Министерство науки и высшего образования Российской Федерации

Санкт-Петербургский политехнический университет Петра Великого (СПбПУ). Институт компьютерных наук и Кибербезопасности

**КУРСОВАЯ РАБОТА**

Тема: «Разработка приложения с графическим интерфейсом»

По дисциплине: «Объектно-ориентированное программирование»

Выполнил студент в5130904/30322 Морозов Н.Д.

Руководитель А.П. Маслаков

Санкт-Петербург

2025

Оглавление

[Введение 3](#_Toc209640811)

[Диаграмма классов 3](#_Toc209640812)

[Перечень выполненных работ 4](#_Toc209640813)

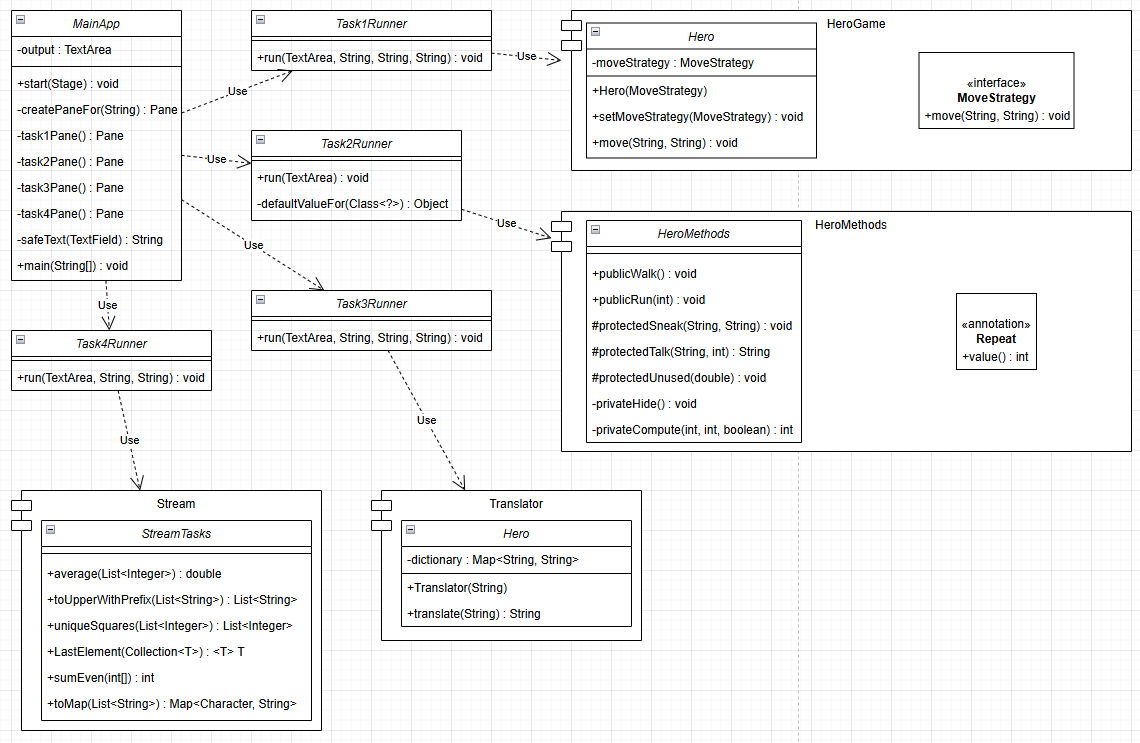
[Заключение 5](#_Toc209640814)

# **Введение**

Целью данной работы является разработка графического приложения-обёртки для четырёх ранее выполненных лабораторных работ. Каждая лабораторная реализует отдельный функционал, а курсовая объединит их в единую систему с удобным интерфейсом.

При разработке будут использованы основные принципы ООП, обеспечивающие структурированность, расширяемость и удобство сопровождения программы. Реализация будет выполнена на языке Java с использованием библиотеки JavaFX для построения графического интерфейса.

# **Диаграмма классов**

****

# **Перечень выполненных работ**

В ходе выполнения курсовой работы было разработано графическое приложение с использованием платформы JavaFX. Оно обеспечивает запуск и тестирование различных задач через удобный графический интерфейс. Главный класс **MainApp** отвечает за построение интерфейса и организацию взаимодействия пользователя с реализованными задачами.

Для каждой задачи был создан отдельный модуль-"запускатель" (Runner-класс), который инкапсулирует логику выполнения и обеспечивает вызовы функциональности подключаемых субмодулей. Такой подход позволил реализовать гибкую архитектуру, в которой основной интерфейс приложения отделён от внутренней реализации задач.

В приложении реализовано четыре независимые задачи. Первая задача связана с использованием паттерна «Стратегия» для управления движением персонажа **Hero** в модуле **HeroGame**. Вторая задача продемонстрировала работу с рефлексией и аннотациями: класс **HeroMethods** содержит методы с различным уровнем доступа, а аннотация **Repeat** используется для задания количества вызовов метода. Третья задача обеспечивает взаимодействие с модулем **Translator**, который выполняет перевод текста на основе словаря. Четвёртая задача демонстрирует работу с коллекциями и числовыми данными через модуль **Stream**, реализованный с использованием Java Stream API.

Все перечисленные компоненты объединены в единое приложение, где их вызов осуществляется через графический интерфейс. Такой подход позволил реализовать модульность и наглядно продемонстрировать возможности работы с ключевыми возможностями языка Java: стратегиями, аннотациями, рефлексией, файловым вводом-выводом, потоковыми операциями.

# **Заключение**

В результате выполнения курсовой работы было создано законченное программное приложение с графическим интерфейсом, позволяющее в интерактивной форме запускать и тестировать разработанные задачи. Программа демонстрирует применение различных возможностей языка Java, включая объектно-ориентированное программирование, использование паттернов проектирования, работу с рефлексией и аннотациями, а также применение потокового API для обработки коллекций.

Реализованная архитектура приложения является модульной и расширяемой: каждая задача выполнена в отдельном классе-«запускателе», что позволяет легко добавлять новые функции без изменения основной структуры. Подключение внешних субмодулей делает проект более гибким и приближает его к практическим сценариям разработки программного обеспечения.

Таким образом, поставленные в рамках курсовой работы цели были достигнуты: разработано приложение, изучены и реализованы ключевые возможности языка Java, а также закреплены навыки построения архитектуры программного обеспечения и документирования результатов в виде UML-диаграмм и структурированного отчёта.