

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

ОТЧЕТ
по лабораторной работе №4
по дисциплине «ООП»
Тема: Создание класса управления игрой.

Студент гр. 9304

Попов Д.С

Преподаватель

Размочаева Н.В.

Санкт-Петербург

2020

Цель работы.

Создать и внедрить класс управления игрой.

Задание.

Создать класс игры, через который пользователь взаимодействует с игрой. Управление игроком, начало новой игры, завершение игры. Могут быть созданы дополнительные необходимые классы, которые отвечают отдельно за перемещение, создание игры и т.д. Но пользователь должен взаимодействовать через интерфейс одного класса.

Обязательные требования:

- Создан класс управления игрой
- Взаимодействие сохраняет инвариант

Дополнительные требования:

- Пользователь взаимодействует с использованием паттерна

Команды

- Взаимодействие с компонентами происходит через паттерн

Фасад

Выполнение работы.

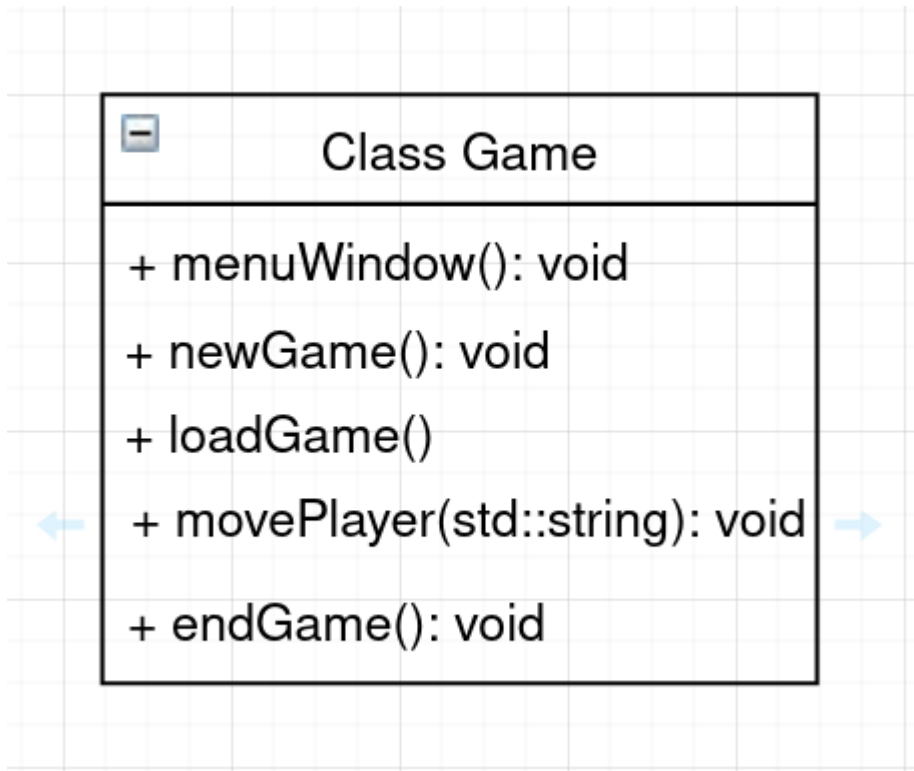


Рис. 1 - UML — диаграмма.

Создан класс *Game*, который осуществляет контроль за ходом выполнения игры. Данный класс содержит публичные методы :

- *menuWindow()* – отвечает за создание окна меню игры, в котором можно выбрать один из трех возможных вариантов(Новая игра, Загрузка, Выход).
- *newGame()* – запускает основной цикл игры и создает окно для ее отображения.
- *loadGame()* – на данный момент этот метод не используется, но понадобится для загрузки игры.
- *endGame()* – завершает основной процесс игры и закрывает окно.

- *playerMove(std::string route)* – передвигает игрока на одну ячейку в заданном строкой направлений.

Выводы.

Мы реализовали класс логирования, перегрузили операторы вывода в файловый поток, ознакомились с идиомой RAII.