

**Министерство науки и высшего образования Российской  
Федерации**  
**Санкт-Петербургский политехнический университет Петра  
Великого**  
**Высшая школа программной инженерии**

**КУРСОВАЯ РАБОТА**

по дисциплине: «Объектно-ориентированное программирование»

Выполнил  
студент гр. 30321 \_\_\_\_\_ Д.Ю. Повеличенко

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

«\_\_\_» \_\_\_\_ 20\_\_\_\_ г.

Санкт-Петербург  
2025

## **Задание**

Разработать приложение с графическим интерфейсом для заданий 1–4.

Для этого приложения должна быть реализована возможность выбора из списка любого приложения, ввод входных данных и его выполнение. Модифицировать задания 1–4 так, чтобы весь вывод происходил в текстовых областях, защищённых от редактирования. Предусмотреть для заданий:

- 3 - выбор файлов словаря и текста для перевода, возможность ручного ввода текста;
- 4 - ввод входных данных для методов.

Отчёт по курсовой должен содержать титульный лист, диаграмму классов и перечень выполненных работ.

## **Ход выполнения работы**

Приложение состоит из 5 модулей: четырёх модулей заданий (lab01...lab04) и модуля интерфейса (ui). Для каждого модуля и для всего приложения была составлена диаграмма классов.

## Задание №1

В компьютерной игре герой (класс Hero) может перемещаться между двумя точками (метод move) различными способами: идти пешком, ехать на лошади, лететь и т. п. Реализовать классы, позволяющие пользователю выбирать и менять в ходе выполнения программы способ перемещения героя, используя паттерн “стратегия” (strategy).

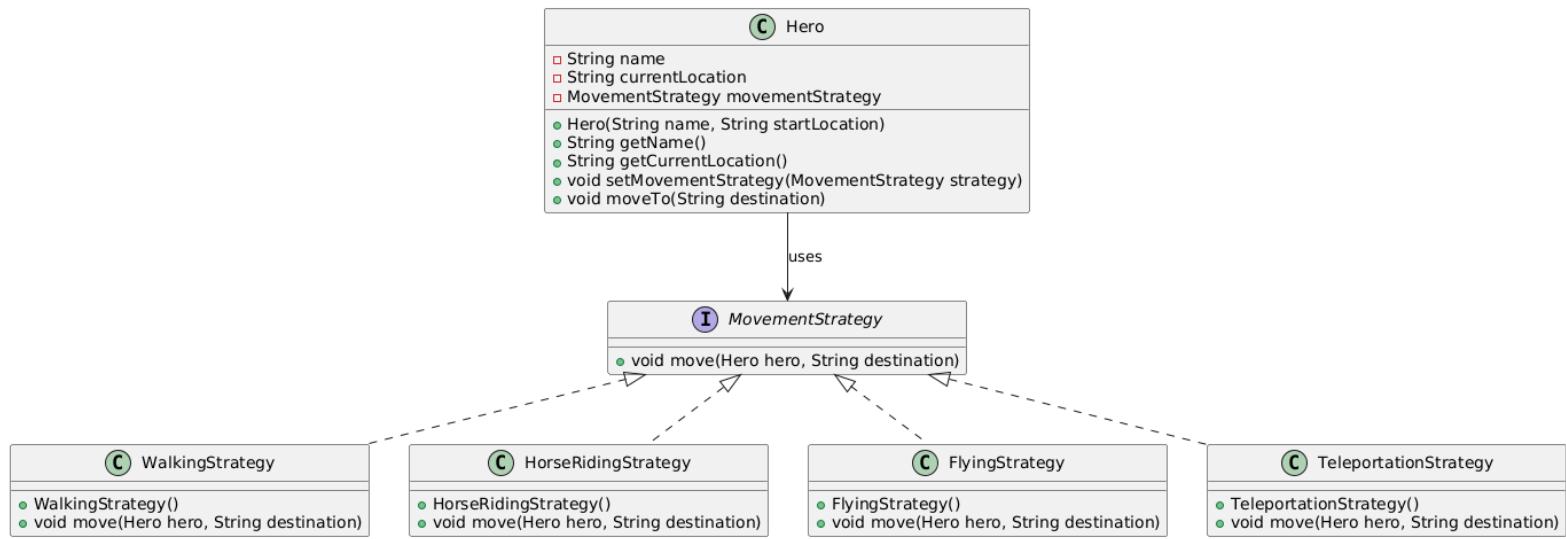


Рисунок 1 – Диаграмма классов для модуля lab01

## Задание №2

Написать аннотацию с целочисленным параметром. Создать класс, содержащий публичные, защищенные и приватные методы (2–3 каждого вида) с параметрами, аннотировать любые из них. Вызвать из другого класса все аннотированные защищенные и приватные методы столько раз, сколько указано в параметре аннотации. Вызывающий методы код не должен зависеть от количества и типов параметров этих методов.

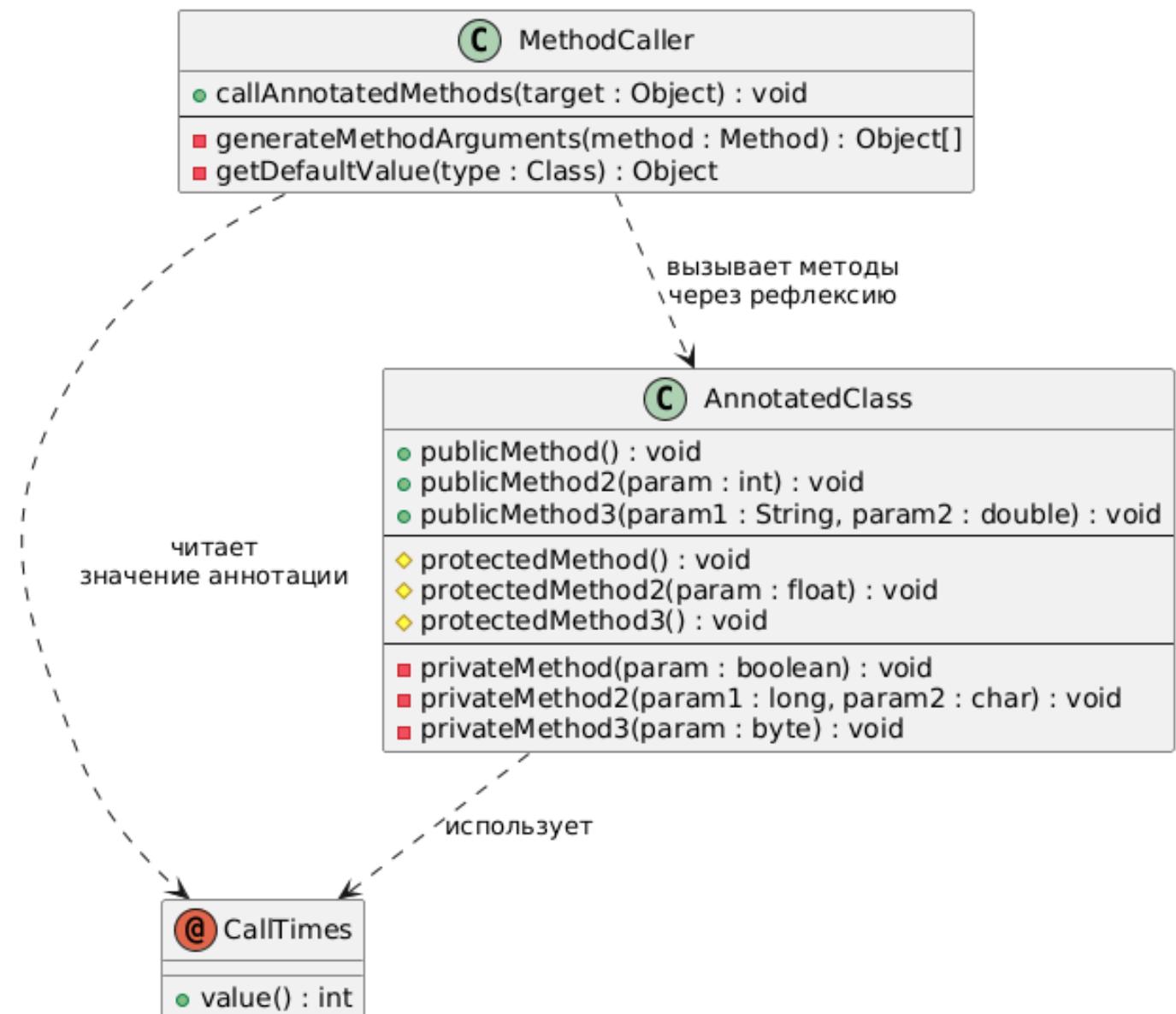


Рисунок 2 – Диаграмма классов для модуля lab02

### **Задание №3**

Реализовать программу-переводчик.

- 1) При запуске программы выполняется чтение словаря из файла, записанного в следующем формате: слово или выражение | перевод
- 2) Затем запрашивается пользовательский ввод в консоли и выполняется перевод введённого текста.
- 3) Перевод осуществляется по следующим правилам:
  - регистр букв игнорируется;
  - если искомого слова нет в словаре – выводится без перевода;
  - если есть несколько подходящих вариантов, выбирается вариант с максимальной длиной левой части. Например:
    - словарь:
      - o look | смотреть
      - o look forward | ожидать
    - текст:
      - o dog look to the window, dog look forward
    - перевод:
      - o dog смотреть to the window, dog ожидать
- 4) Результат перевода выводится в консоль.
- 5) Создать и применить пользовательские исключения:
  - `InvalidFormatException` см. пункт 1;
  - `FileNotFoundException` файла не существует, нет доступа к файлу и т. д.

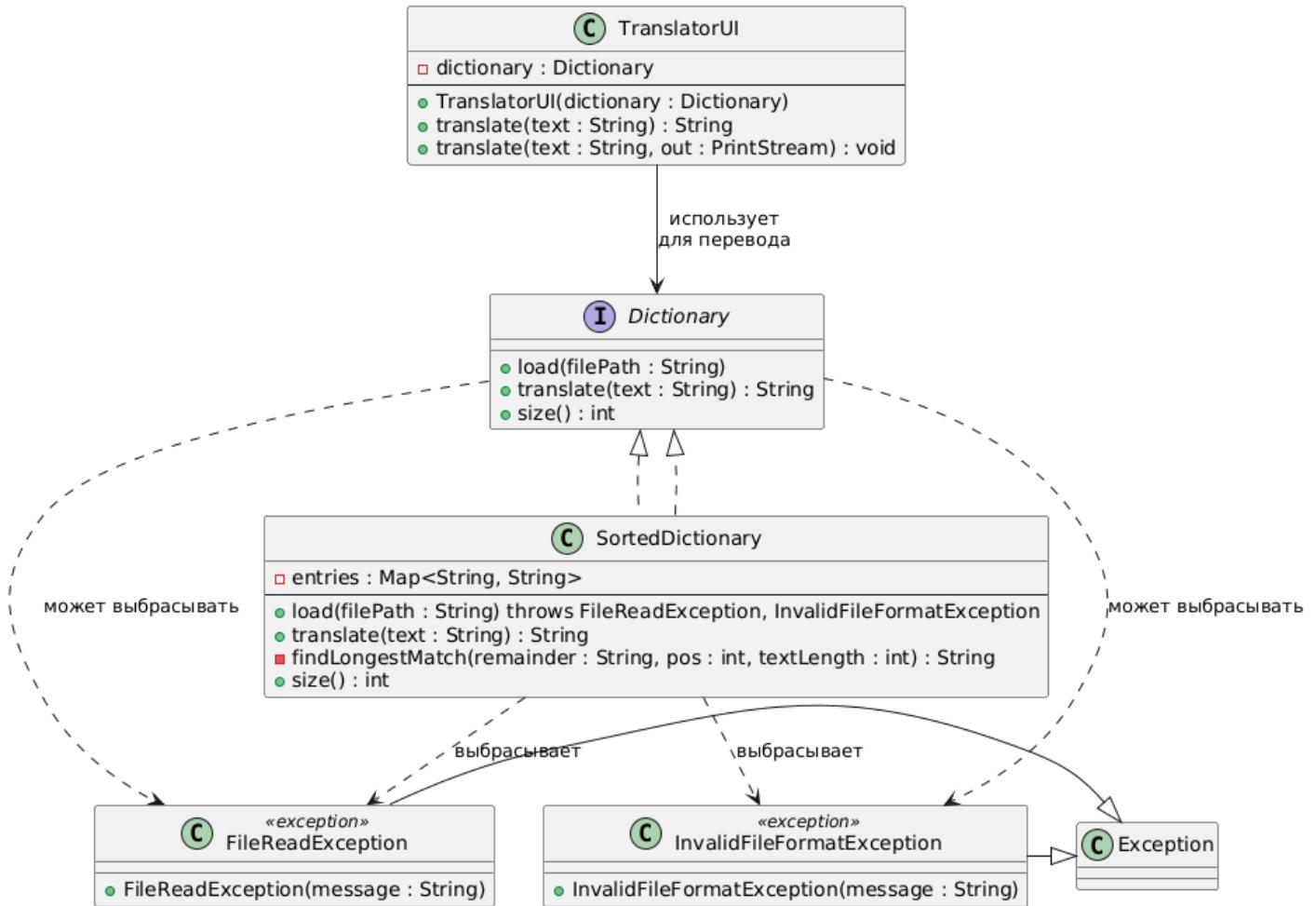


Рисунок 3 – Диаграмма классов для модуля lab03

## Задание №4

С использованием только Stream API реализовать следующие методы:

- метод, возвращающий среднее значение списка целых чисел;
- метод, приводящий все строки в списке в верхний регистр и добавляющий к ним префикс «\_new\_»;
- метод, возвращающий список квадратов всех встречающихся только один раз элементов списка;
- метод, принимающий на вход коллекцию и возвращающий ее последний элемент или кидающий исключение, если коллекция пуста;
- метод, принимающий на вход массив целых чисел, возвращающий сумму чётных чисел или 0, если чётных чисел нет;
- метод, преобразовывающий все строки в списке в Map, где первый символ – ключ, оставшиеся – значение;

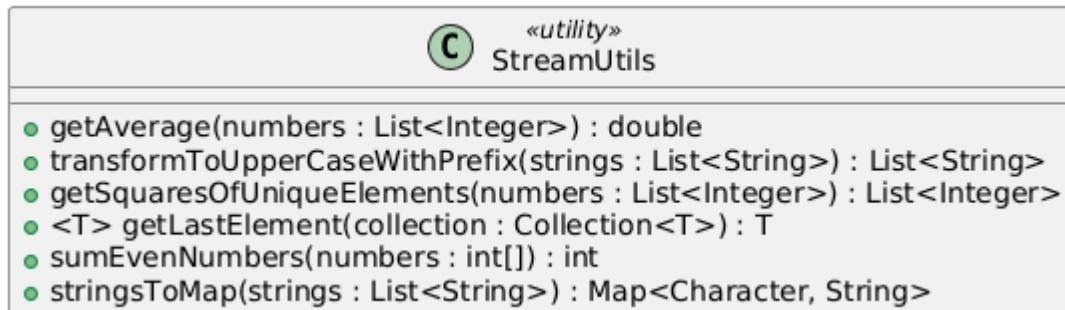


Рисунок 4 – Диаграмма классов для модуля lab04

Для модулей lab01-lab04 был реализован графический интерфейс.

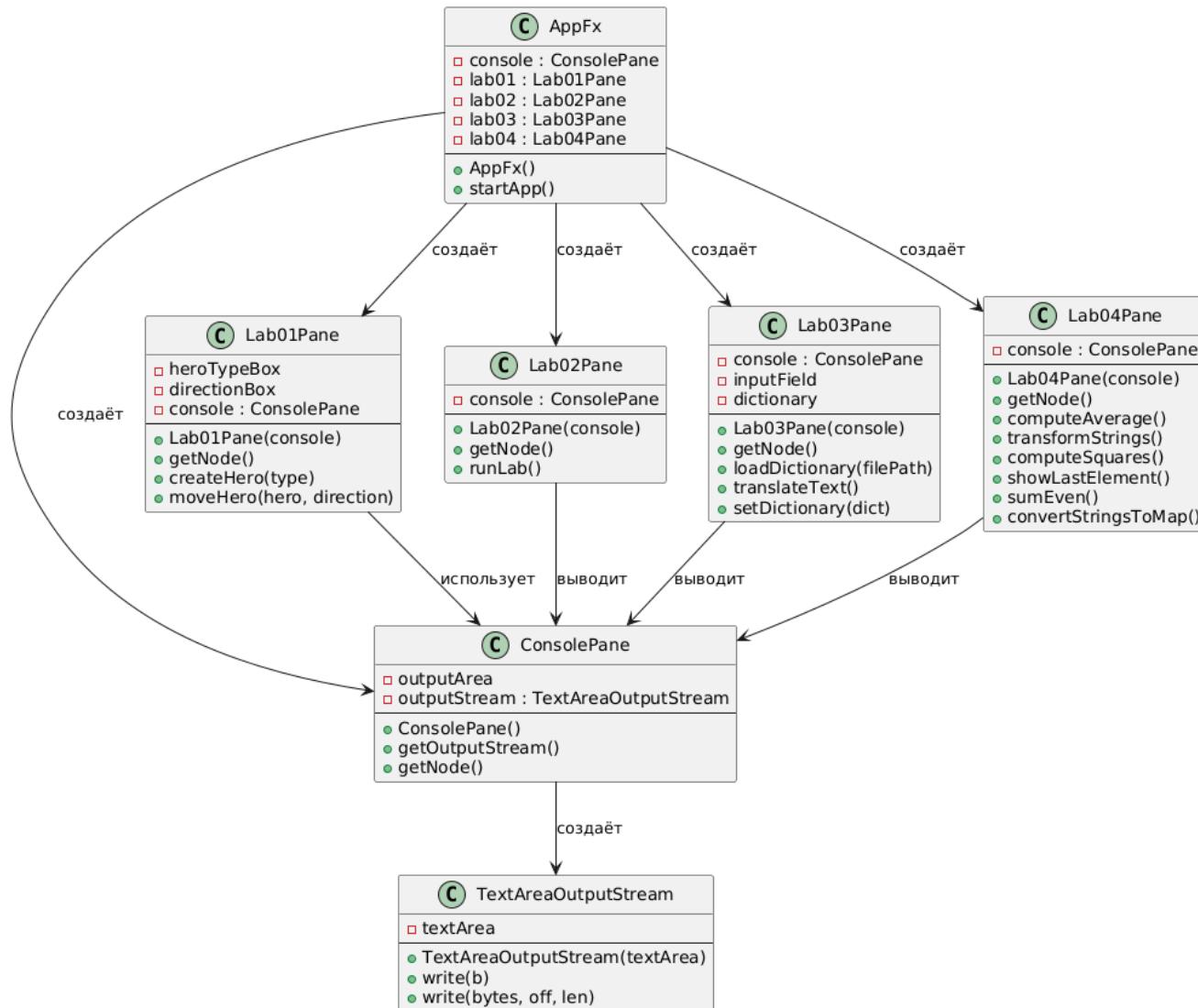


Рисунок 5 – Диаграмма классов для модуля ui + управляющего класса

## Демонстрация работы приложения

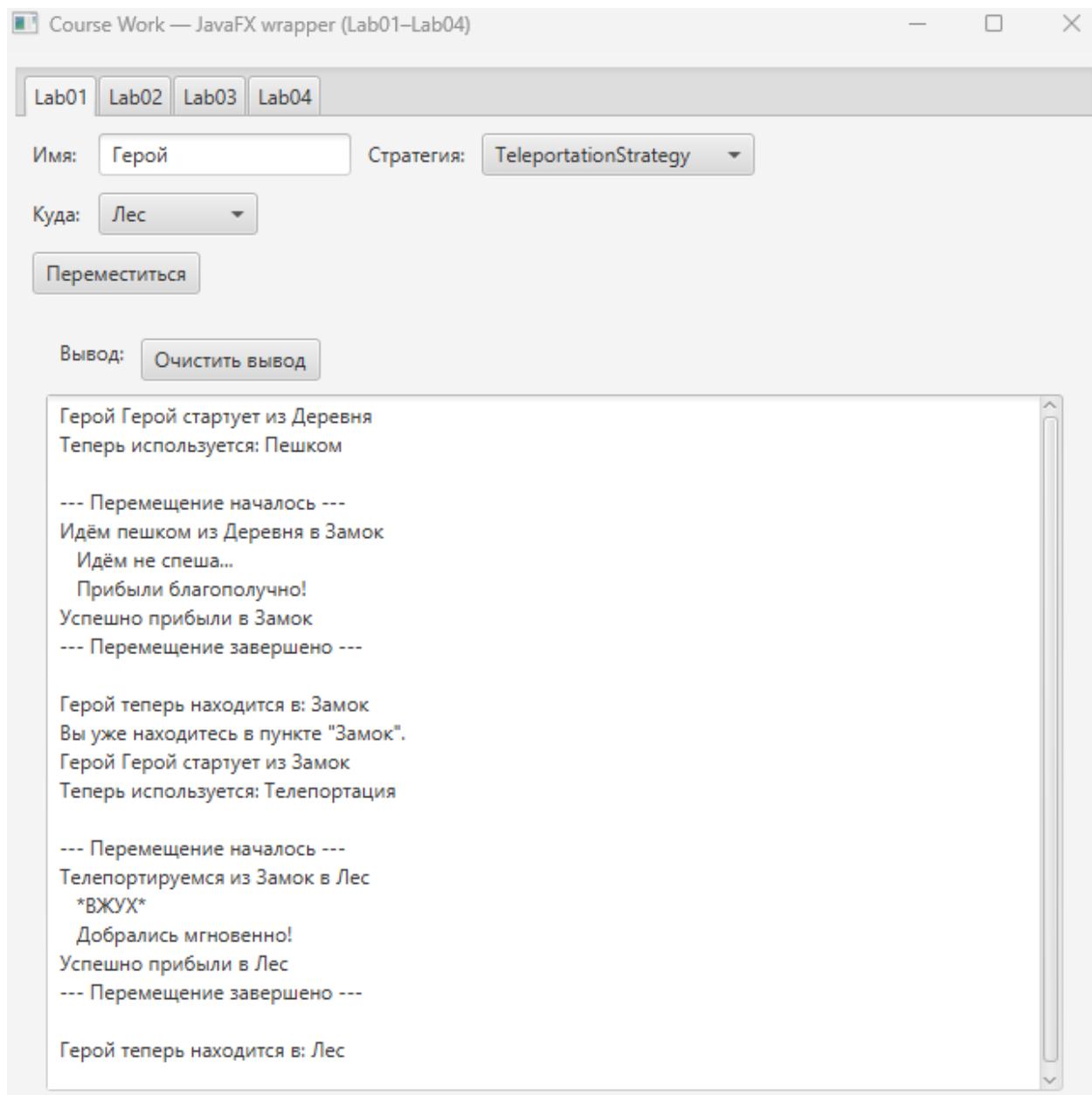


Рисунок 6 – Работа 1-го задания

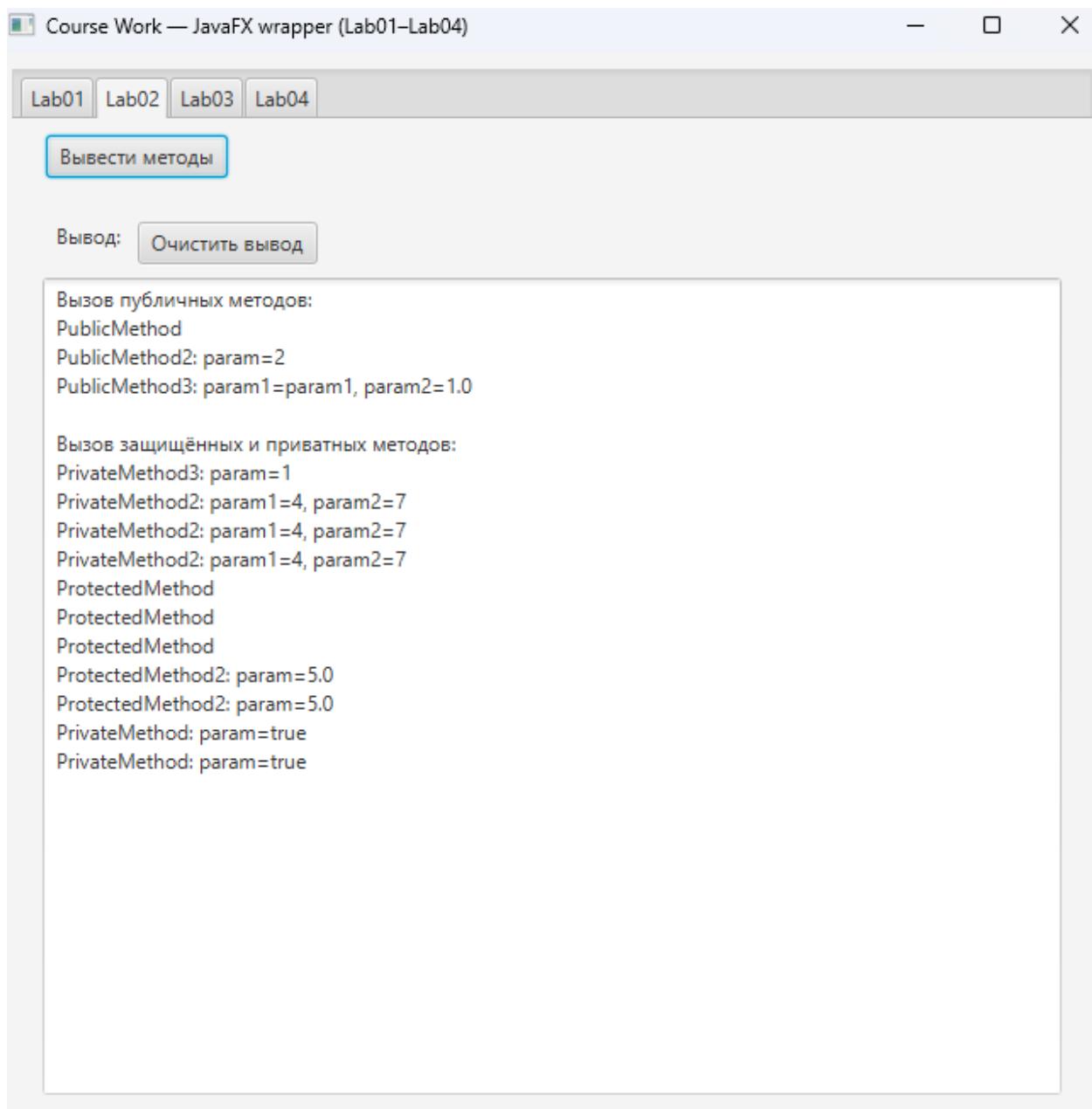


Рисунок 7 – Работа 2-го задания

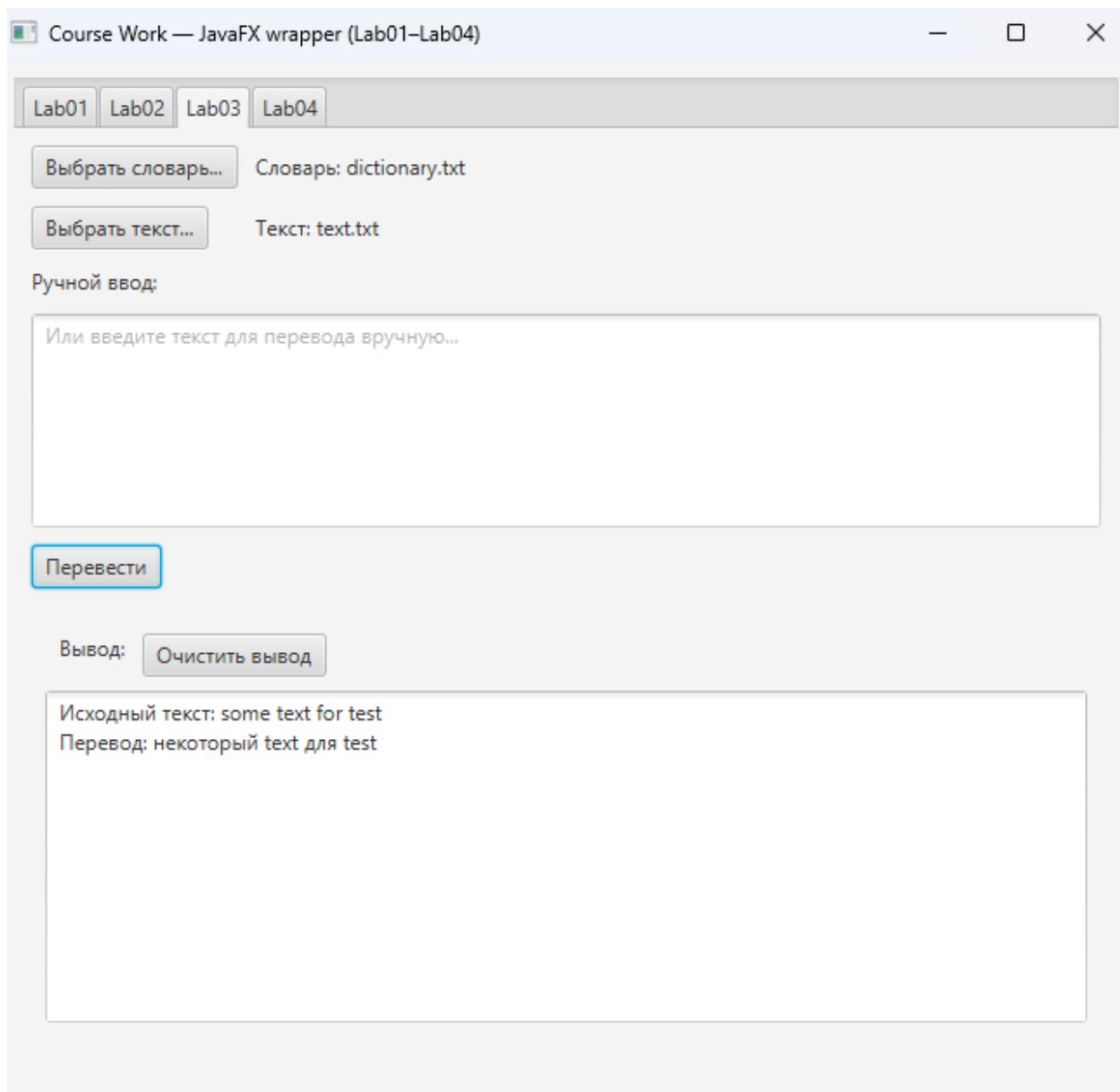


Рисунок 8 – Работа 3-го задания

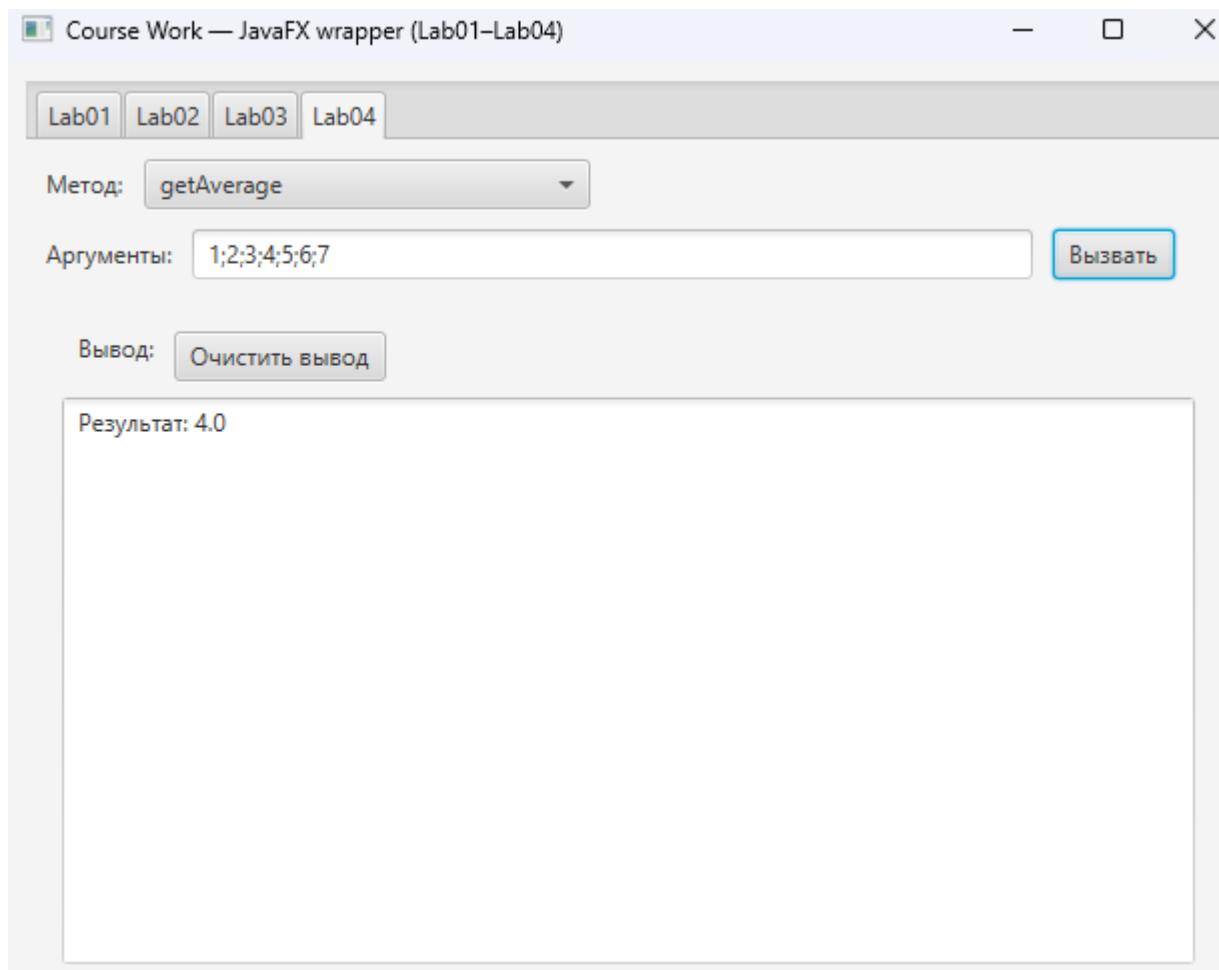


Рисунок 8 – Работа 4-го задания