

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 классов

Студент гр. 0303

Болкунов В.О.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург 2021

Цель работы.

Реализовать заданные классы, для главного класса (поля) реализовать конструкторы копирования, перемещения и соответствующие операторы присвоения.

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 `std`

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.

Основные теоретические положения.

Класс — в объектно-ориентированном программировании, представляет собой шаблон для создания объектов, обеспечивающий начальные значения состояний: инициализация полей и реализация поведения функций или методов. Класс является ключевым понятием в ООП.

Конструктор класса — специальный блок инструкций, вызываемый при создании объекта.

Выполнение работы.

Структура **Vec2D** — двумерный целочисленный вектор с перегруженными арифметическими операторами (работают как алгебраические операции с векторами).

Класс **Entity** — интерфейс игрового объекта (сущности). Содержит виртуальный метод **interact** для взаимодействия с другими сущностями.

Класс **AbstractCell** — абстрактный класс для представления клетки, содержит метод перемещения объекта на клетку и взаимодействия объекта с объектом на клетке.

Наследники **AbstractCell**:

Wall — непроходимая клетка

Cell — стандартная клетка, от которой наследуются вход и выход: **Entrance**, **Exit**

Класс **Field** — класс поля, хранит в себе двумерный динамический массив указателей на клетки **AbstractCell** *******, содержит конструктор копирования, перемещения и соответствующие операторы присваивания. Для генерации поля используются методы класса **FieldBuilder**.

Для получения и вывода информации о клетках и поле используются классы [CellView](#), [FieldView](#) и [ConsoleFieldView](#).

Разработанный программный код см. в директории */game_lib*.

Разработанную диаграмму классов UML см. в *Lab1_UML.pdf*.

Тестирование.

Разработанные тесты см. в директории */tests*.

Выводы.

Был реализован набор необходимых классов, созданы конструкторы, конструкторы копирования и перемещения для безопасности работы динамического массива.