как входные данные, шаг – минута\5минут\15 минут\час\день.  
   Предоставляется список время-цена за запрашиваемый период.
5. Получение информации об акции по ее названию.  
   Требование, существующее для выполнения запросов информации по конкретной акции – либо получение информации (цена\валюта\статус покупки\продажи\лот) на момент запроса, т. е. актуальной, либо получение информации по конкретной акции за произвольный период в прошлом.

**ПРИЛОЖЕНИЕ 4**

# СРАВНЕНИЕ СУЩЕСТВУЮЩИХ API И ОБОСНОВАНИЕ ВЫБОРА

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | [Finage.co.uk](http://finage.co.uk/)[15] | [Alpha Vantage API](https://www.alphavantage.co/documentation/) [17] | [EXANTE](https://exante.eu/ru/technology/)[18] | [Yahoo Finance API](https://www.yahoofinanceapi.com)[16] | Tinkoff API[12] |
| Бесплатная подписка | + | + | + | + | + |
| Верификация данных пользователя (удостоверения личности) для использования сервиса | - | - | + | - | + |
| Лимиты на запрос (обращения к API) | + | \_ | \_ | + | - |
| Подробная документация и примеры использования | + | + | \_ | \_ | + |
| Возможность получения текущих цен акций | + | \_ | + | + | + |
| Возможность получения данных за прошедший период | + | + | + | \_ | + |
| Возможность использования для реальной торговли на бирже (личное пользование) | - | - | + | - | + |

Цветом в табличке обозначена польза для проекта. Зеленый – подходит, красный – не подходит. Насыщенный красный – категорически не подходит. По цветовой схеме видно, что под все предъявляемые требования идеально подошел Tinkoff API.   
  
Следует отметить, что этот выбор не лишен недостатков: для его использования необходим открытый в банке инвестиционный счет, а публичный сервис на основе TINKOFF INVEST API сделать невозможно, т. к. это противоречит установленным условиям использования (TINKOFF INVEST API предоставляется только клиентам Тинькофф без права на ретрансляцию данных).

# ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего листов (страниц в докум.) | № документа | Входящий № сопроводительного докум. и дата | Подп. | Дата |
| Изм. | Измененных | Замененных | Новых | Аннулированных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |