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

ОТЧЕТ

по лабораторной работе № 1

по дисциплине «Объектно ориентированное программирование»

Тема: Создание классов, конструкторов и методов класса

Студент гр. 0383 _____ Коротков А.В.

Преподаватели _____ Жангиров Т.Р.

Санкт-Петербург
2021

Цель работы.

Реализовать классы “клетка” и “поле”. Для класса клетки необходимо сделать интерфейс, а для поля конструкторы копирования и перемещения, а также соответствующие операторы.

Задание.

Игровое поле представляет из себя прямоугольную плоскость разбитую на клетки. На поле на клетках в дальнейшем будут располагаться игрок, враги, элементы взаимодействия. Клетка может быть проходимой или непроходимой, в случае непроходимой клетки, на ней ничего не может располагаться. На поле должны быть две особые клетки: вход и выход. В дальнейшем игрок будет появляться на клетке входа, а затем выполнив определенный набор задач дойти до выхода.

При реализации класса поля запрещено использовать контейнеры из stl

Требования:

- Реализовать класс поля, который хранит набор клеток в виде двумерного массива.
- Реализовать класс клетки, которая хранит информацию о ее состоянии, а также того, что на ней находится.
- Создать интерфейс элемента клетки.
- Обеспечить появление клеток входа и выхода на поле. Данные клетки не должны быть появляться рядом.
- Для класса поля реализовать конструкторы копирования и перемещения, а также соответствующие операторы.
- Гарантировать отсутствие утечки памяти.

Потенциальные паттерны проектирования, которые можно использовать:

Итератор (Iterator) - обход поля по клеткам и получение косвенного доступа к ним

Строитель (Builder) - предварительное конструирование поля с необходимым параметрами. Например, предварительно задать кол-во непроходимых клеток и алгоритм их расположения

Выполнение работы.

Был реализован класс *Field*: в классе хранится поле из клеток класса *Tile* в виде двумерного массива. У класса *Field* есть конструкторы копирования и присваивания, а также соответствующие операторы.

Для вывода графики была использована библиотека SFML.

Класс *Tile* хранит в себе информацию о том, что находится на клетке, её тип, а также *sf::Sprite sprite* для отрисовки клетки. У класса Tile реализован интерфейс.

UML диаграмму см. в приложении А.

Выводы.

В ходе выполнения лабораторной работы была написана программа, реализующая классы клетки и поля, а также конструкторы копирования и присваивания и соответствующие операторы.

Приложение А

UML-диаграмма