Министерство образования и науки РФ

ФГАОУ ВО «Уральский федеральный университет   
имени первого Президента России Б.Н. Ельцина»

Институт новых материалов и металлургии

Кафедра «Теплофизика и информатика в металлургии»

Оценка работы: \_\_\_\_\_\_\_\_\_\_\_\_\_

Члены комиссии:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись расшифровка подписи

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подпись расшифровка подписи

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2024 г.

**“Приложение для расчета инвестиционных доходов на основе данных об акциях, облигациях и других финансовых**

**инструментах”**

**«Введение в технологию разработки программного обеспечения»**

Направление 09.03.02 «Информационные системы и технологии»   
(уровень бакалавриата)

Образовательная программа 09.03.02/03.01   
«Информационные системы и технологии в металлургии»

Руководитель

ст. преподаватель, к.т.н. А.С. Истомин

должность, звание подпись расшифровка подписи

Нормоконтролер

ст. преподаватель, к.т.н. А.С. Истомин

должность, звание подпись расшифровка подписи

Студенты:

НМТ-223901 Д.С. Назина

номер группы подпись расшифровка подписи

Н.А. Рыжков

подпись расшифровка подписи

Д.А. Федотова

подпись расшифровка подписи

Екатеринбург

2024

**РЕФЕРАТ**

Пояснительная записка изложена на 23 листах и содержит 14 рисунков и 1 приложение.

РАСЧЕТ ИНВЕСТИЦИОННЫХ ДОХОДОВ, ПРОГРАММИРОВАНИЕ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, АРХИТЕКТУРА, ИНТЕРФЕЙС, VISUAL STUDIO C#, ИНСТАЛЛЯЦИЯ.

Проект по модулю посвящен разработке программного обеспечения для расчета инвестиционных доходов на основе данных об акциях, облигациях и других финансовых инструментах.

Отражены основные этапы разработки программного обеспечения: постановка задачи, реализация тестового варианта расчета в электронных таблицах Microsoft Excel; проектирование и реализация программного средства – математической библиотеки и пользовательского интерфейса; разработка системы автоматизированного тестирования математической библиотеки; создание справочной документации; подготовка дистрибутива. Размещение исходного программного кода выполнено в системе удаленного контроля версий GitHub.

Основными функциями программного обеспечения является учет приобретенных инвестиционных инструментов (акций, валют, драгоценных металлов), их стоимости и доходности; расчет фиксированной прибыли, ожидаемой прибыли и доходности портфеля; визуализация данных об активах: доля активов в портфеле, доля типов продуктов, динамика цен активов); экспорт отчетов (PDF, Excel, Word).

Основные конечные пользователи программного обеспечения – инвесторы, финансовые аналитики, управляющие активами или любые лица, интересующиеся управлением инвестициями и оценкой их доходности.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 5](#_Toc169126010)

[1 ПОСТАНОВКА ЗАДАЧИ, ПРОВЕРКА КОРРЕКТНОСТИ АЛГОРИТМА РАСЧЕТА И ОРГАНИЗАЦИЯ РАБОТЫ НАД ПРОЕКТОМ В СИСТЕМЕ YOUTRACK 6](#_Toc169126011)

[1.1 Физическая постановка задачи 6](#_Toc169126012)

[1.2 Математическая модель задачи 6](#_Toc169126013)

[1.3 Создание тестового варианта расчета в электронных таблицах Microsoft Excel 7](#_Toc169126014)

[1.4 Постановки задач для выполнения отдельных этапов проекта в системе YouTrack 8](#_Toc169126015)

[В системе YouTrack были поставлены задачи для выполнения проекта поэтапно. Всего 5 задач, они представлены на рисунке 4. 8](#_Toc169126016)

[2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 9](#_Toc169126017)

[2.1 Разработка архитектуры системы 9](#_Toc169126018)

[2.2 Разработка блок-схемы работы пользователя с программой 9](#_Toc169126019)

[2.3 Создание программного обеспечения в системе управления версиями Gitlab 10](#_Toc169126020)

[2.4 Разработка математической библиотеки 11](#_Toc169126021)

[2.5 Реализация пользовательского интерфейса 11](#_Toc169126022)

[2.6 Обработка исключительных ситуаций 15](#_Toc169126023)

[2.7 Создание справочной помощи 16](#_Toc169126024)

[2.8 Создание дистрибутива 17](#_Toc169126025)

[3 РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ МАТЕМАТИЧЕСКОЙ БИБЛИОТЕКИ 18](#_Toc169126026)

[3.1 Проектирование системы 18](#_Toc169126027)

[3.2 Реализация системы 19](#_Toc169126028)

[4 ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 20](#_Toc169126029)

[4.1 Установка и настройка программного средства 20](#_Toc169126030)

[4.2 Функциональные возможности программного продукта 20](#_Toc169126031)

[4.3 Технология выполнения расчетов 20](#_Toc169126032)

[ЗАКЛЮЧЕНИЕ 21](#_Toc169126033)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 22](#_Toc169126034)

[ПРИЛОЖЕНИЕ 23](#_Toc169126035)

# ВВЕДЕНИЕ

Инвестиции имеют огромное значение в современном мире, поскольку позволяет людям и компаниям сохранять и увеличивать свой капитал, превратив его в активы, и создавать долгосрочные финансовые планы. Стабильное увеличение затрат и инфляция делают хранение денег в наличной форме менее эффективным способом управления финансами.

Более того, современные технологии и доступность информации сделали инвестирование более доступным для широкого круга людей. Онлайн-платформы, роботизированное инвестирование и приложения для управления инвестициями предоставляют индивидуальным инвесторам возможность участвовать в рынке наравне с профессиональными инвесторами.

Важность инвестиций также проявляется в их роли в поддержке экономического развития. Инвестирование в стартапы и инновационные проекты способствует росту новых отраслей, созданию рабочих мест и повышению производительности. Кроме того, инвестиции в инфраструктуру, образование и здравоохранение имеют долгосрочное воздействие на экономику и общество в целом.

Поэтому основываясь на выше перечисленные факты было разработано приложение для инвестиционнных доходов на основе данных об акциях, облигациях и других финансовых инструментах разработано для облегчения управления инвестициями и максимизации доходности. Оно помогает в учете приобретенных инвестиционных инструментов, расчете фиксированной и ожидаемой прибыли, а также доходности портфеля. Визуализация данных об активах и возможность экспорта отчетов обеспечивают удобство анализа и принятия решений.

# 1 ПОСТАНОВКА ЗАДАЧИ, ПРОВЕРКА КОРРЕКТНОСТИ АЛГОРИТМА РАСЧЕТА И ОРГАНИЗАЦИЯ РАБОТЫ НАД ПРОЕКТОМ В СИСТЕМЕ YOUTRACK

## 1.1 Физическая постановка задачи

Физическая постановка задачи включает в себя следующие аспекты:

* Описание функциональных требований: приложение должно предоставлять возможность ввести учет приобретенных инвестиционных инструментов, их стоимости и доходности; ввести учет фиксированной и ожидаемой прибыли, потенциальной прибыли от продаж активов и доходности портфеля; визуализация данных об активах: доля активов в портфеле, доля типов продуктов, динамика цен активов; экспорт отчетов (PDF, Excel, Word)
* Архитектура приложения: разработка приложения должна быть основана на модульной архитектуре для обеспечения легкости расширения функциональности и поддержки.
* Выбор инструментов разработки: необходимо выбрать фреймворки и инструменты разработки, которые обеспечат эффективное создание и поддержку приложения.

## 1.2 Математическая модель задачи

Математическая модель задачи состоит из нескольких формул:

1. Расчет доходности
2. Расчет потенциальной прибыли

1. Расчет фиксированной прибыли

## 1.3 Создание тестового варианта расчета в электронных таблицах Microsoft Excel

В Microsoft Excel создан Excel-файл с тестовыми данными. Excel-файл состоит из 2-х таблиц.

На рисунке 1 представлена таблица с историей операций в портфеле.

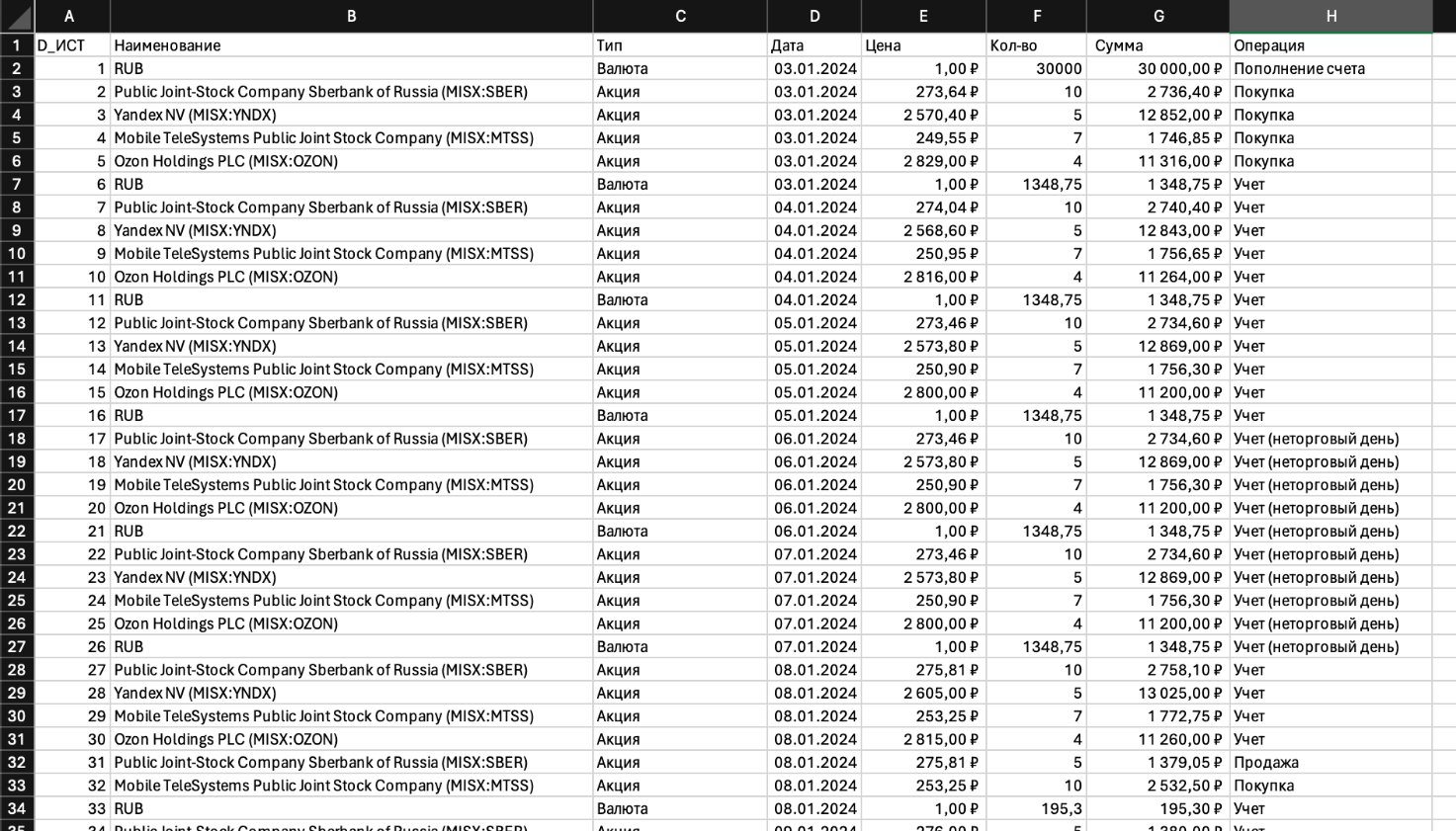


Рис. 1 – таблица «История»

На рисунке 2 и 3 представлена таблица с портфелем, который содержит модель расчета ожидаемой и фиксированной прибыли, а также доходности по каждому инструменту и в общей сумме.

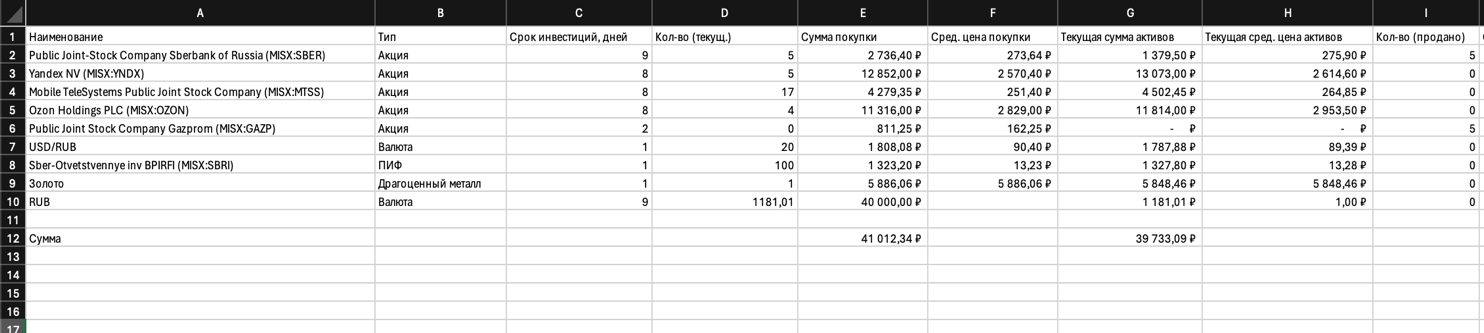


Рис. 2 – таблица «Портфель»

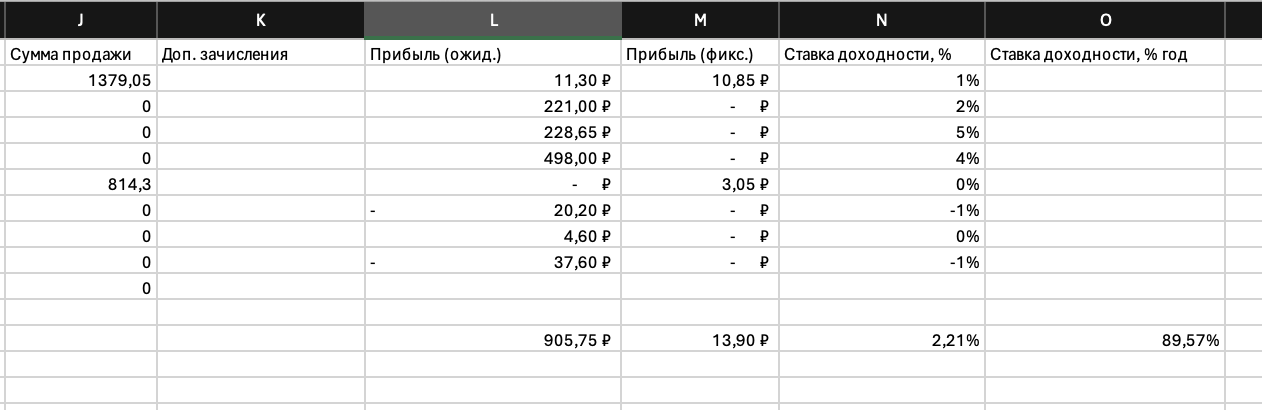


Рис. 3 – таблица «Портфель»

## 1.4 Постановки задач для выполнения отдельных этапов проекта в системе YouTrack

## В системе YouTrack были поставлены задачи для выполнения проекта поэтапно. Всего 5 задач, они представлены на рисунке 4.

Рис. 4 – список задач

# 2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

## 2.1 Разработка архитектуры системы

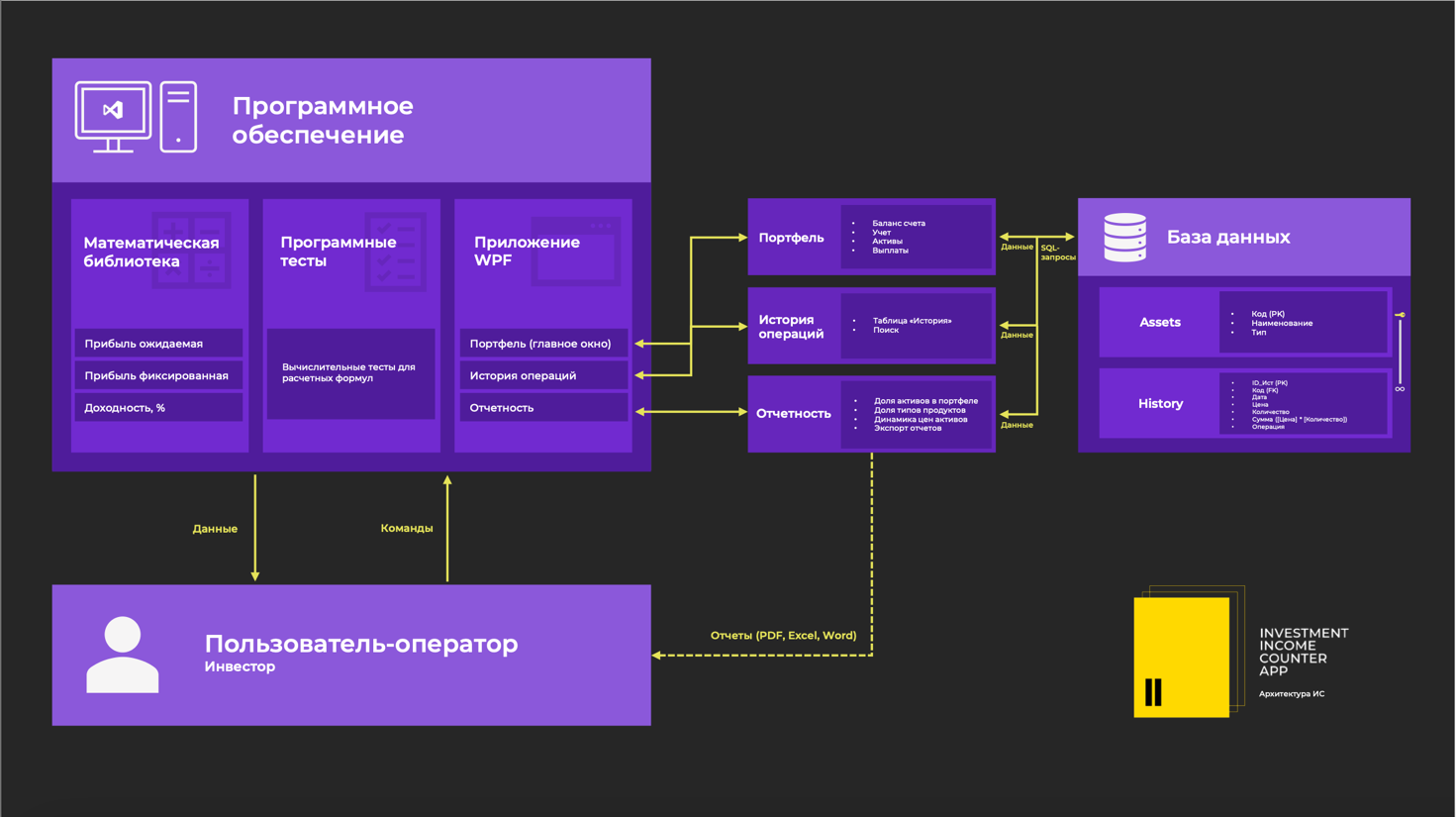
На рисунке 5 представлена высокоуровневая схема архитектуры системы.

Рис. 5 – Архитектура системы

## 2.2 Разработка блок-схемы работы пользователя с программой

На рисунке 6 представлена блок-схема пользовательского интерфейса.

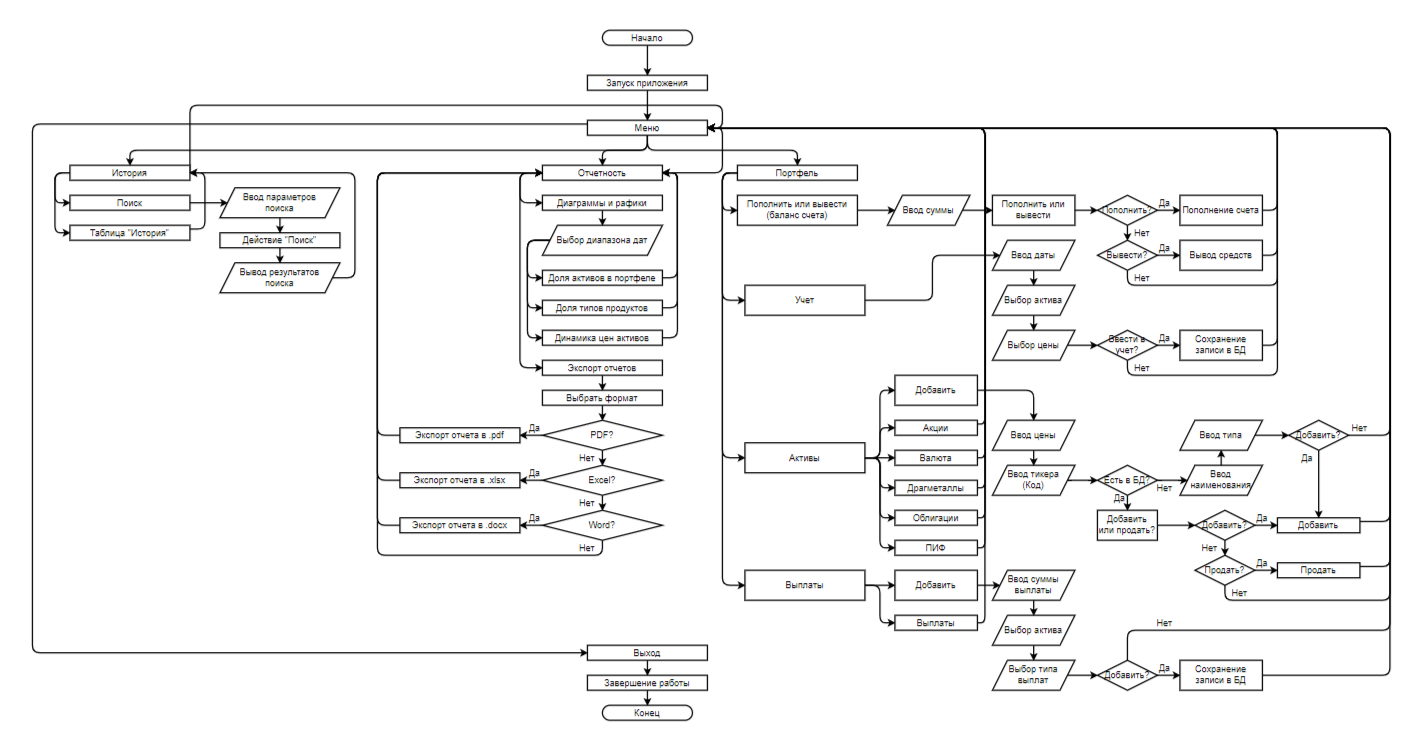


Рис. 6 – блок-схема работы пользователя с программой

## 2.3 Создание программного обеспечения в системе управления версиями Gitlab

Разработка программного обеспечения происходила в системе контроля версий GitHub, благодаря которому мы могли эффективно управлять изменениями, отслеживать историю, откатывать изменения и сотрудничать над проектом.

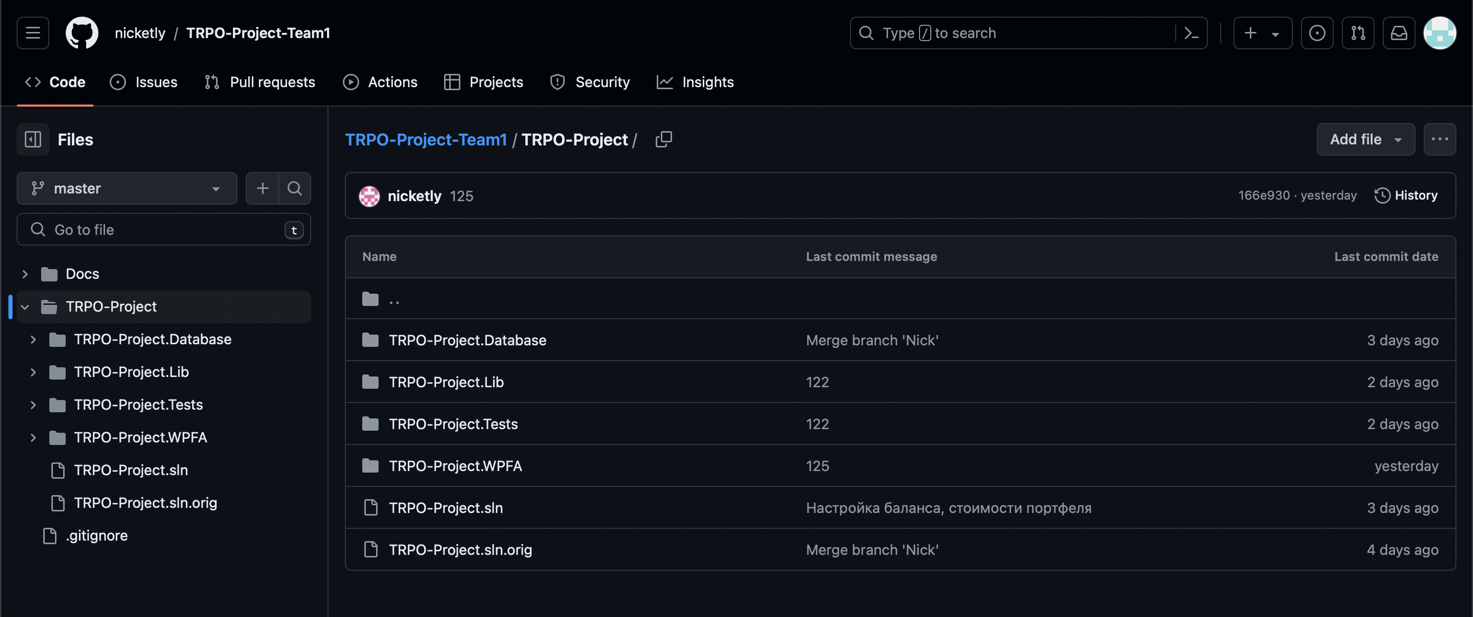


Рис. 7 – система GitHub

## 2.4 Разработка математической библиотеки

Также была разработана математическая библиотека, которая состоит из трех формул:

* Расчет потенциальной прибыли
* Расчет фиксированной прибыли
* Расчет доходности

Математическая библиотека представлена на рисунке 8.

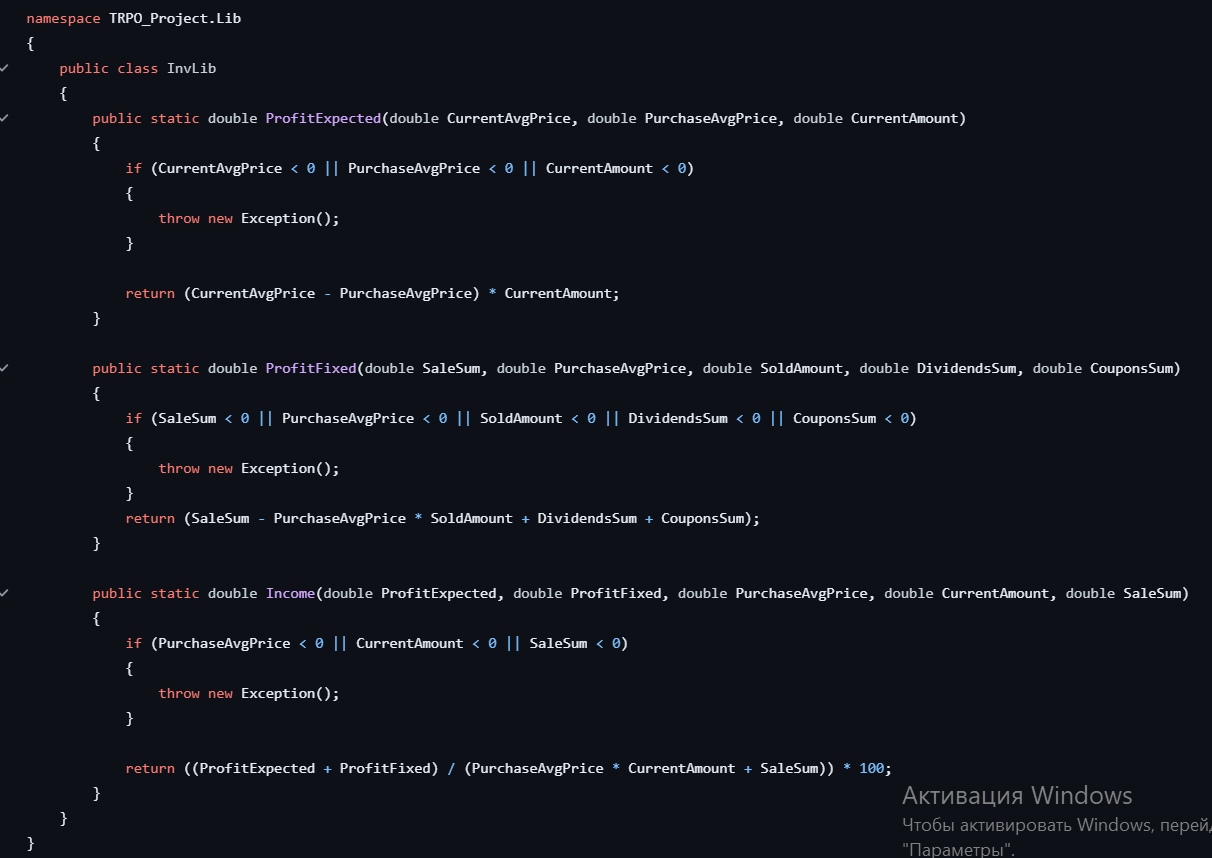


Рис. 8– математическая библиотека

## 2.5 Реализация пользовательского интерфейса

С помощью технологий Windows Presentation Foundation (WPF) было разработано приложение в среде разработки Visual Studio с использованием языка программирования С#.

В приложении было разработано 5 окон, представляющих различные функциональные блоки и возможности программы:

* Главное окно «Портфель»

На рисунке 9 представлен рисунок с главным окном, в котором возможно просмотреть инвестиционные инструменты (акции, валюты, драгоценные металлы, облигации и ПИФ), выплаты, календарь, баланс счета и стоимость портфеля. Также есть возможность перейти в другие окна, такие как «История», «Отчетность», «Учет» и «Брокерский счет»

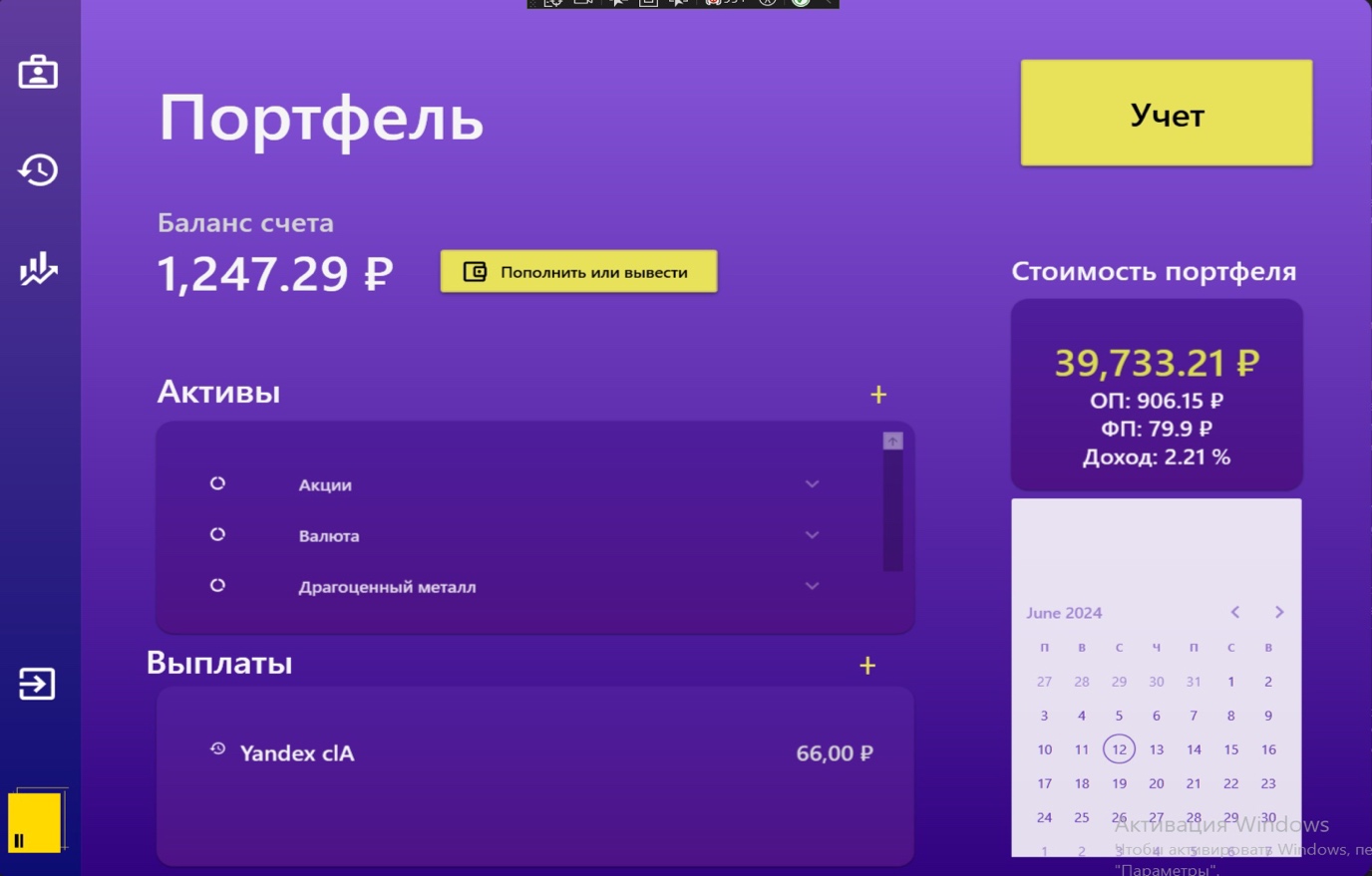


Рис. 9 – окно «Портфель»

* Окно «История»

Окно «История» представлена на рисунке 10.

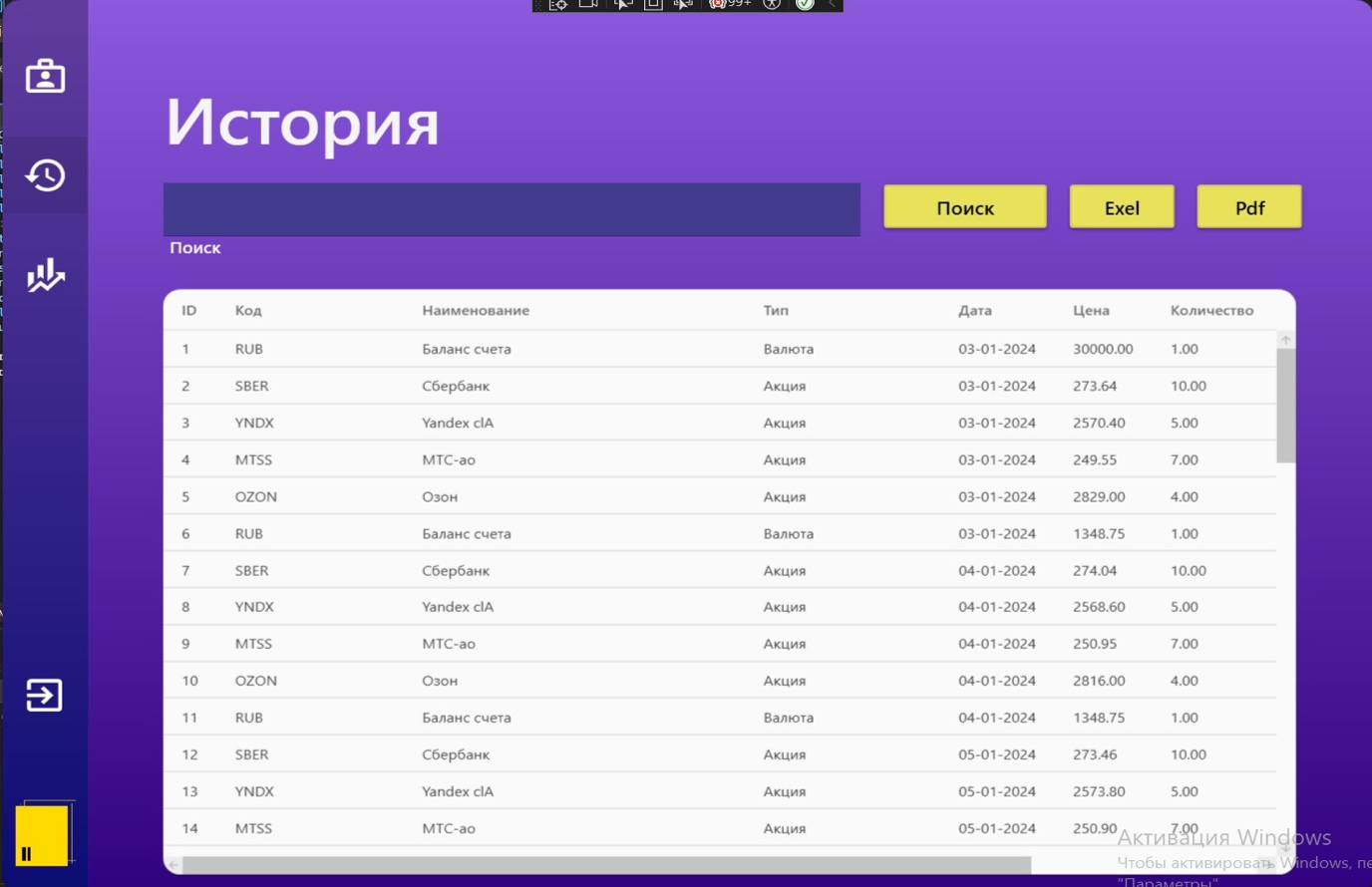


Рис. 10 – окно «История»

* Окно «Отчетность»

На рисунке 11 представлено окно с отчетностью с помощью графиков и диаграмм, и с возможностью экспортировать отчеты в любые форматы.

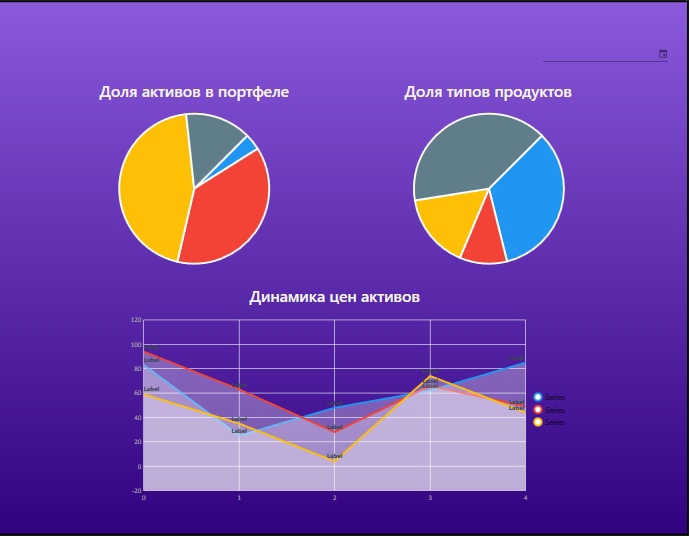


Рис. 11 – окно «Отчетность»

* Окно «Учет»

Окно с учетом представлен на рисунке 12.

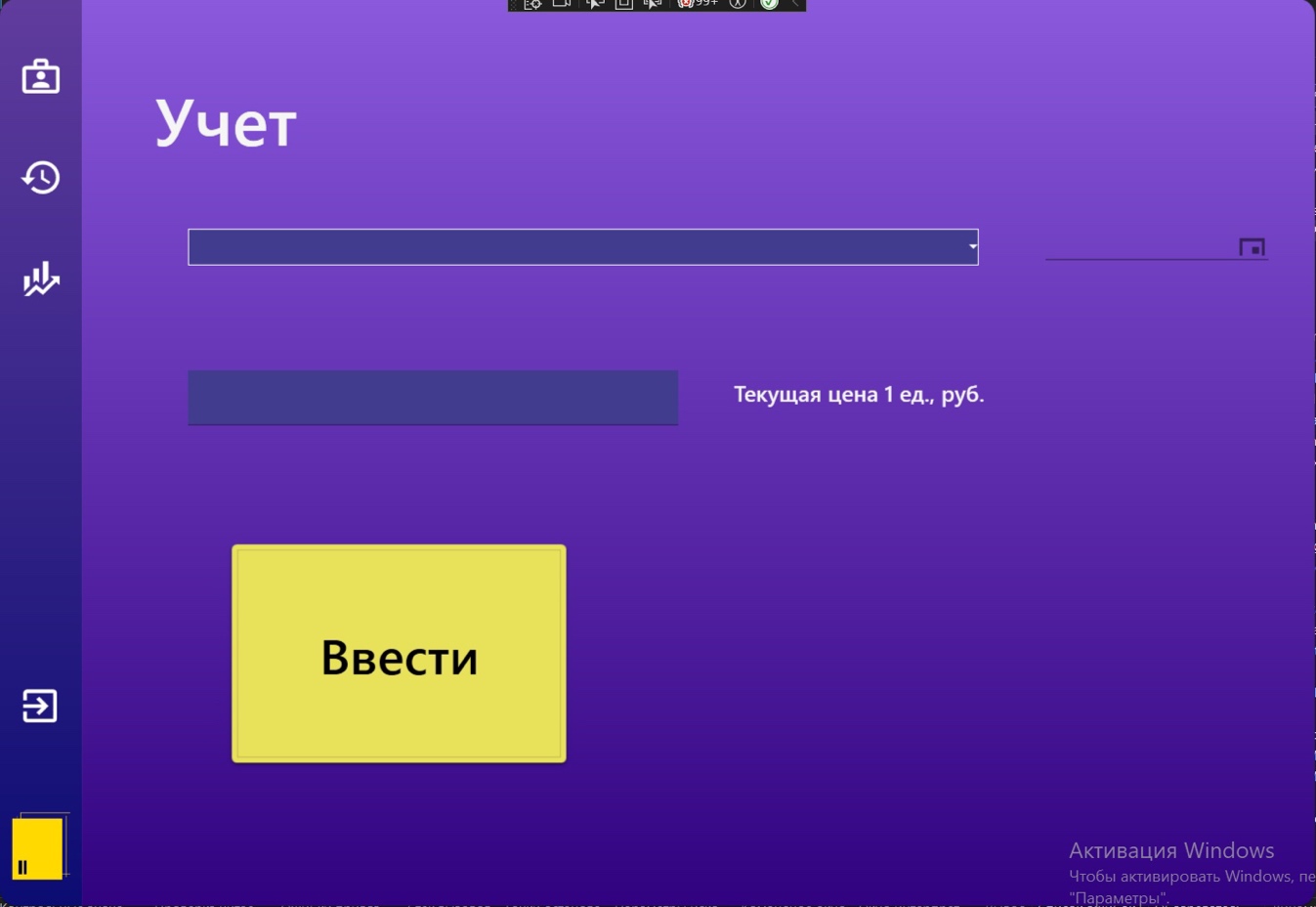


Рис. 12 – окно «Учет»

* Окно «Брокерский счет»

На рисунке 13 представлен рисунок с окном брокерского счета, в котором возможно пополнить или вывести определенную сумму баланса.

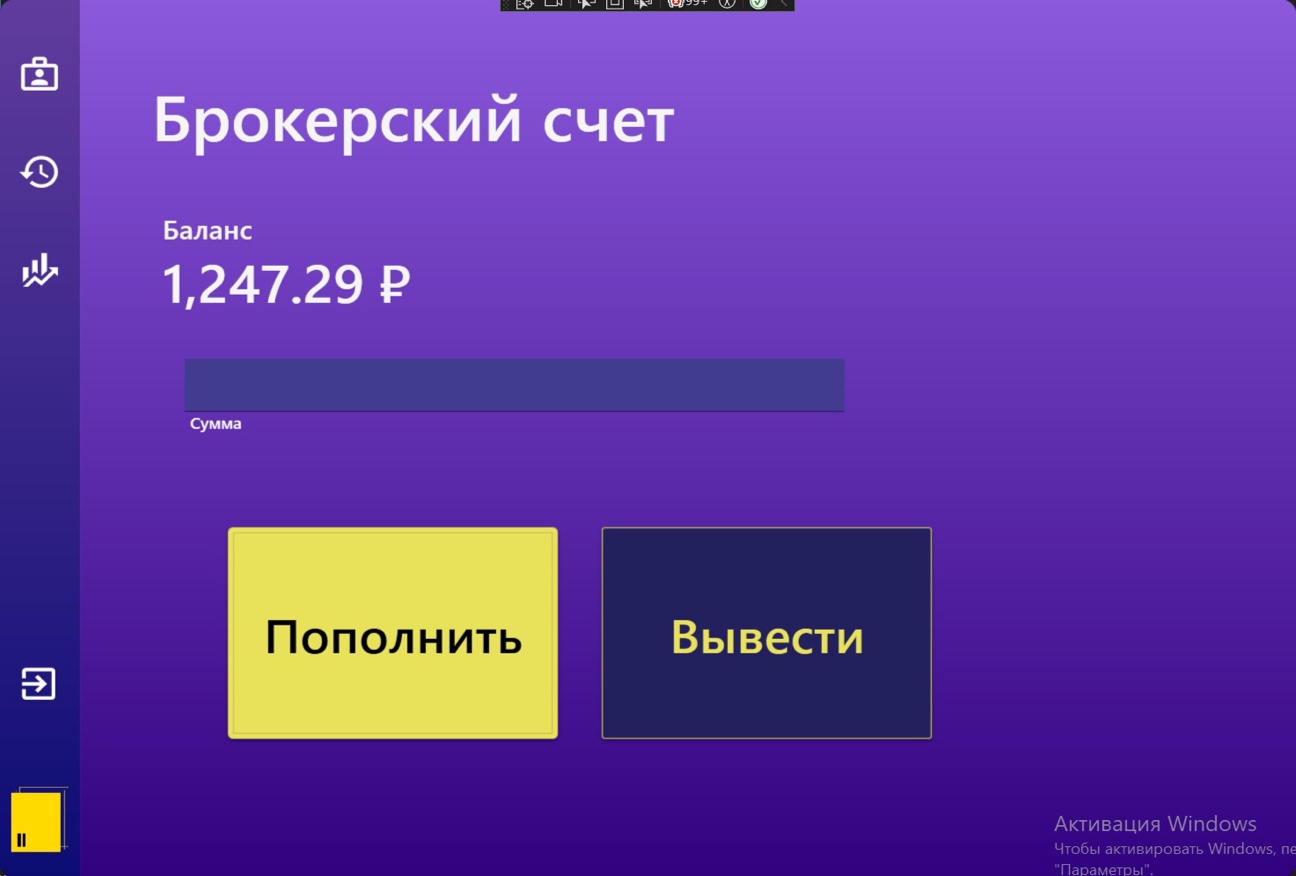


Рис. 13 – окно «Брокерский счет»

## 2.6 Обработка исключительных ситуаций

В приложении был разработан механизм обработки исключительных ситуаций, что позволяет эффективно управлять потенциальными проблемами и обеспечивать бесперебойную работу программы.

Пример такой обработки является обработка ситуации, когда вводятся неправильные данные для вставки записи в таблицу истории по операции «Вывод средств.

На рисунке 14 представлен код обработки этой исключительной ситуации.

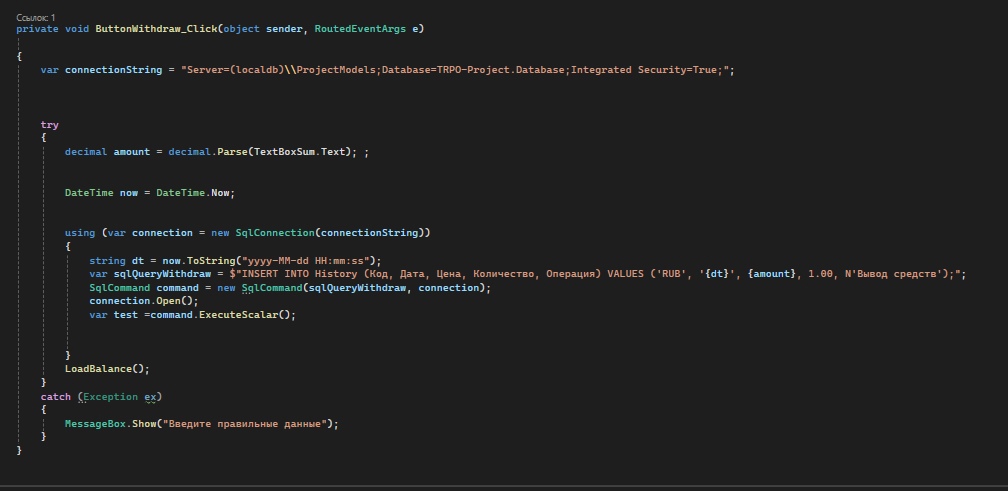


Рис. 14 – обработка исключительной ситуации.

## 2.7 Создание справочной помощи

Инструкция по использованию приложения

1. Запуск приложения

• Начало работы: Запустите приложение.

1. . Главное меню

• Меню: После запуска приложения вы попадаете в главное меню, где доступны следующие основные разделы:

- История

- Отчетность

- Портфель

1. Раздел "История"

• Просмотр истории: В разделе "История" отображается таблица "История". Здесь вы можете просмотреть ранее выполненные действия и операции.

• Поиск информации:

- Введите параметры поиска.

- Нажмите кнопку "Поиск".

- Результаты поиска будут выведены на экран.

1. Раздел "Отчетность"

• Работа с отчетами:

• Диаграммы и графики: Просмотр различных диаграмм и графиков.

• Выбор диапазона дат: Установите временной интервал для отчета.

• Доля активов в портфеле: Просмотр распределения активов.

• Доля типов продуктов: Анализ типов продуктов в портфеле.

• Динамика цен активов: Просмотр изменения цен активов во времени.

• Экспорт отчетов: Выберите формат для экспорта отчета (PDF, Excel, Word).

• Экспорт в PDF: Если выбрали PDF, отчет будет экспортирован в этот формат. • Экспорт в Excel: Если выбрали Excel, отчет будет экспортирован в формат .xlsx.

• Экспорт в Word: Если выбрали Word, отчет будет экспортирован в формат .docx.

1. Раздел "Портфель"

• Управление портфелем:

- Пополнение или вывод средств:

- Ввод суммы: Введите сумму для пополнения или вывода.

- Выбор действия: Выберите "Пополнить" или "Вывести".

- Пополнение счета: Введите данные актива и сохраните запись в БД.

- Вывод средств: Введите данные актива и сохраните запись в БД.

- Учет активов:

- Добавление актива: Выберите тип актива (Акции, Валюта, Драгметаллы, Облигации, ПИФы).

- Ввод данных: Введите цену, тикер, наименование, и сохраните запись в БД.

- Продажа актива: Введите данные продажи, выберите актив и сохраните запись в БД.

- Выплаты:

- Добавление выплаты: Введите сумму выплаты, выберите актив и тип выплаты, затем сохраните запись в БД.

6. Выход из приложения

• Выход: Вы можете выйти из приложения через главное меню, нажав на соответствующую кнопку, затем приложение завершит работу

## 2.8 Создание дистрибутива

Для создания дистрибутива использовалось расширение Microsoft Visual Studio Installer Projects. Был создан проект установщика TRPO-Project.Setup, в качестве выходного элемента выбраны файлы проекта с WPF-приложением. Также были настроены ярлыки приложения.

# 3 РАЗРАБОТКА СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ МАТЕМАТИЧЕСКОЙ БИБЛИОТЕКИ

## 3.1 Проектирование системы

Для автоматизированного тестирования математической библиотеки использовался фреймворк NUnit, который позволяет создавать и выполнять тесты. В этом разделе представлено проектирование системы тестирования, включая примеры кода для тестов и самой библиотеки.

**Проектирование системы тестирования**

Система тестирования состоит из следующих компонентов:

* Математическая библиотека TRPO\_Project.Lib, содержащая основные вычислительные функции.
* Набор тестов TRPO\_Project.Tests, проверяющий корректность работы функций библиотеки.

**Используемая библиотека**

Библиотека TRPO\_Project.Lib содержит три основные функции: ProfitExpected, ProfitFixed и Income. Каждая функция выполняет определенные вычисления и проверяет входные параметры на корректность.

**Описание функций и тестов**

* ProfitExpected(double currentAvgPrice, double purchaseAvgPrice, double currentAmount): Вычисляет ожидаемую прибыль. Принимает текущую среднюю цену, среднюю цену покупки и текущее количество. Если любой из параметров отрицательный, выбрасывается исключение.
* ProfitFixed(double saleSum, double purchaseAvgPrice, double soldAmount, double dividendsSum, double couponsSum): Вычисляет фиксированную прибыль. Принимает сумму продажи, среднюю цену покупки, проданное количество, сумму дивидендов и сумму купонов. Если любой из параметров отрицательный, выбрасывается исключение.
* Income(double profitExpected, double profitFixed, double purchaseAvgPrice, double currentAmount, double saleSum): Вычисляет доходность. Принимает ожидаемую прибыль, фиксированную прибыль, среднюю цену покупки, текущее количество и сумму продажи.

Тесты проверяют корректность каждой из функций, используя заранее определенные значения входных параметров и ожидаемые результаты. Таким образом, система автоматизированного тестирования обеспечивает надежную проверку математической библиотеки на корректность и устойчивость к некорректным данным

## 3.2 Реализация системы

Для реализации системы автоматизированного тестирования была использована библиотека NUnit. В данном разделе описана реализация системы, включая настройку проекта и написание тестов.

**Шаг 1: Настройка проекта**

1. Создание проекта библиотеки:
   * Был создан новый проект библиотеки классов.
   * Проекту было дано имя TRPO\_Project.Lib.
   * В проект был добавлен файл InvLib.cs с функциями:
2. Создание проекта для тестов:
   * Был создан новый проект тестов.
   * Проекту было дано имя TRPO\_Project.Tests.
   * Была добавлена ссылка на проект библиотеки TRPO\_Project.Lib.
   * В проект был добавлен файл с тестами для функций библиотеки.

**Шаг 2: Реализация тестов**

Тесты были написаны для проверки различных аспектов функций библиотеки, включая корректность вычислений и обработку некорректных данных. Ниже представлен код тестов.

**Шаг 3: Запуск тестов**

Запуск тестов в IDE:

* + Было открыто окно тестов в IDE.
  + Все тесты были запущены и прошли успешно.

# 4 ОПИСАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

## 4.1 Установка и настройка программного средства

1. Начало установки:
   * В начальном окне установщика нажмите кнопку «Далее», чтобы начать установку.
2. Выбор папки для установки:
   * Выберите папку для установки ПО.
   * Выберите для кого будет установлено ПО: для текущего пользователя или для всех пользователей компьютера.
3. Подтверждение установки:
   * Для подтверждения установки нажмите кнопку «Далее».
4. Завершение установки:
   * Установка завершена, приложение готово к использованию.

## 4.2 Функциональные возможности программного продукта

Приложение имеет следующую функциональность:

1. Учёт стоимости активов в портфеле;
2. Учет выплат дивидендов по акциям и купонов по облигациям;
3. Расчет ожидаемой и фиксируемой прибыли;
4. Расчет доходности портфеля;
5. Просмотр истории операций, поиск по записям и сортировка;
6. Просмотр отчетности в диаграммах и графиках;
7. Экспорт отчетов в форматах PDF и Excel;
8. Модель баланса брокерского счета: пополнение, вывод средств, начисление выплат и средств от продажи активов.

## 4.3 Технология выполнения расчетов

Расчеты выполняются, учитывая данные из истории операций в портфеле: записи добавляются в базу данных со значениями, введёнными пользователем на формах ввода. Ожидаемая прибыль (ОП), фиксированная прибыль (ФП) и доходность рассчитываются, используя функции класса InvLib математической библиотеки TRPO-Project.Lib. Рядом с каждым инвестиционным инструментом в портфеле выводятся результаты расчета ОП. На панели стоимости портфеля также отображаются ОП, ФП и доходность.

# ЗАКЛЮЧЕНИЕ

В результате выполнения данного проекта была разработана и внедрена система для управления инвестициями, которая предоставляет пользователям широкие возможности для анализа и максимизации доходности своих вложений. Приложение позволяет учитывать различные финансовые инструменты, такие как акции, облигации и другие активы, а также осуществлять расчеты фиксированной и ожидаемой прибыли и доходности портфеля.

Разработанная математическая библиотека обеспечивает точные и надежные вычисления на основе введенных данных, что подтверждается системой автоматизированного тестирования, реализованной с использованием фреймворка NUnit. В ходе работы были созданы и успешно выполнены тесты, которые проверяют корректность работы всех основных функций библиотеки.

Интерфейс пользователя, созданный с помощью технологий Windows Presentation Foundation (WPF), обеспечивает удобство и интуитивность работы с приложением. Включение таких функциональных возможностей, как визуализация данных, экспорт отчетов и управление брокерским счетом, делает приложение полезным инструментом для широкого круга пользователей.

Стоит отметить, что приложение соответствует паттерну MVVM (Model-View-ViewModel), что обеспечивает гибкость, тестируемость и расширяемость кода. Это также упрощает разделение логики представления и бизнес-логики, что способствует лучшему управлению и поддержке приложения в долгосрочной перспективе.

Процесс разработки включал в себя тщательное планирование и реализацию, начиная с постановки задачи и проектирования системы, до создания и тестирования конечного продукта. Интеграция системы контроля версий GitHub позволила эффективно управлять изменениями и сотрудничать над проектом.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кириченко, И. А. Инвестиционные процессы в цифровой экономике: монография. — М.: Экономика, 2019. — 328 с.
2. Журбин, С. Б. Финансовые технологии и управление инвестициями: учебное пособие. — М.: Юрайт, 2020. — 384 с.
3. Липсиц, И. В. Управление инвестициями: учебник для вузов. — М.: Экономистъ, 2018. — 456 с.
4. Максимова, Е. И. Автоматизация финансовых операций: современные подходы и технологии. — М.: Финансы и статистика, 2021. — 290 с.
5. Смирнова, Е. А. Современные технологии управления инвестиционными портфелями: учебное пособие. — СПб.: Питер, 2017. — 312 с.
6. Лавров В.В. Методические указания к выполнению курсовой работы для студентов бакалавриата, обучающихся по направлению «Информационные системы и технологии» и студентов магистратуры, обучающихся по программе «Информационные системы в металлургии» / В.В. Лавров, К.А. Щипанов, А.А. Бурыкин – Екатеринбург: УрФУ, 2014. – 49 с.
7. Лошкарев Н.Б. Указания к оформлению дипломных и курсовых проектов и работ: методические указания / Н.Б. Лошкарев, А.Н. Лошкарев, Л.А. Зайнуллин. – Екатеринбург: УГТУ–УПИ, 2007. – 49 с.

# ПРИЛОЖЕНИЕ

Фрагменты листинга программного обеспечения

Математическая библиотека:

namespace TRPO\_Project.Lib

{

public class InvLib

{

public double ProfitExpected(double CurrentAvgPrice, double PurchaseAvgPrice, double CurrentAmount)

{

if (CurrentAvgPrice < 0 || PurchaseAvgPrice < 0 || CurrentAmount < 0)

{

throw new Exception();

}

return (CurrentAvgPrice - PurchaseAvgPrice) \* CurrentAmount;

}

public double ProfitFixed(double SaleSum, double PurchaseAvgPrice, double SoldAmount, double DividendsSum, double CouponsSum)

{

if (SaleSum < 0 || PurchaseAvgPrice < 0 || SoldAmount < 0 || DividendsSum < 0 || CouponsSum < 0)

{

throw new Exception();

}

return (SaleSum - PurchaseAvgPrice \* SoldAmount + DividendsSum + CouponsSum);

}

public double Income(double ProfitExpected, double ProfitFixed, double PurchaseAvgPrice, double CurrentAmount, double SaleSum)

{

if (PurchaseAvgPrice < 0 || CurrentAmount < 0 || SaleSum < 0)

{

throw new Exception();

}

return ((ProfitExpected + ProfitFixed) / (PurchaseAvgPrice \* CurrentAmount + SaleSum)) \* 100;

}

}

}

Тесты математической библиотеки:

using TRPO\_Project.Lib;

namespace TRPO\_Project.Tests

{

public class Tests

{

[SetUp]

public void Setup()

{

}

[Test]

public void ComputationalTest\_ProfitExpected() //Вычислительный тест

{

const double currentAvgPrice = 400;

const double purchaseAvgPrice = 200;

const double currentAmount = 200;

const double expected = 40000;

double result = new InvLib().ProfitExpected(currentAvgPrice, purchaseAvgPrice, currentAmount);

Assert.AreEqual(expected, result);

}

[Test]

public void ComputationalTest\_ProfitFixed() //Вычислительный тест

{

const double saleSum = 400;

const double purchaseAvgPrice = 10;

const double soldAmount = 10;

const double dividendsSum = 200;

const double couponsSum = 200;

const double expected = 700;

double result = new InvLib().ProfitFixed(saleSum, purchaseAvgPrice, soldAmount, dividendsSum, couponsSum);

Assert.AreEqual(expected, result);

}

[Test]

public void ComputationalTest\_Income() //Вычислительный тест

{

const double profitExpected = 510;

const double profitFixed = 510;

const double purchaseAvgPrice = 100;

const double currentAmount = 100;

const double saleSum = 200;

const double expected = 10;

double result = new InvLib().Income(profitExpected, profitFixed, purchaseAvgPrice, currentAmount, saleSum);

Assert.AreEqual(expected, result);

}

}

}