

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ  
ИНСТИТУТ КОМПЬЮТЕРНЫХ НАУК И КИБЕРБЕЗОПАСНОСТИ.**

**Направление подготовки:**

**09.03.04 «Программная инженерия» в5130904/30321 гр.**

## **КУРСОВОЙ ПРОЕКТ**

Разработка приложения с графическим интерфейсом для задания 1-4.

По дисциплине: «Объектно-ориентированное программирование»

Выполнил: студент гр. в5130904/30321 \_\_\_\_\_ Д.О. Баранов

Преподаватель \_\_\_\_\_ А.П. Маслаков

«\_\_»\_\_\_\_\_ 2025г.

Санкт-Петербург 2025

## **СОДЕРЖАНИЕ**

Диаграмма классов.....	3
Перечень выполненных работ .....	6
Скриншоты программы .....	10

# Диаграмма классов

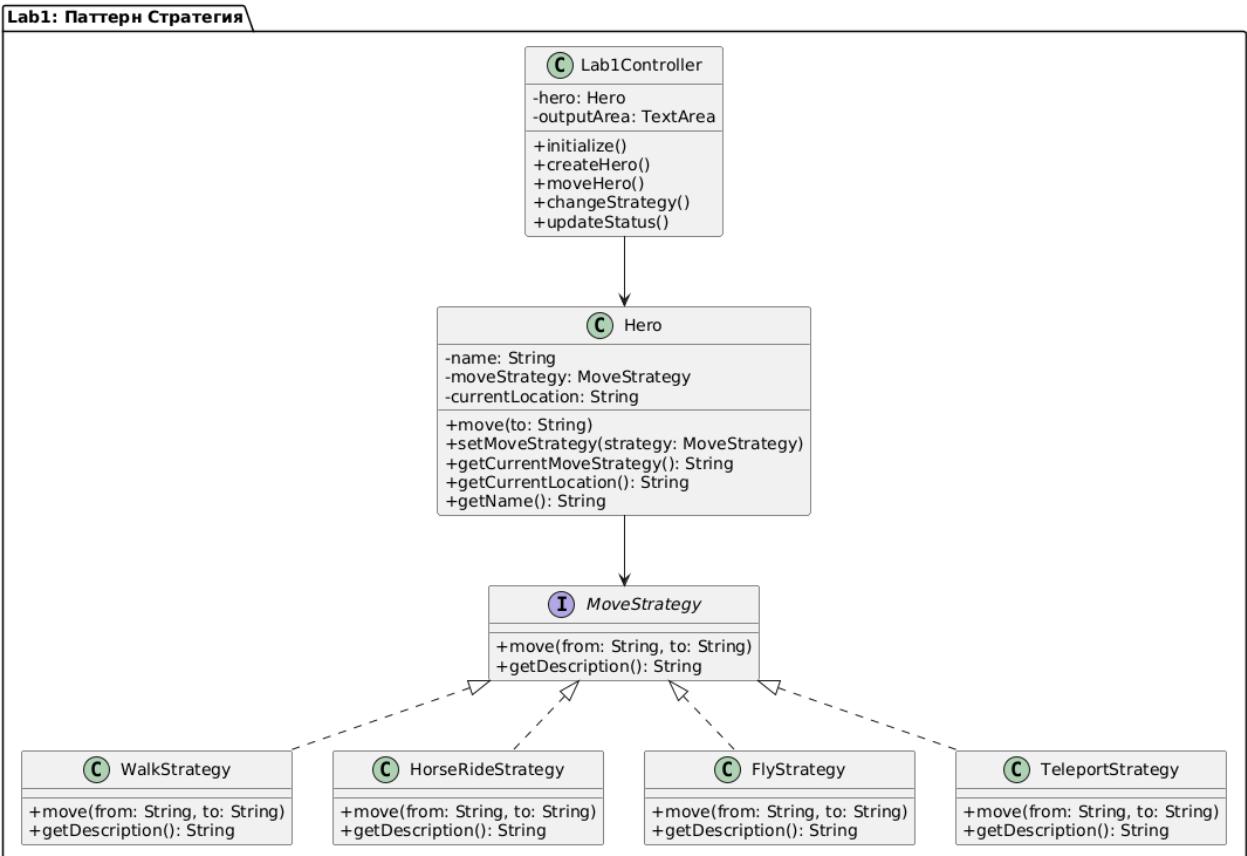


Рисунок 1 – диаграмма классов лабораторной 1

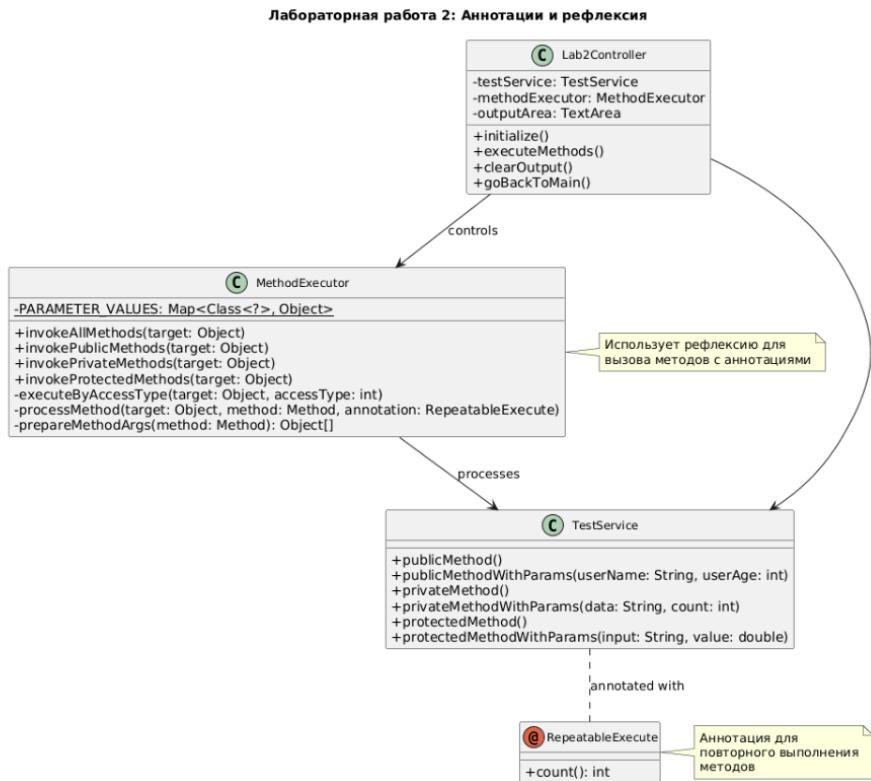


Рисунок 2 – диаграмма классов лабораторной 2

### Лабораторная работа 3: Переводчик

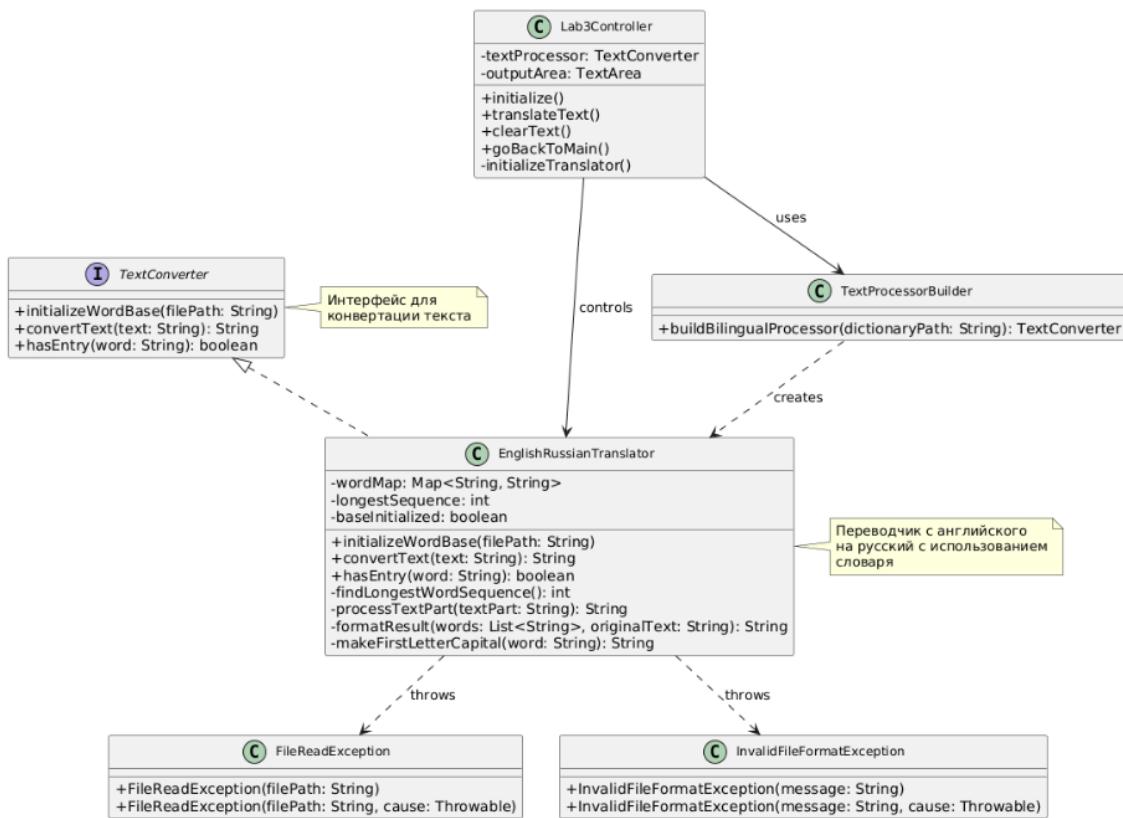


Рисунок 3 – диаграмма классов лабораторной 3

### Лабораторная работа 4: Stream API и коллекции

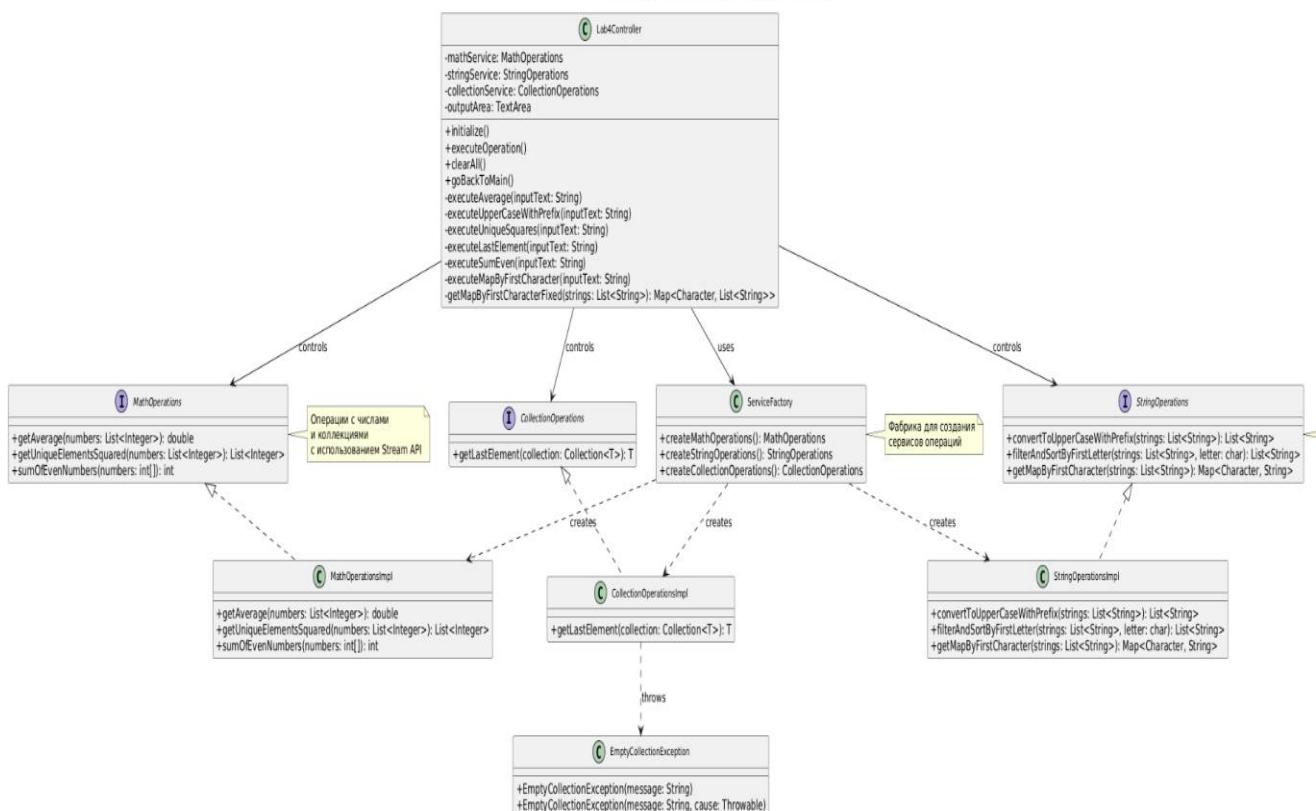


Рисунок 4 – диаграмма классов лабораторной 4

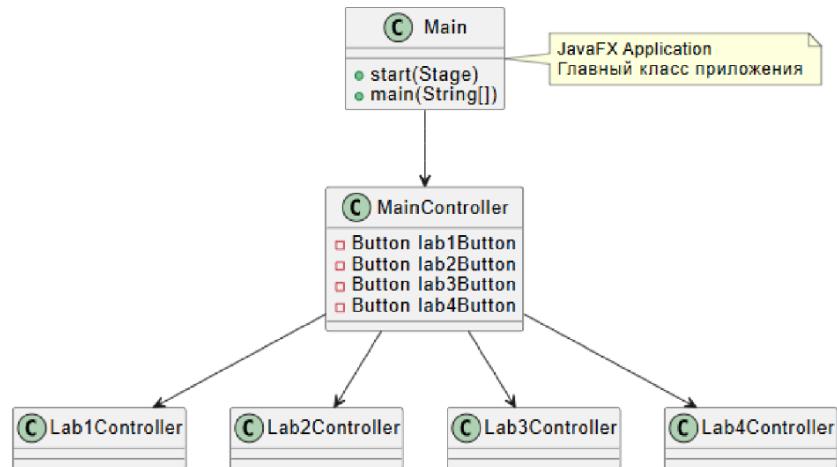


Рисунок 5 – диаграмма классов общей архитектуры приложения

## **Перечень выполненных работ**

Разработано полноценное графическое приложение на JavaFX

- Создана модульная архитектура с главным меню
- Реализована навигация между 4 лабораторными работами
- Оформлен современный пользовательский интерфейс
- Обеспечена обработка пользовательского ввода

Реализована система управления проектом

- Организована структура пакетов (controller, lab1, lab2, lab3, lab4)
- Настроена сборка Maven
- Создана документация и диаграммы классов

### **Лабораторная работа 1: Паттерн "Стратегия"**

Реализована архитектура паттерна "Стратегия". Интерфейс MoveStrategy с методами move() и getDescription().

Конкретные стратегии:

- WalkStrategy - перемещение пешком
- HorseRideStrategy - перемещение на лошади
- FlyStrategy - перемещение полетом
- TeleportStrategy - телепортация

Создан класс Hero

- Динамическая смена стратегий перемещения
- Отслеживание текущего местоположения
- Взаимодействие с пользовательским интерфейсом

Разработан графический интерфейс

- Поле для ввода имени героя
- Выбор начальной локации
- Выбор способа перемещения
- Кнопки управления (создать героя, переместить, сменить стратегию)
- Журнал событий с выводом в TextArea

## **Лабораторная работа 2: Аннотации и рефлексия**

Создана пользовательская аннотация

- @RepeatableExecute с параметром count
- Аннотация применяется к методам с разными модификаторами доступа

Реализован механизм рефлексии

- Класс MethodExecutor для выполнения методов через reflection
- Обработка методов с параметрами разных типов
- Обход ограничений доступа (private/protected методы)
- Группировка методов по типам доступа

Создан графический интерфейс

- RadioButton для выбора типа методов
- Кнопка выполнения методов
- Текстовая область для вывода результатов
- Очистка вывода

## **Лабораторная работа 3: Переводчик**

Реализована система перевода

- Интерфейс TextConverter для конвертации текста
- Класс EnglishRussianTranslator с алгоритмом перевода
- Поиск самых длинных совпадений в тексте
- Обработка многословных выражений

Создана работа с файлами

- Загрузка словаря из файла dictionary.txt
- Обработка исключений: FileReadException, InvalidFormatException
- Валидация формата файла словаря

Разработан графический интерфейс

- Поле ввода текста для перевода
- Кнопка выполнения перевода
- Область вывода результатов

- Статус загрузки словаря
- Поддержка команд help и exit

## Лабораторная работа 4: Stream API и коллекции

Реализованы операции с числами

- Интерфейс MathOperations
- Вычисление среднего значения списка
- Поиск квадратов уникальных элементов
- Сумма четных чисел массива

Созданы операции со строками

- Интерфейс StringOperations
- Преобразование строк в верхний регистр с префиксом
- Фильтрация и сортировка по первой букве
- Преобразование в Map (первый символ → оставшаяся часть)

Реализованы операции с коллекциями

- Интерфейс CollectionOperations
- Получение последнего элемента коллекции
- Обработка пустых коллекций через EmptyCollectionException

Применены Stream API операции

- Использование stream(), filter(), map(), collect()
- Группировка элементов с Collectors.groupingBy()
- Работа с Map и преобразование данных

Разработан графический интерфейс

- ComboBox для выбора операции (6 вариантов)
- Адаптивные подсказки для ввода данных
- Динамическое обновление интерфейса
- Поддержка разных форматов ввода (числа/строки)

## **Дополнительные работы**

Обеспечено качество кода

- Единый стиль оформления
- Обработка исключительных ситуаций
- Логирование ключевых операций
- Валидация пользовательского ввода

Реализована навигация

- Возврат в главное меню из любой лабораторной
- Согласованный пользовательский опыт
- Обработка событий и переходов

## **Итоговые технологии и навыки**

- Java & JavaFX - создание desktop приложений
- Паттерны проектирования - Strategy, Factory, MVC
- Рефлексия и аннотации - метапрограммирование
- Stream API - функциональные операции с данными
- Работа с файлами - I/O операции, исключения
- Коллекции - List, Map, работа с данными
- FXML - декларативное создание интерфейсов
- Maven - управление зависимостями и сборка

**Вывод:** создано полнофункциональное образовательное приложение, демонстрирующее ключевые аспекты программирования на Java.

## Скриншоты программы

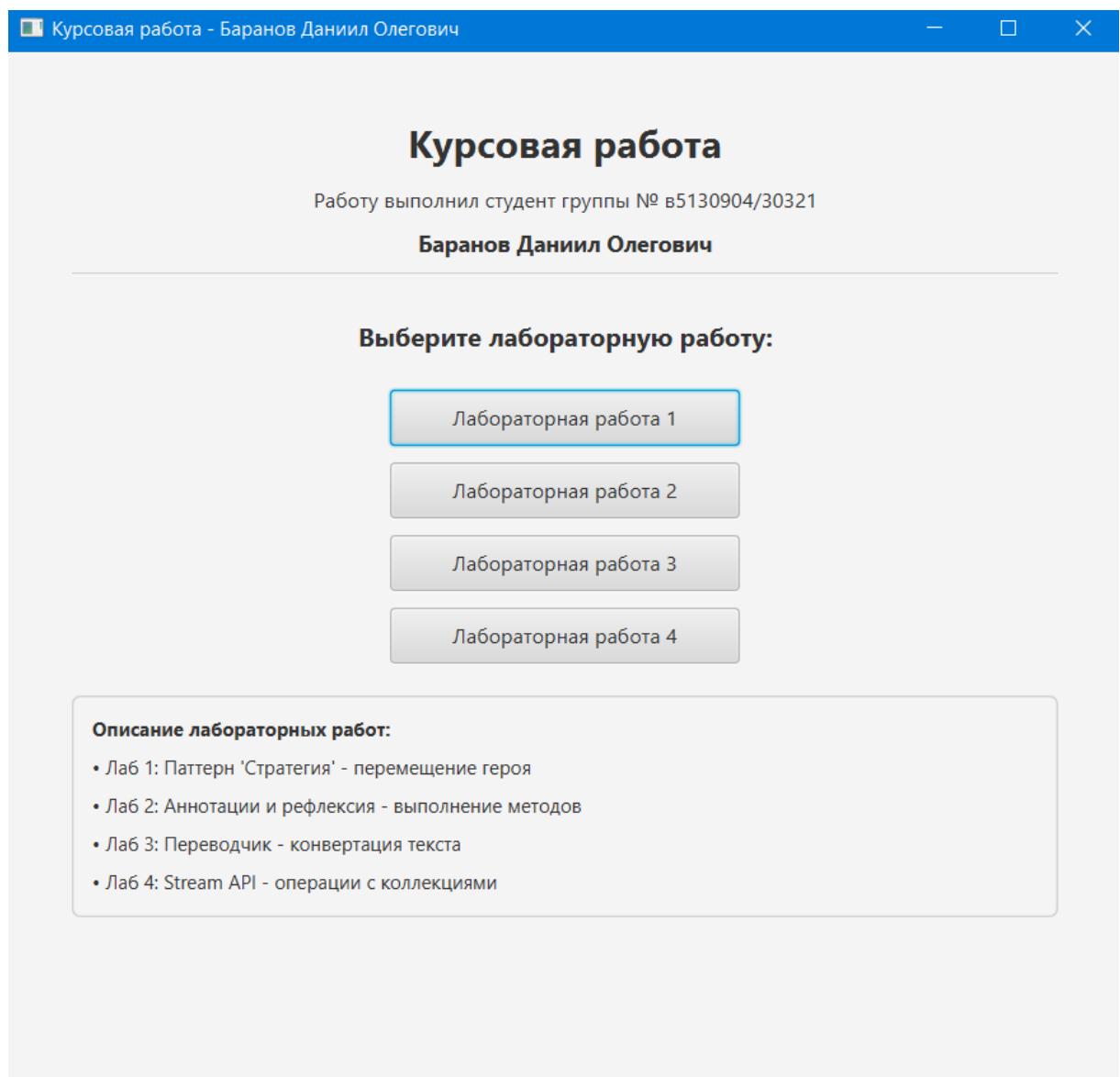


Рисунок 6 – Главное меню

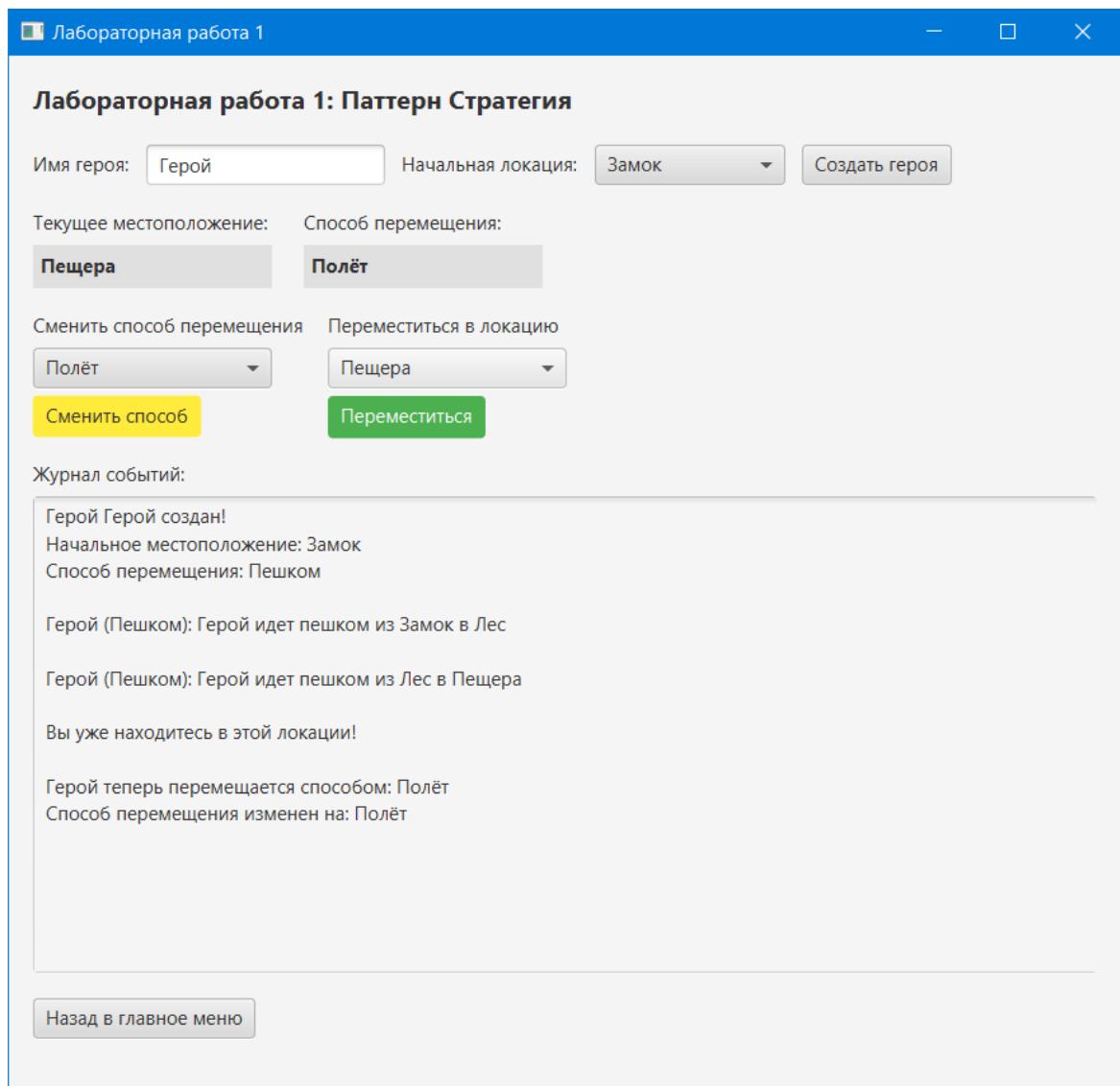


Рисунок 7 – Окно лабораторной работы 1

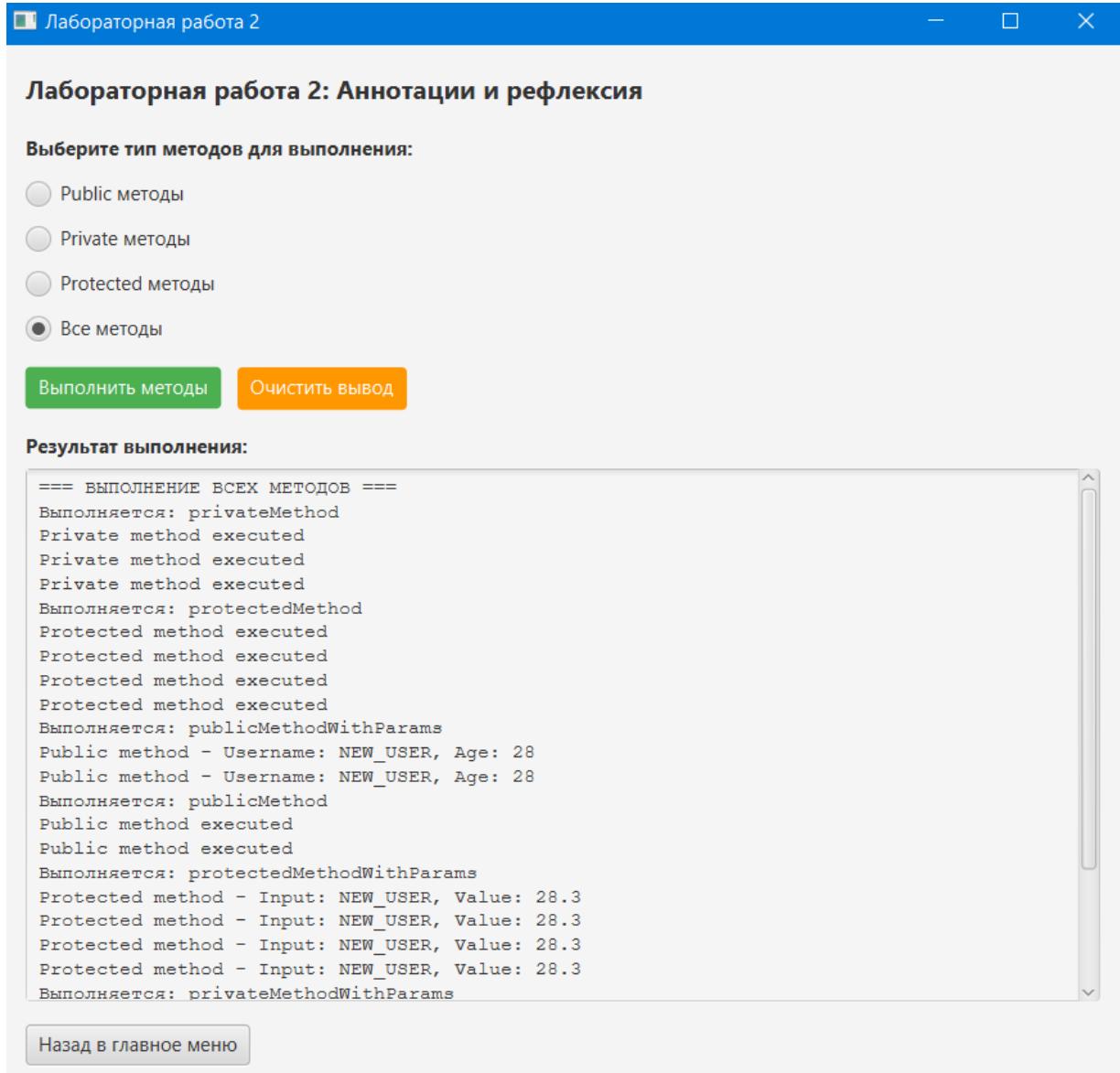


Рисунок 8 – Окно лабораторной работы 2

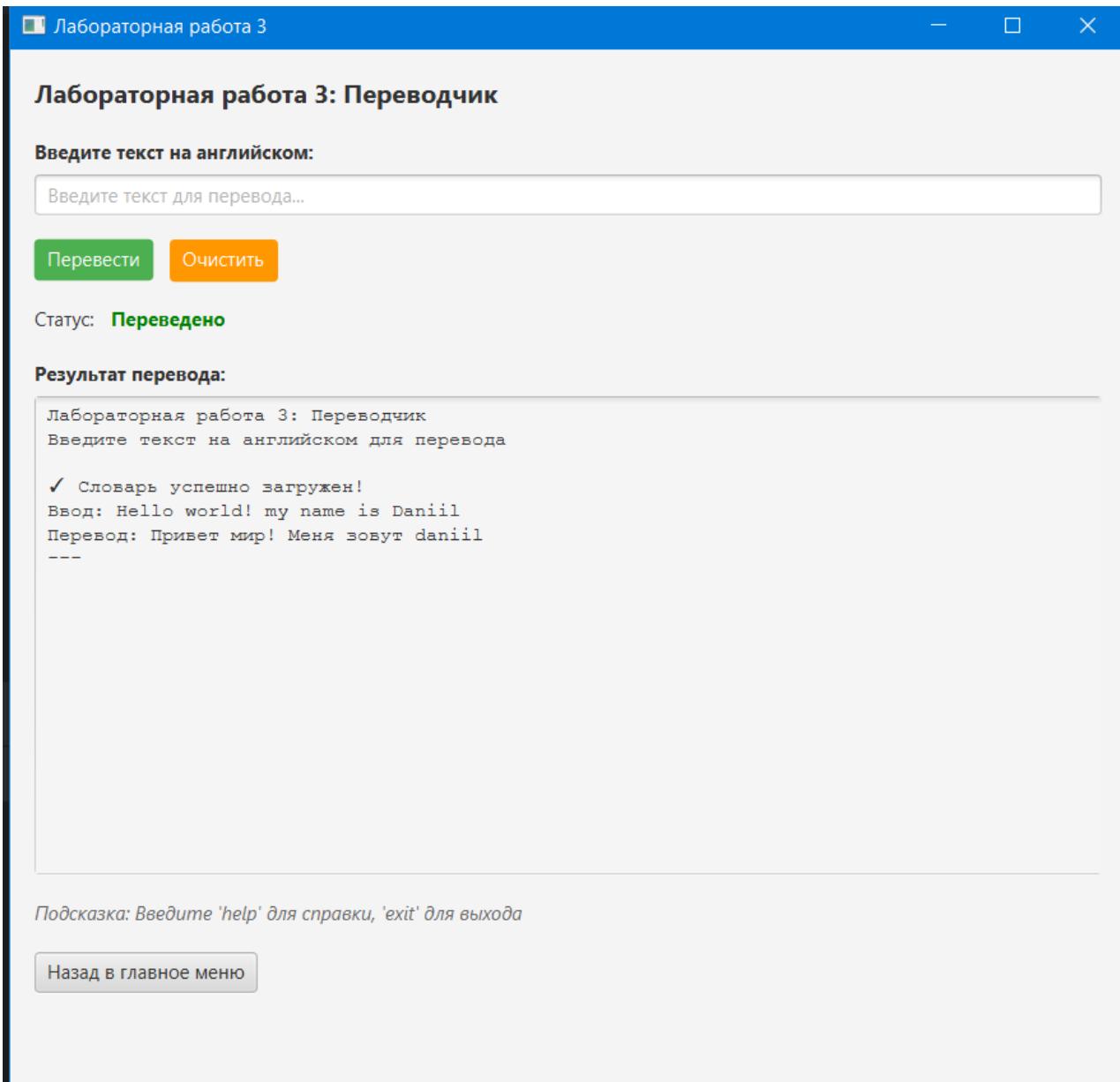


Рисунок 9 – Окно лабораторной работы 3

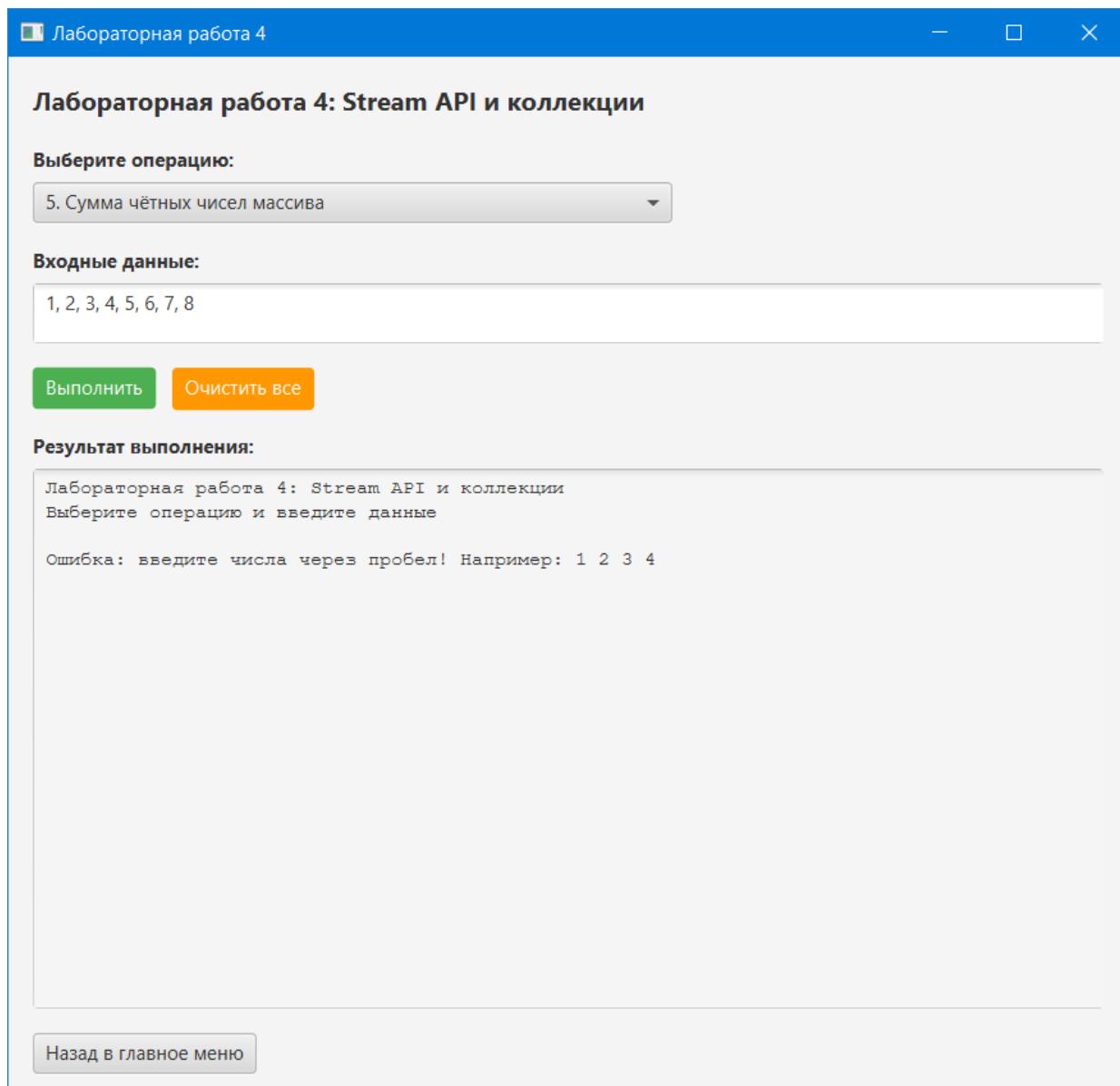


Рисунок 10 – Окно лабораторной работы 4