Санкт-Петербургский политехнический университет Петра Великого (СПБПУ)

Институт компьютерных наук и кибербезопасности

**КУРСОВОЙ ПРОЕКТ**

Разработка приложения с графическим интерфейсом для задания 1-4.

## По дисциплине: «Объектно-ориентированное программирование»

|  |  |
| --- | --- |
| Выполнил студент  гр. в5130904/30022 | Гапонов Д. П. |
| Преподаватель гр. в5130904/30022 | Маслаков А. П. |

Санкт-Петербург 2025

**СОДЕРЖАНИЕ**

[Диаграмма классов 3](#_Toc211709250)

[Перечень выполненных работ. 6](#_Toc211709251)

[Скриншоты. 9](#_Toc211709252)

# Диаграмма классов

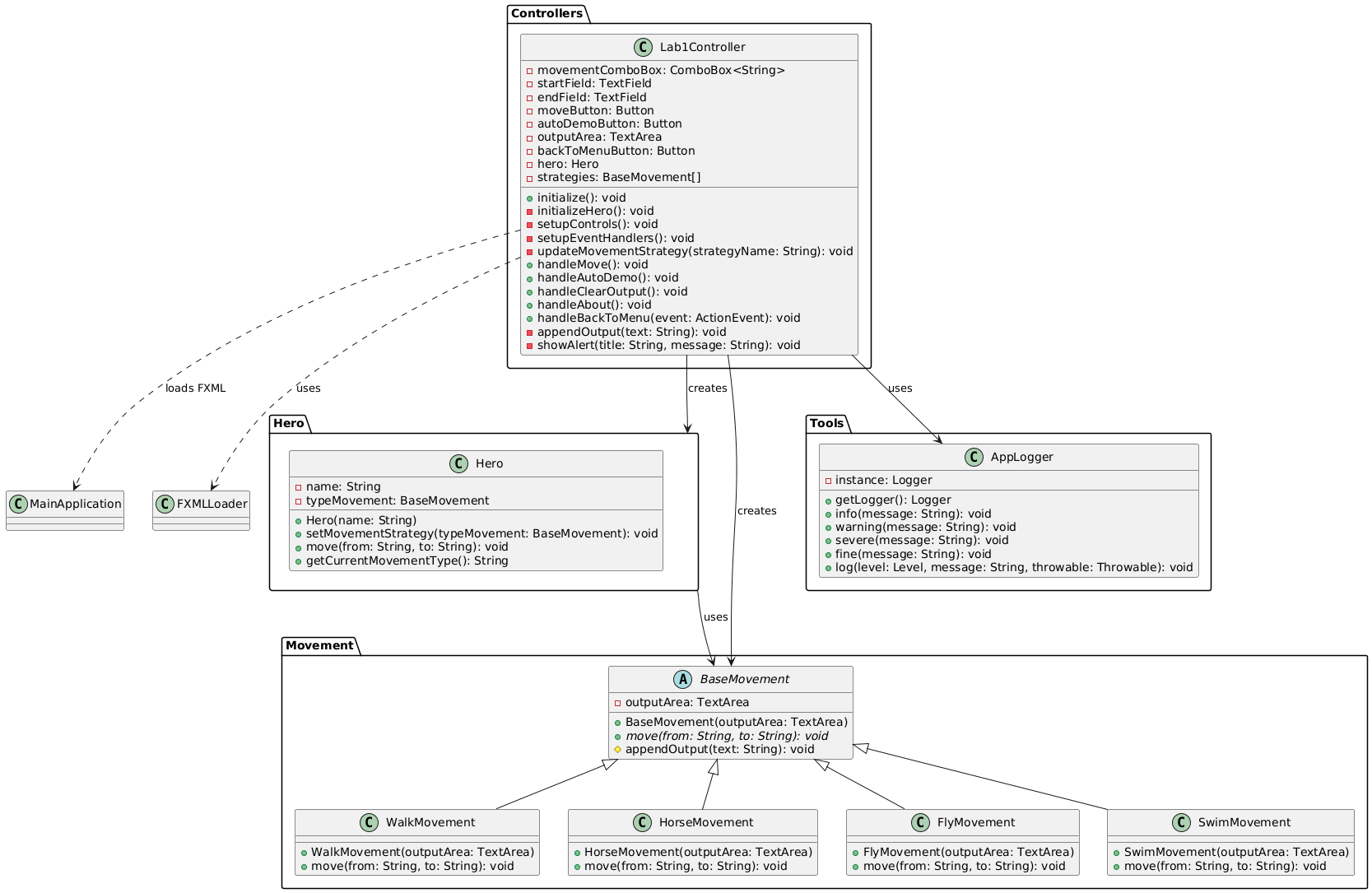
****

Рисунок 1 – диаграмма классов лабораторной 1

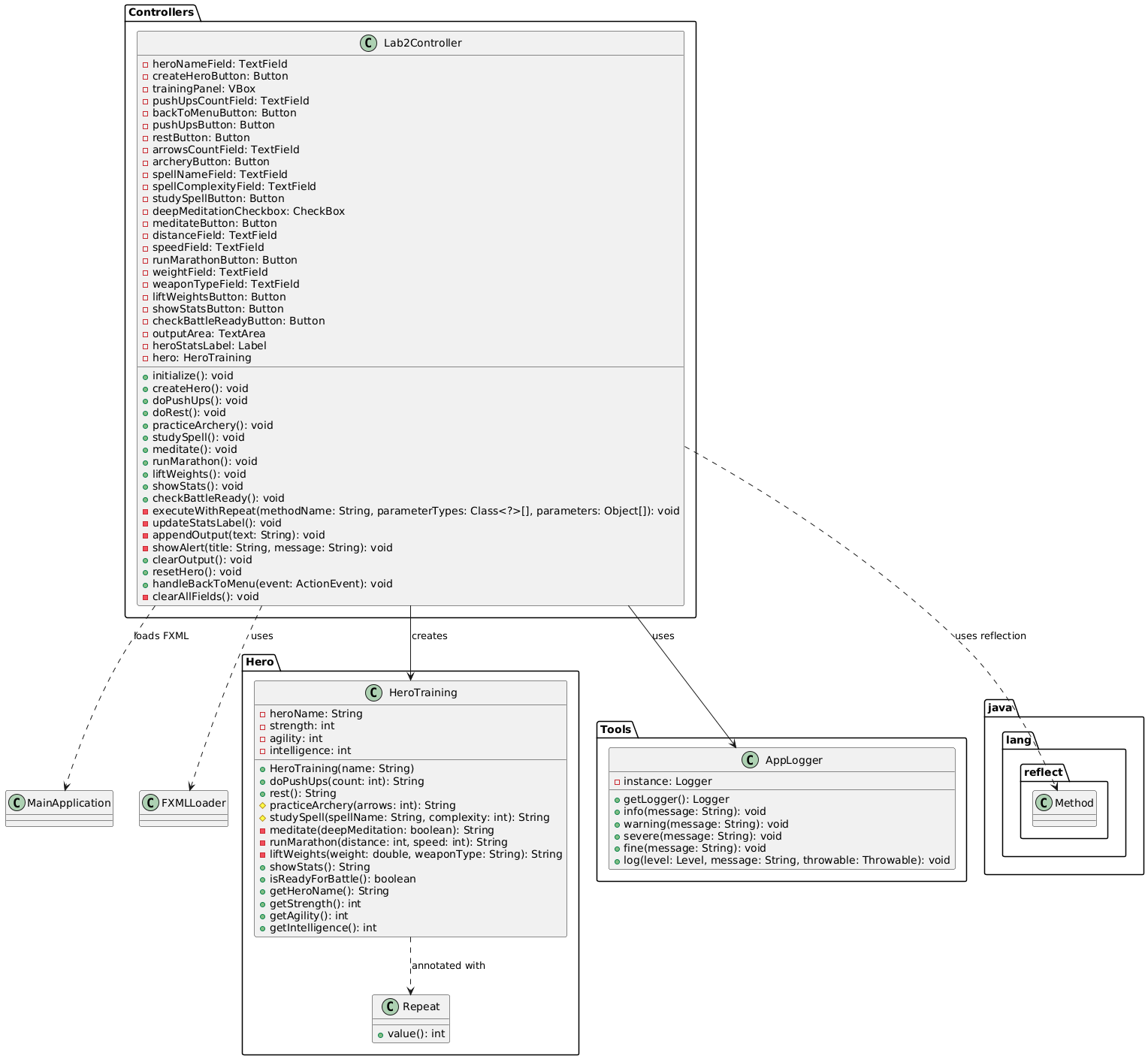


Рисунок 2 – диаграмма классов лабораторной 2

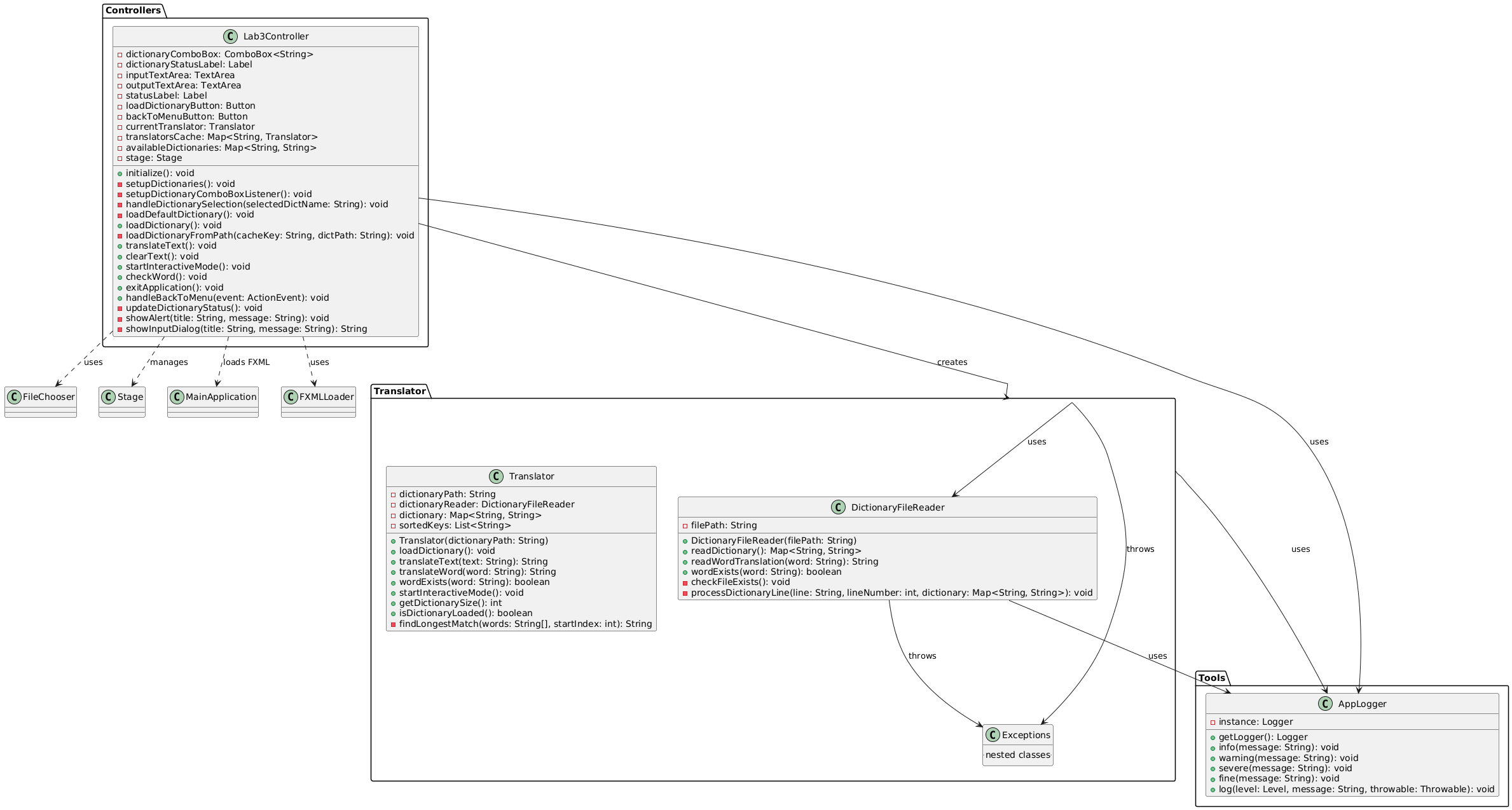


Рисунок 3 – диаграмма классов лабораторной 3

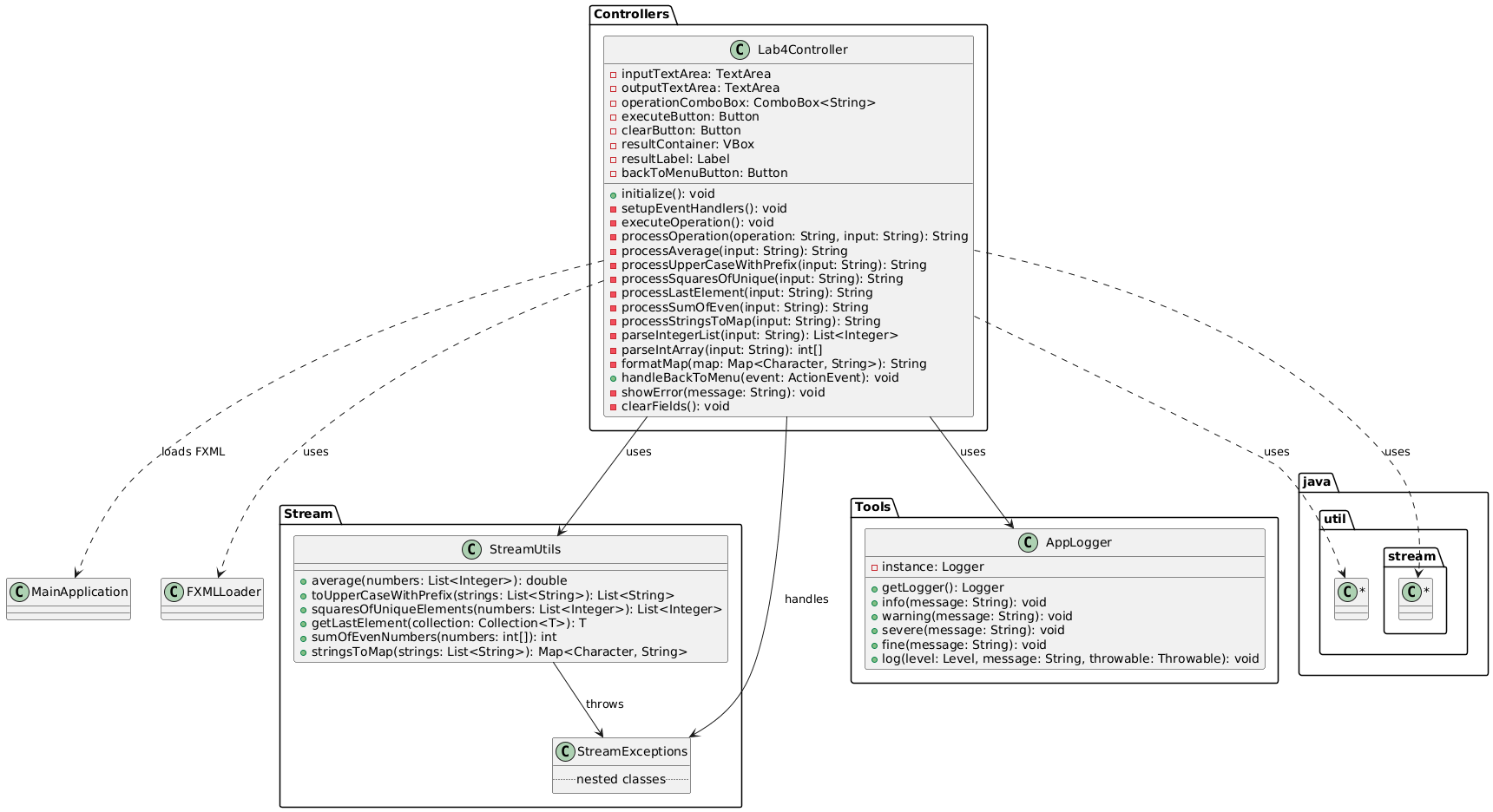


Рисунок 4 – диаграмма классов лабораторной 4

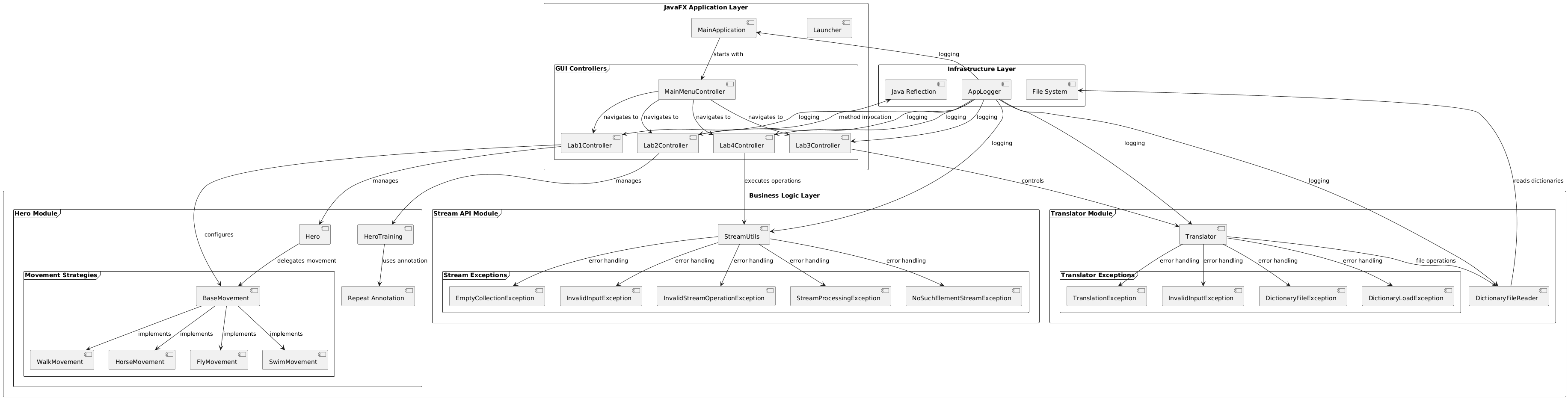


Рисунок 5 – диаграмма классов общей архитектуры приложения

# Перечень выполненных работ.

## 1. Паттерн «Стратегия» (Lab1)

* **Реализован класс**Hero с методом move() и динамической сменой стратегий перемещения через setMovementStrategy()
* **Созданы классы стратегий перемещения:**
  + - WalkMovement - перемещение пешком
    - HorseMovement - перемещение на лошади
    - FlyMovement - воздушное перемещение
    - SwimMovement - водное перемещение
* **Абстрактный базовый класс**BaseMovement обеспечивает единый интерфейс для всех стратегий
* **Графический интерфейс (**Lab1Controller**)** предоставляет:
  + - Выбор стратегии через ComboBox
    - Поля для ввода начальной и конечной точек
    - Кнопки для выполнения перемещения и автоматической демонстрации
    - Защищенную текстовую область для вывода результатов

## 2. Аннотации и рефлексия (Lab2)

* **Создана пользовательская аннотация** @Repeat с параметром количества вызовов
* **Разработан класс** HeroTraining с методами различных уровней доступа:
  + - public: doPushUps(), rest(), showStats()
    - protected: practiceArchery(), studySpell()
    - private: meditate(), runMarathon(), liftWeights()
* **Реализован механизм вызова через рефлексию** в методе executeWithRepeat():
  + - Автоматическое определение количества повторений из аннотации
    - Обработка методов с различными параметрами
    - Динамический вызов через Method.invoke()
* **Графический интерфейс** (Lab2Controller) включает:
  + - Панель создания героя с именем
    - Разделы для различных типов тренировок
    - Отображение статистики характеристик (сила, ловкость, интеллект)
    - Проверку готовности к битве

## 3. Программа-переводчик (Lab3)

* **Реализованы классы** для работы со словарями:
  + - Translator - основной класс для перевода текста
    - DictionaryFileReader - загрузчик словарей из файлов
* **Поддержка различных форматов словарей:**
  + - Предустановленные словари (англо-русский, русско-английский, немецко-русский)
    - Пользовательские словари через файловый диалог
* **Алгоритм перевода** с greedy matching**:**
  + - Поиск самых длинных совпадений в тексте
    - Игнорирование регистра символов
    - Кэширование загруженных словарей
* **Пользовательские исключения** (Exceptions):
  + - DictionaryFileException - ошибки файловой системы
    - DictionaryLoadException - ошибки загрузки словаря
    - TranslationException - ошибки перевода
    - InvalidInputException - некорректные входные данные
* **Графический интерфейс** (Lab3Controller) предоставляет:
  + - Выбор словаря через ComboBox
    - Поля для ввода текста и отображения перевода
    - Интерактивный режим перевода
    - Проверку отдельных слов в словаре

## 4. Работа со Stream API (Lab4)

* **Реализован класс StreamUtils** с методами:
  + - average() - вычисление среднего значения списка чисел
    - toUpperCaseWithPrefix() - преобразование строк в верхний регистр с префиксом "new"
    - squaresOfUniqueElements() - квадраты уникальных элементов списка
    - getLastElement() - получение последнего элемента коллекции
    - sumOfEvenNumbers() - сумма четных чисел массива
    - stringsToMap() - преобразование строк в Map по первому символу
* **Пользовательские исключения** (StreamExceptions):
  + - EmptyCollectionException - обработка пустых коллекций
    - InvalidInputException - некорректные входные данные
    - StreamProcessingException - ошибки обработки потоков
* **Графический интерфейс** (Lab4Controller) включает:
  + - Выбор операции через ComboBox
    - Единое текстовое поле для ввода данных
    - Отображение результатов операций
    - Обработку различных форматов ввода (числа, строки)

## 5. Система логирования

* **Централизованный логгер** AppLogger обеспечивает**:**
  + - Единый формат вывода логов
    - Различные уровни логирования (INFO, WARNING, SEVERE, FINE)
    - Интеграцию со всеми компонентами приложения

## 6. Интеграция в единое приложение

* **Главное меню** (MainMenuController) обеспечивает навигацию между лабораторными работами
* **Архитектура приложения:**
  + - MainApplication - основной класс JavaFX приложения
    - Launcher - точка входа для совместимости с различными средами выполнения
    - Разделение на пакеты: Controllers, Classes/Hero, Classes/Movement, Classes/Translator, Classes/Stream, Classes/Tools
* **Согласованный пользовательский интерфейс:**
  + - Единый размер окон (800×600 для меню, 900×700 для лабораторных)
    - Стандартизированные элементы управления
    - Обработка ошибок с выводом в интерфейс

## 7. Особенности реализации

* **Полная поддержка русского языка** в интерфейсе и сообщениях
* **Валидация входных данных** во всех контроллерах
* **Обработка исключительных ситуаций** с информативными сообщениями
* **Интуитивная навигация** между окнами приложения
* **Модульная архитектура** с четким разделением ответственности

## 8. Технические характеристики

* **Язык программирования:** Java
* **Фреймворк GUI:** JavaFX
* **Архитектурные паттерны:** MVC (Model-View-Controller), Strategy, Factory
* **Используемые технологии:** Аннотации, Рефлексия, Stream API, Коллекции
* **Система логирования:** java.util.logging

# Скриншоты.

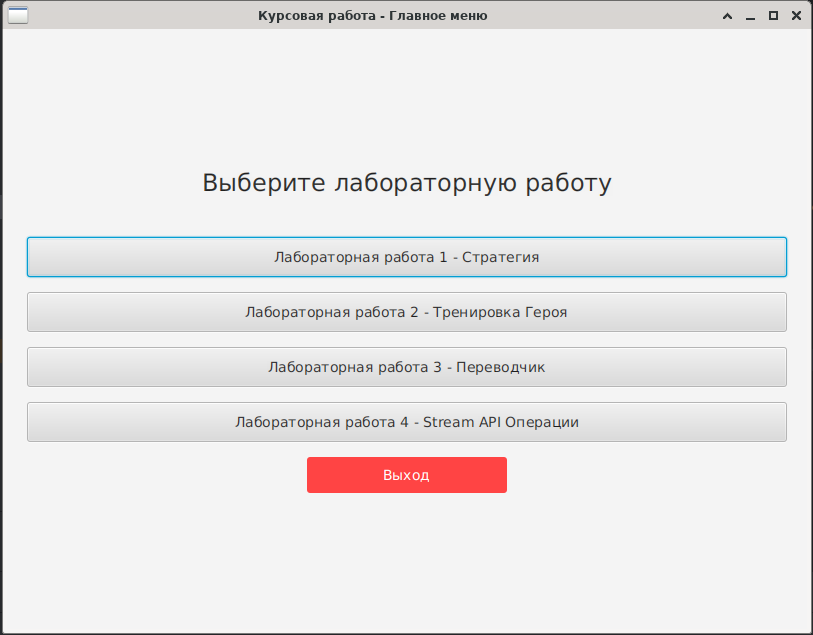


Рисунок 6 – Главное окно

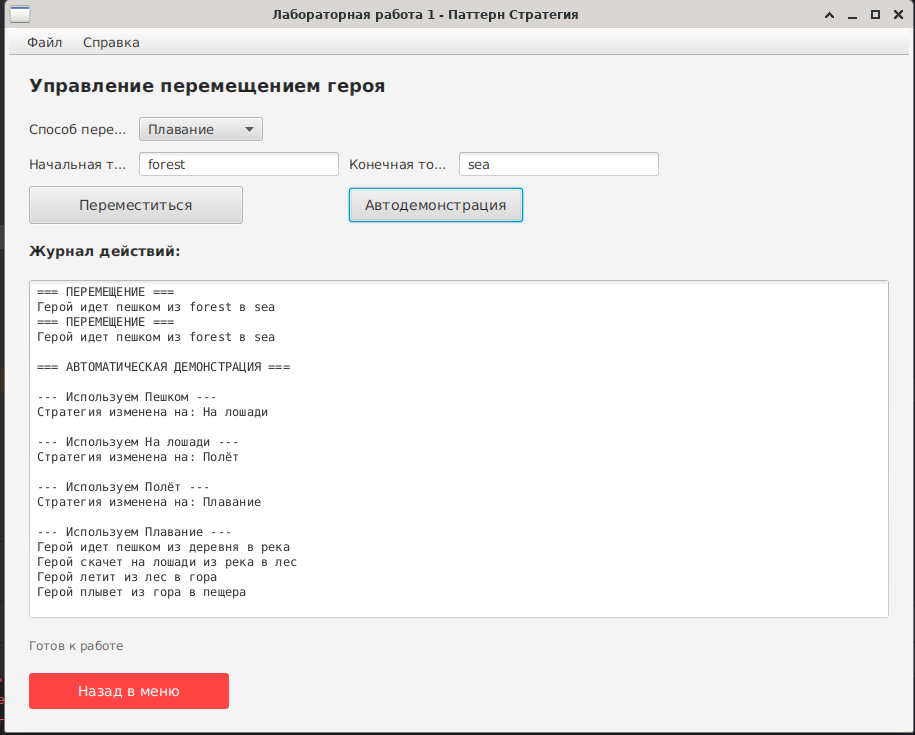


Рисунок 7 – Окно лабораторной работы 1

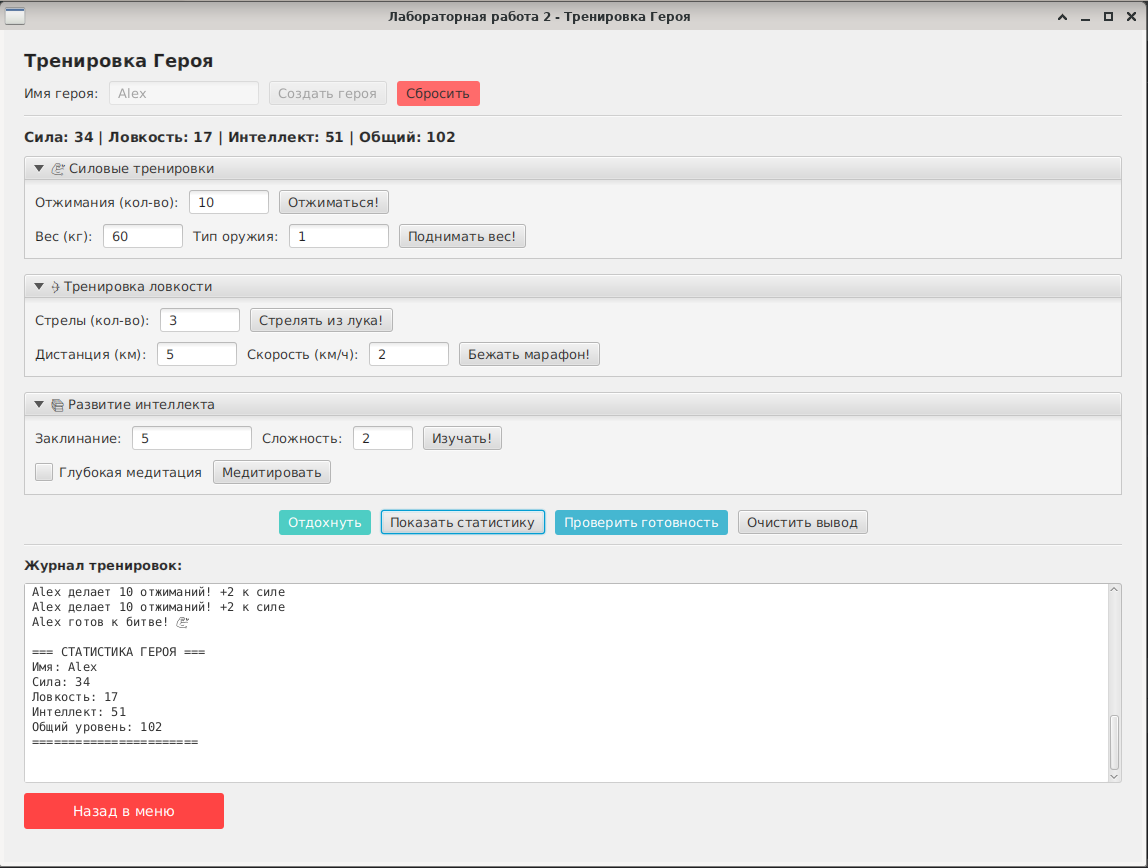


Рисунок 8 – Окно лабораторной работы 2

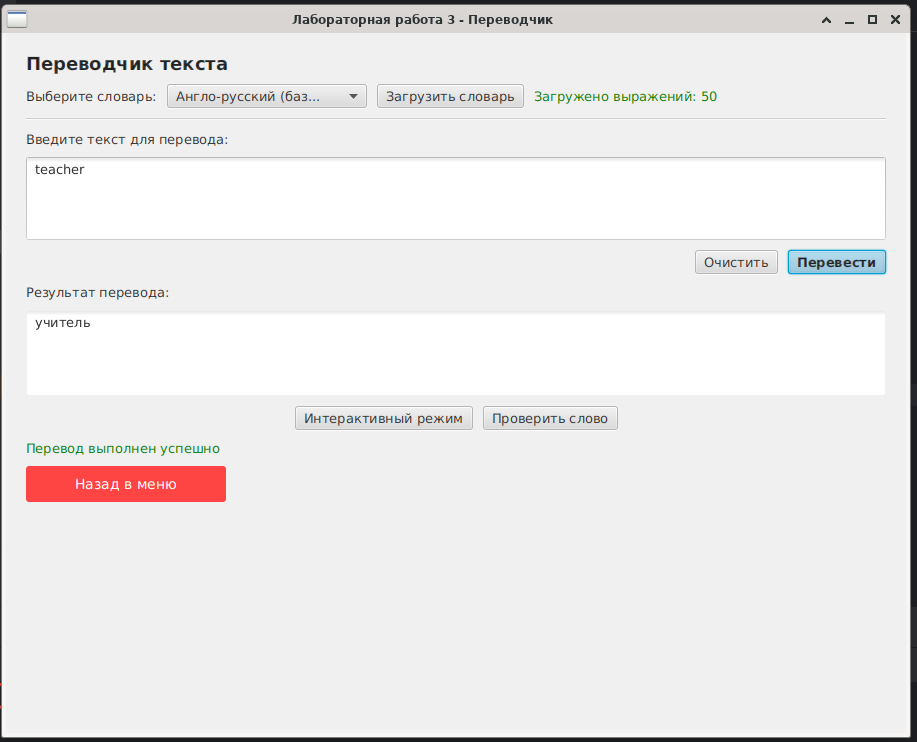


Рисунок 9 – Окно лабораторной работы 3

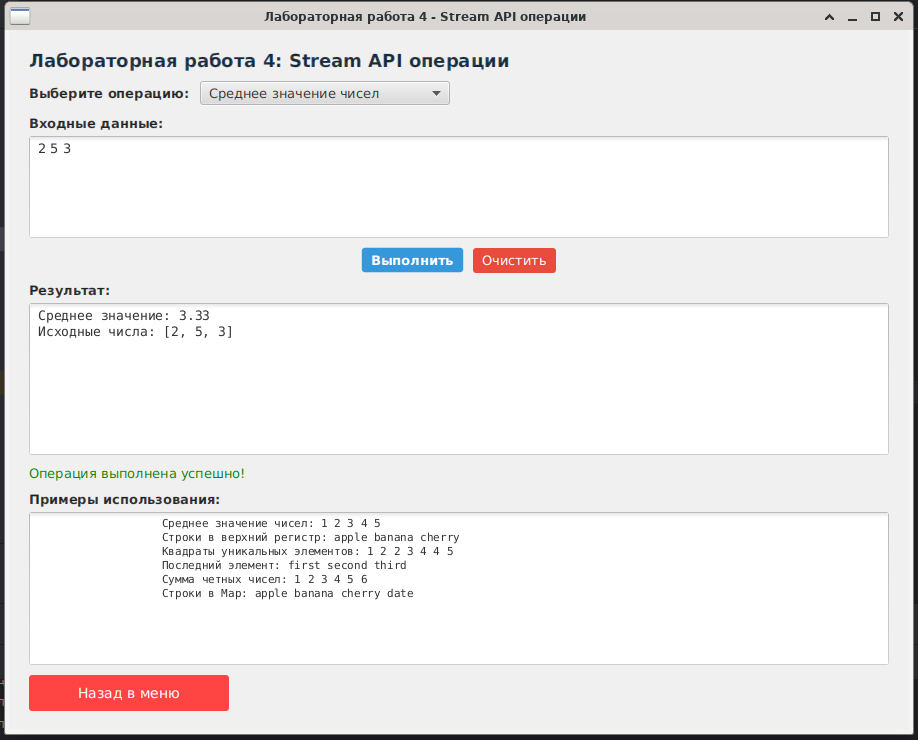


Рисунок 10 – Окно лабораторной работы 4