

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №4**  
**по дисциплине «Объектно-ориентированное программирование»**  
**Тема: Уровни абстракции, управление игроком**

Студент гр. 1303

\_\_\_\_\_

Коренев Д.А.

Преподаватель

\_\_\_\_\_

Жангиров Т.Р.

Санкт-Петербург

2022

### **Цель работы.**

Реализовать набор классов отвечающих за считывание команд пользователя, обрабатывающих их и изменяющих состояния программы.

### **Задание.**

Реализовать набор классов отвечающих за считывание команд пользователя, обрабатывающих их и изменяющих состояния программы (начать новую игру, завершить игру, сохраниться, управление игроком, и т.д.). Команды/клавиши, определяющие управление должны считываться из файла.

#### **Требования:**

- Реализован класс/набор классов обрабатывающие команды.
- Управление задается из файла (определяет какая команда/нажатие клавиши отвечает за управление, например, w - вверх, s - вниз, и т.д.).
- Реализованные классы позволяют добавить новый способ ввода команд без изменения существующего кода (например, получать команды из файла или по сети). По умолчанию, управление из терминала или через GUI, другие способы реализовывать не надо, но должна быть такая возможность.
- Из метода, считывающего команду, не должно быть “прямого” управления игроком.

#### **Примечания:**

- Для реализации управления можно использовать цепочку обязанностей, команду, посредника, декоратор, мост, фасад.

### **Выполнение работы.**

Для выполнения лабораторной работы созданы и дополнены классы, отвечающие за считывание, преобразование и обработку данных пользователя.

Интерфейс Configuration служит описанием всех объектов задающих настройки управления. Имеет чистый виртуальный метод `getSettings()`.

Класс `FileConfig` реализует вышеописанный интерфейс. Во время создания экземпляра открывается файл конфигурации (бросается ошибка, если его нельзя открыть), считываются данные, формируется конфигурация. Метод `getSettings()` возвращает конфигурацию. Процесс формирования конфигурации гарантируется корректные настройки.

Интерфейс `InputReader` служит описанием всех объектов считывающих данные от пользователя. Имеет несколько чистых виртуальных методов, необходимых для инициализации всех требуемых объектов и полей.

Класс `InputKeyboard` реализует интерфейс `InputReader`. Все виртуальные методы интерфейса определены на считывание(вывод) данных в(из) консоль(консоли).

Класс `CommandReader` отвечает за запрос у пользователя данных. Имеет указатель на `InputReader`, т. е. может запрашивать у пользователя данные независимо от того какой объект хранится по данной ссылке, т. к. он реализует `InputReader`.

Полученные от `CommandReader` данные объект класса `Mediator` передает объекту `Controller`. Он имеет метод `setConfig`, изменение настроек взаимодействия пользователя с игрой. Настройки хранятся в виде хэш карты, где ключ — символ, значение — действие. Таким образом приняв входные данные от игрока, они формируются в команду. Если действие служебное (выход или помощь), то вызываются требуемые для них методы, иначе (это является передвижение) делегирует взаимодействие на эту команду модели.

### **Тестирование.**

Тестирование программы: создание и обработка файла конфигурации.

```

1      UP s
2      UP r
3      DOWN r
4      DOWN w
5      LEFT w
6      RIGHT a
7      EXIT n
8      BRED bred
9      LOGGING l
10     LOGGING e
11     DOWN ]
12     HELP h

```

Рисунок 1 Пример  
файла конфигурации

```

h
Your config
a:RIGHT
h:HELP
n:EXIT
r:DOWN
s:UP
w:LEFT

```

Рисунок 2  
Интерпритация  
данных файла  
конфигурации

### **UML-диаграмма межклассовых отношений.**

UML-диаграмма межклассовых отношений, созданных и измененных во время выполнения лабораторной работы, представлена на рисунке 3.

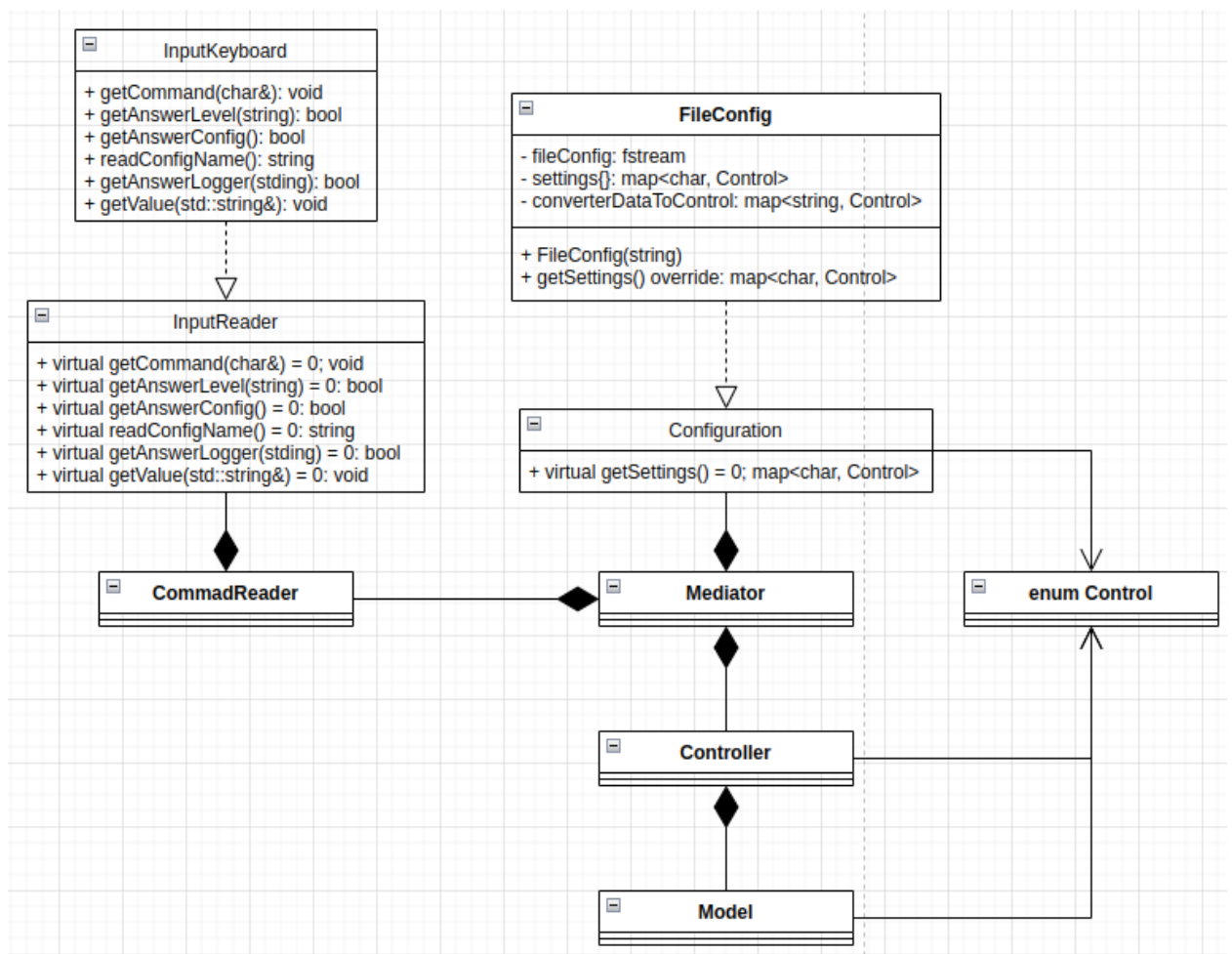


Рисунок 3 UML диаграмма межклассовых отношений

## Выводы.

Реализован набор классов отвечающих за считывание команд пользователя, обрабатывающих их и изменяющих состояния программы.