

**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. 0381

Дзаппала Д.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2021

### **Цель работы.**

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 эл-та клетки. Обеспечить отсутствие утечки памяти.

### **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.

#### **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 необходимыми параметрами. Например, предварительно задать кол-во непроходимых клеток и алгоритм их расположения*

## **Выполнение работы.**

### **Класс FieldCage.**

Класс представляет клетку поля. В private области содержатся свойства клетки: x координату, y координату, тип клетки и ее статус. В header файле клетки также определены три класса перечислений: Status, Type и CageSize. Status содержит два эл-та: NOT\_AVAILABLE и AVAILABLE. Статус клетки обозначает свободна ли эта клетка, то есть можно ли на нее перейти. Type содержит 4 эл-та: Exit, Common, Entry, Wall. CageSize содержит эл-т Size, то есть размер пикселей для клетки. В классе объявлены геттеры/сеттеры для всех полей класса.

### **Класс Field.**

Класс предоставляет поле игры. Объект хранит в себе двумерный массив объектов клетки cageArray, статические поля height и width. В классе реализованы конструкторы: конструктор по умолчанию (выделяет память под массив), конструктор копирования, перемещением, конструктор, который принимает объект класса CreateField, после чего делаем swap массивов классов. Также, в классе Field есть вложенный класс iterator — поведенческий паттерн проектирования, который дает возможность последовательно обходить эл-ты массива клеток.

### **Класс CreateField.**

Класс нужен только как аргумент для конструктора объекта Field. Он принимает в конструктор константную строку с названием файла, в котором содержится карта поля. Класс содержит приватное поле cages — двумерный массива клеток. В конструкторе выделяется память под массив, открывается файл, считываются типы клеток, которые записываются в клетки.

### **Класс FieldView.**

Класс для отрисовки или вывода в консоль клеток. В header файле есть класс перечисления OutMode с двумя эл-тами: CLI и GUI. Класс содержит два поля: метод вывода (OutMode), ассоциативный контейнер map, с парой ключ-значение Type-std::string. В случае вывода поля графическим способом, нам нужны картинки для спрайтов. В классе есть метод loadPictures, который принимает std::initializer\_list<std::string> с упорядоченными по типу клетки константными строками с путем до картинки. Для вывода поля перегружен оператор (<<) выходного потока, который принимает аргументом ссылку на объект Field(поле).

В проекте используется графическая библиотека SFML.

### **Класс IObject.**

Класс, представляющий интерфейс объекта, который будет находиться на поле на клетке. Клетка будет иметь поле типа указатель на Object. Класс содержит виртуальный метод Name.

### **Выводы.**

Были реализованы классы для представления клетки и поля, которое и содержит эти клетки. В итоге получилось игровое поле, которое отрисовывается с помощью библиотеки SFML.

## ПРИЛОЖЕНИЕ А

### UML ДИАГРАММА КЛАССОВ

