Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и кибербезопасности

Высшая школа программной инженерии

Отчёт по курсовой работе

Дисциплина: Объектно-ориентированное программирование

Выполнил студент гр. в5130904/30022 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Иванов Р. А.

(подпись)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_ Маслаков А. П.

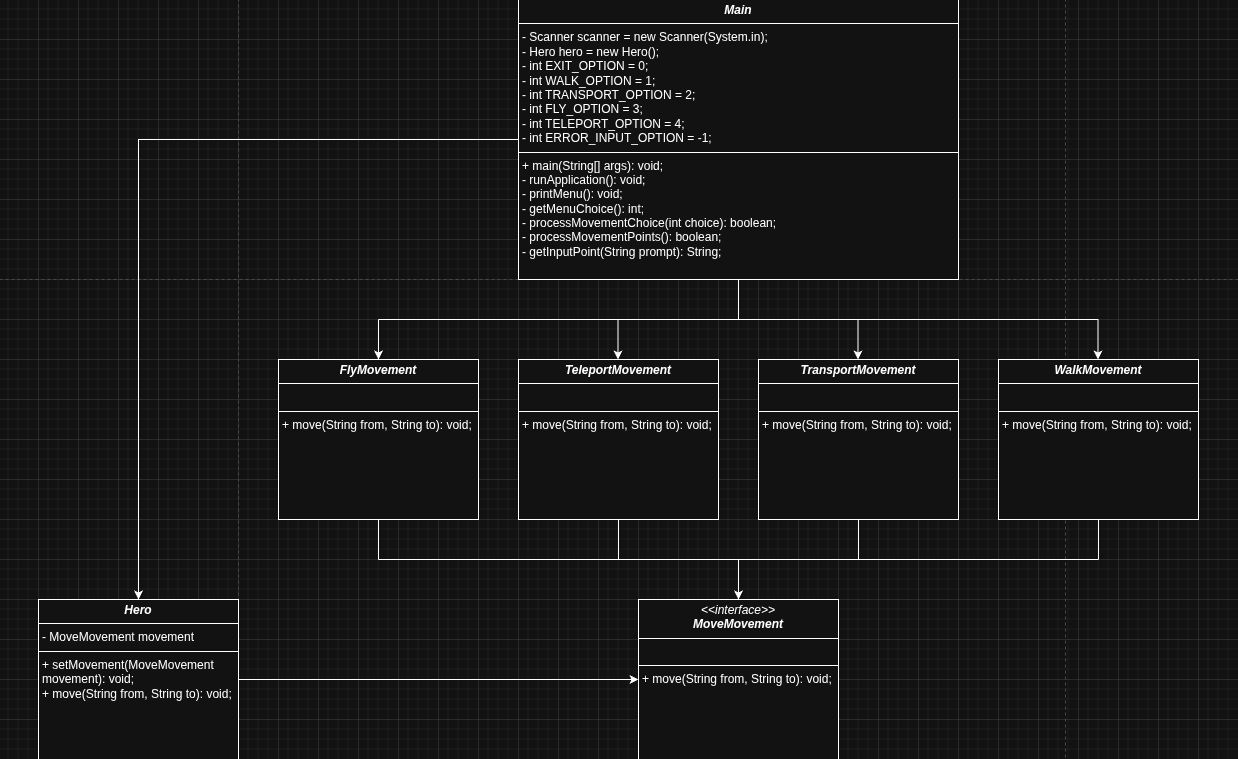
(подпись)

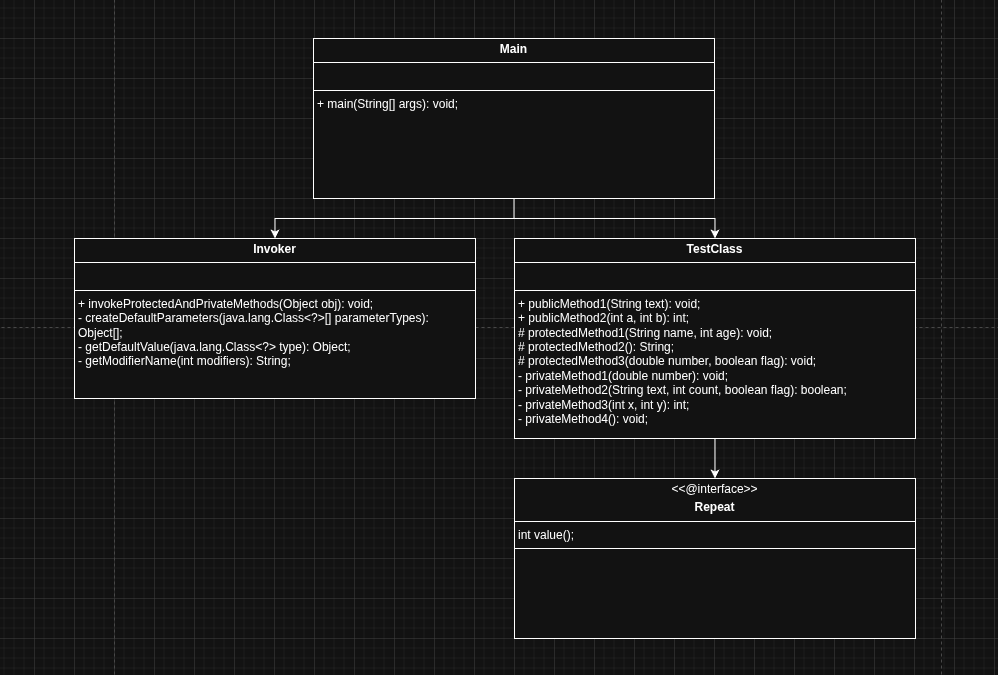
«18» октября 2025 г.

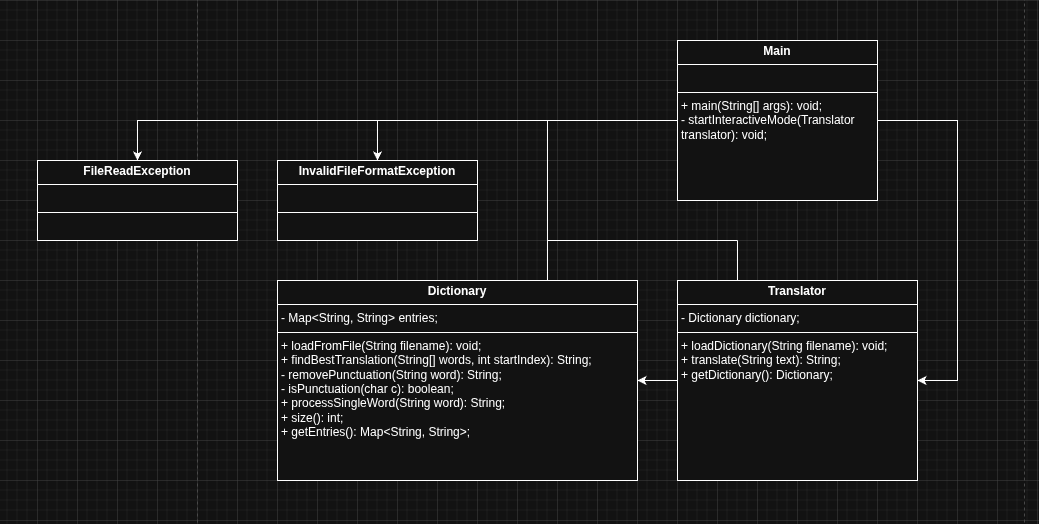
Санкт-Петербург

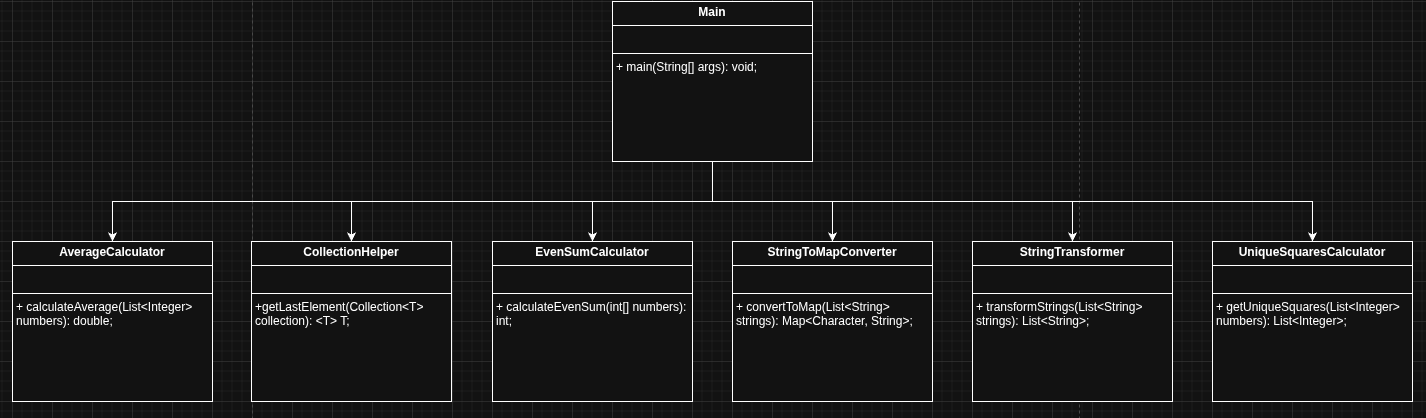
2025

**Диаграммы классов**

**Рисунок 1.** Диаграмма классов Лабораторной работы 1.

**Рисунок 2.** Диаграмма классов Лабораторной работы 2.

**Рисунок 3.** Диаграмма классов Лабораторной работы 3.

**Рисунок 4.** Диаграмма классов Лабораторной работы 4.

**Перечень выполненных работ**

**Часть 1: Создание базовой структуры приложения**

- Разработана иерархия пакетов согласно требованиям

- Создано главное меню с 5 пунктами выбора

- Реализована навигация между окнами

- Создано базовое окно для лабораторных работ

- Реализован единый интерфейс с кнопкой "Назад"

**Часть 2: Лабораторная работа #1 - "Компьютерная игра"**

- Реализован паттерн "Стратегия" для перемещения героя

- Созданы классы:

1) Hero - основной класс героя

2) MoveMovement - интерфейс стратегии

3) WalkMovement - перемещение пешком

4) TransportMovement - перемещение на транспорте

5) FlyMovement - перемещение полетом

6) TeleportMovement - телепортация

- Разработан графический интерфейс:

1. Поля ввода точек маршрута
2. Кнопки выбора способа перемещения
3. Область вывода результатов
4. Кнопка очистки лога

**Часть 3: Лабораторная работа #2 - "Аннотации и рефлексия"**

- Создана аннотация @Repeat с целочисленным параметром

- Реализован класс TestClass с методами разных модификаторов доступа

- Разработан класс Invoker для вызова методов через рефлексию

- Создан графический интерфейс с функциями:

1. Просмотр информации о классе
2. Вызов публичных методов
3. Вызов аннотированных методов
4. Вызов приватных методов
5. Вызов защищенных методов
6. Вызов всех методов

**Часть 4: Лабораторная работа #3 - "Переводчик текста"**

- Реализован класс Dictionary для работы со словарем

- Создан класс Translator для выполнения перевода

- Разработаны пользовательские исключения:

1. FileReadException - ошибки чтения файла
2. InvalidFileFormatException - неверный формат файла

- Реализован алгоритм перевода с учетом:

1. Игнорирования регистра
2. Поиска наиболее длинного соответствия
3. Сохранения пунктуации

- Создан графический интерфейс с:

1. Загрузкой словаря из файла
2. Загрузкой текста для перевода из файла
3. Ручным вводом текста
4. Отображением результатов перевода
5. Статус-баром операций

**Часть 5: Лабораторная работа #4 - "Stream API операции"**

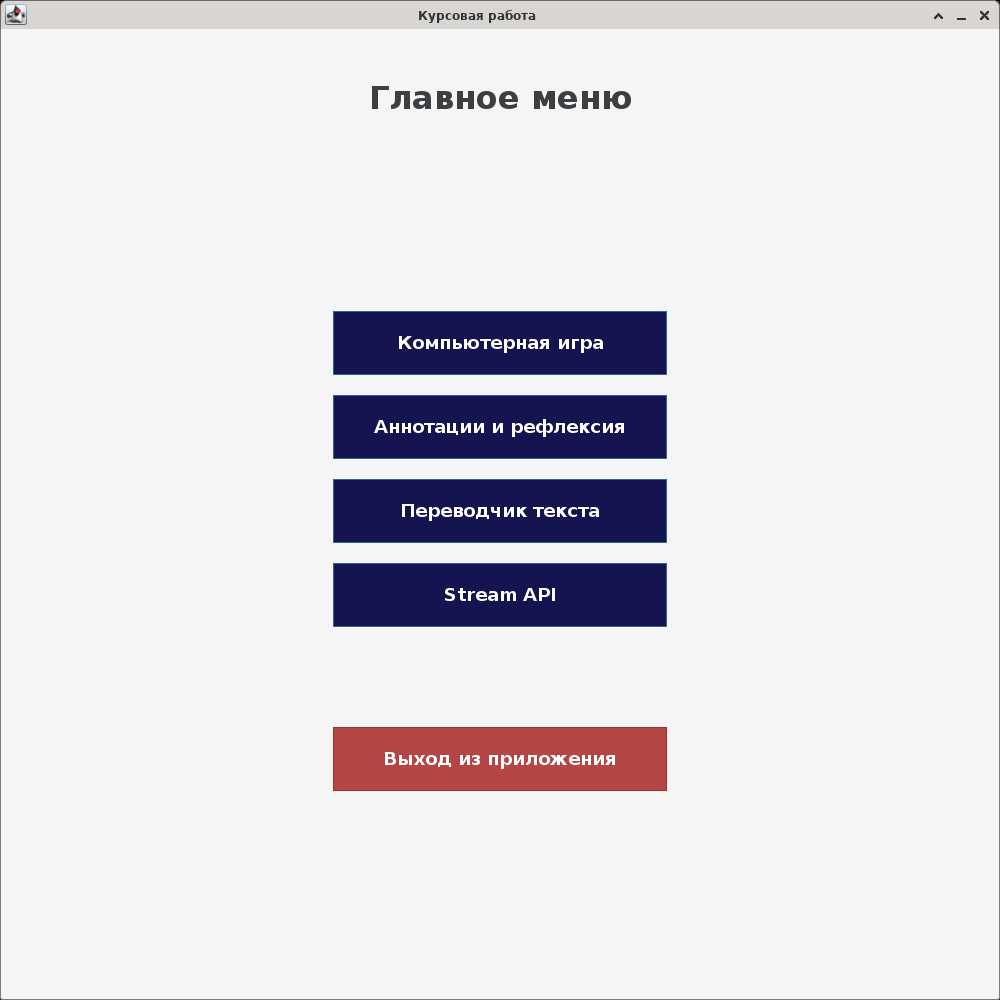
- Реализованы 6 методов с использованием Stream API:

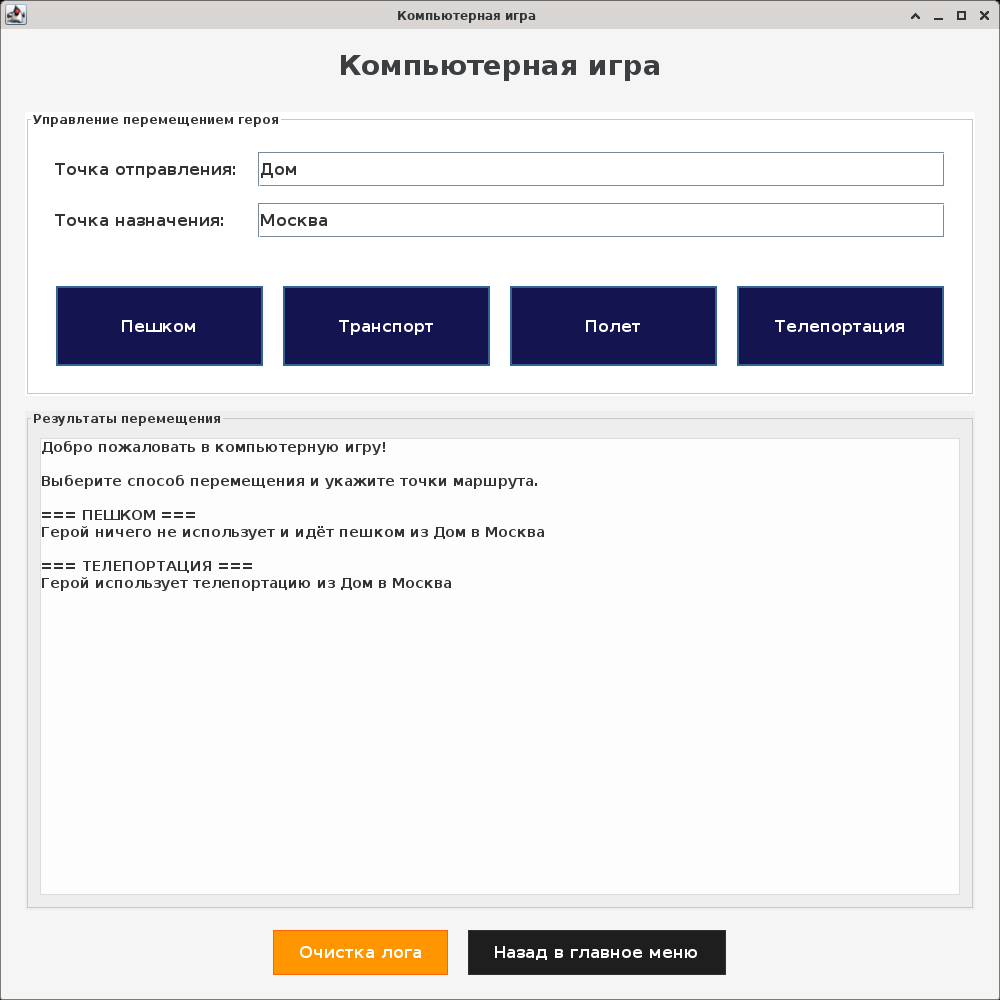
1. AverageCalculator.calculateAverage() - среднее значение
2. StringTransformer.transformStrings() - трансформация строк
3. UniqueSquaresCalculator.getUniqueSquares() - квадраты уникальных элементов
4. CollectionHelper.getLastElement() - последний элемент коллекции
5. EvenSumCalculator.calculateEvenSum() - сумма четных чисел
6. StringToMapConverter.convertToMap() - преобразование строк в Map

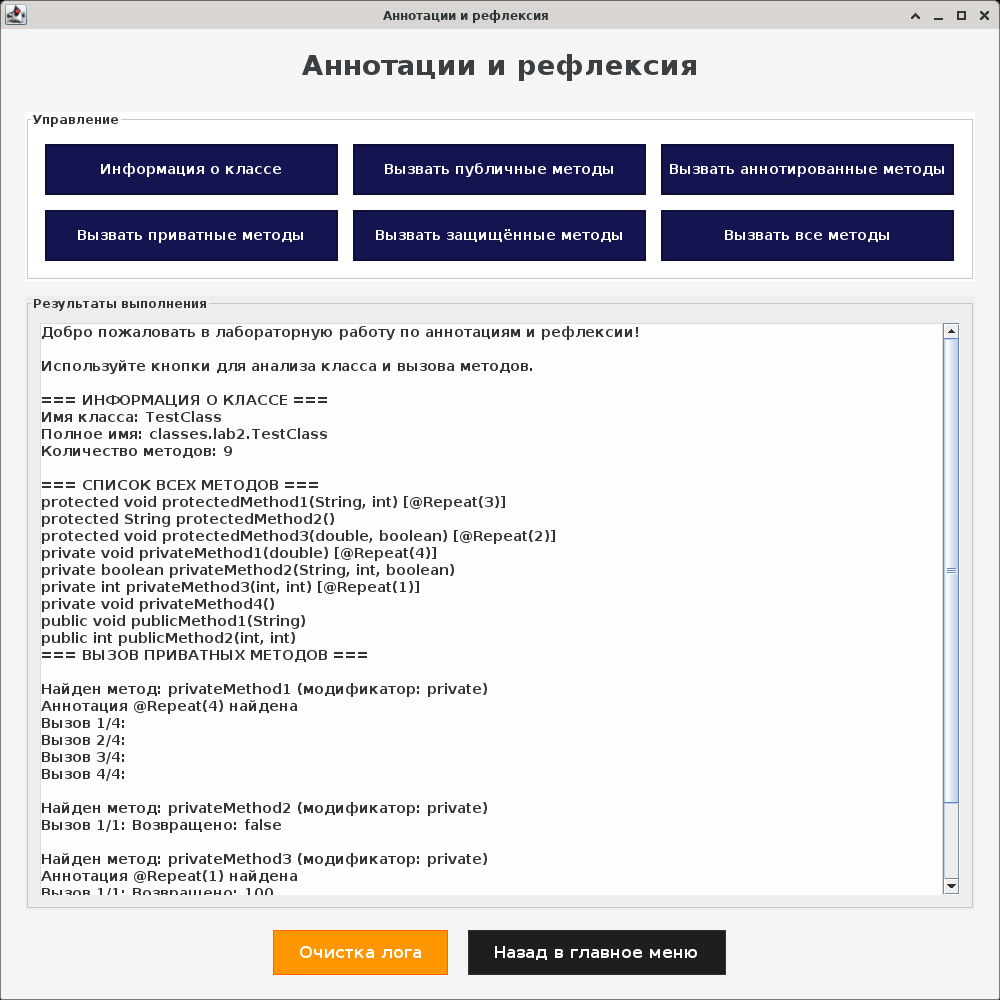
- Создан универсальный графический интерфейс:

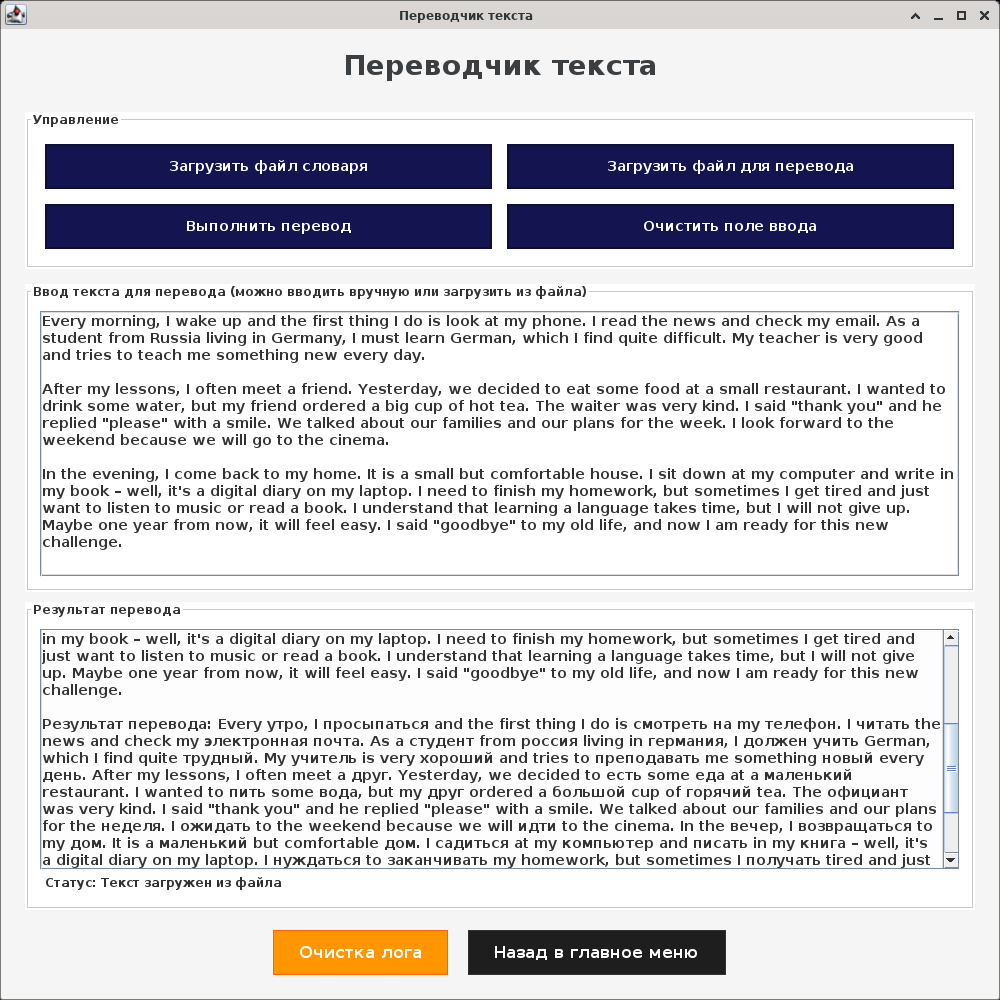
1. Выбор метода через ComboBox
2. Умный парсинг входных данных
3. Подробный вывод результатов
4. Обработка ошибок ввода

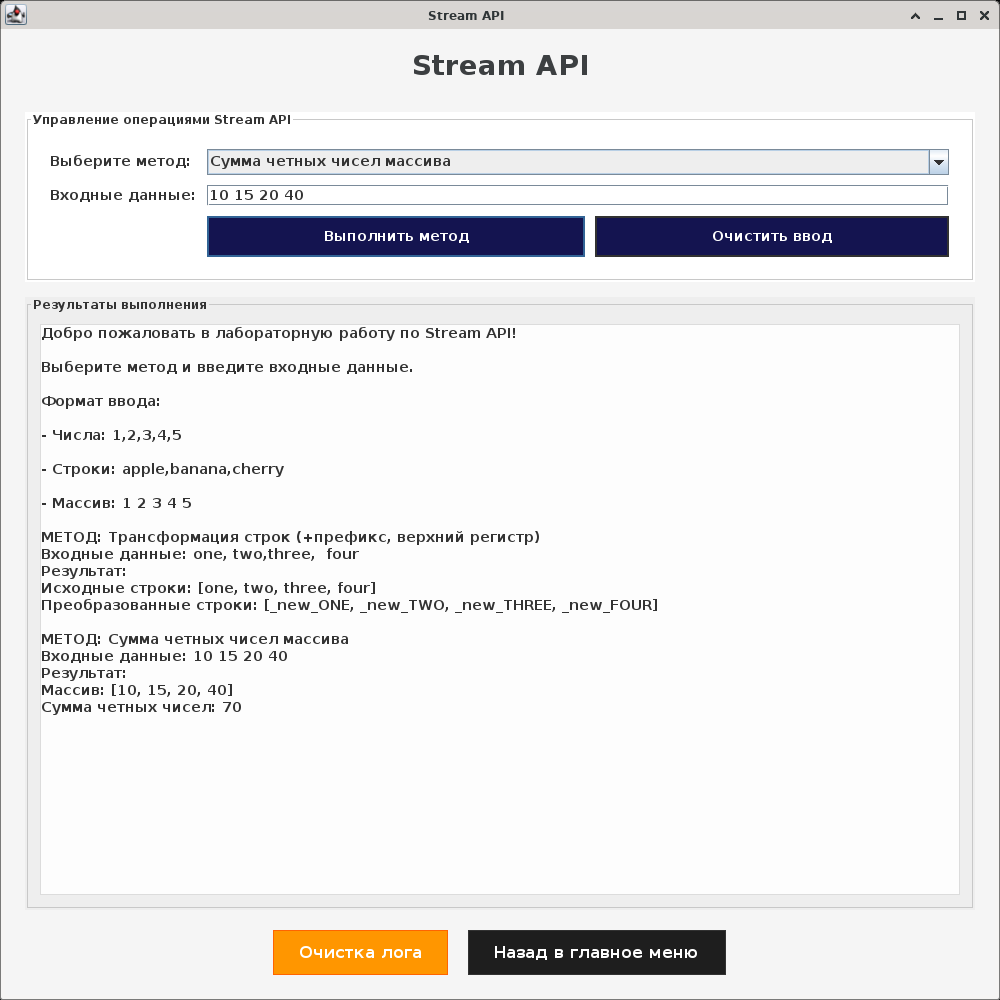
**Интерфейс приложения**

**Рисунок 5.** Главное меню.

**Рисунок 6.** Компьютерная игра.

**Рисунок 7.** Аннотации и рефлексия.

**Рисунок 8.** Переводчик текста.

**Рисунок 9.** Stream API.