

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

Мыратгелдиев А. М.

Преподаватель

Жангиров Т. Р.

Санкт-Петербург

2021

Цель работы.

Научиться реализовывать классы, а также конструкторы и деструктор, и методы классов.

Задание.

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.

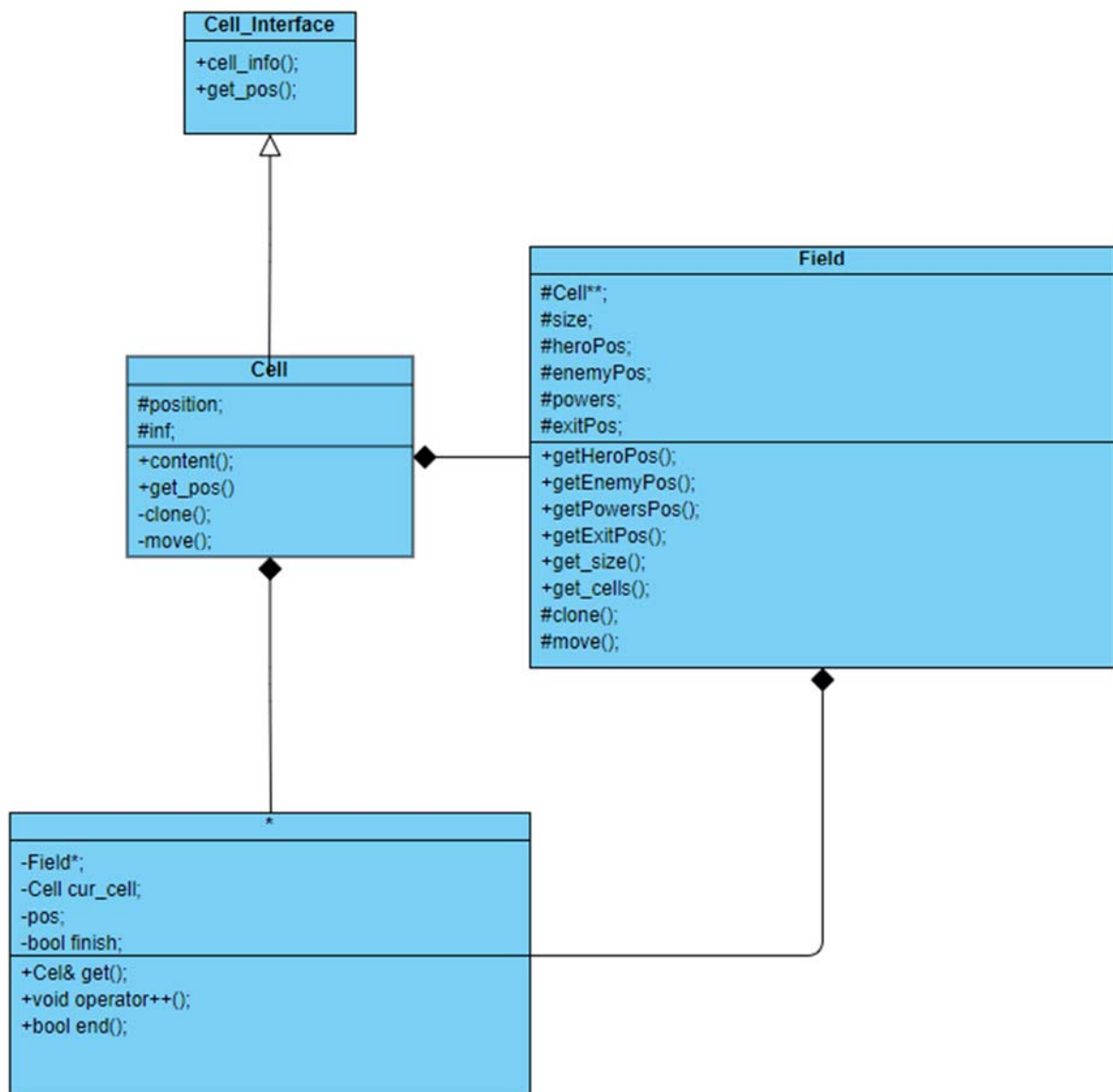
Был создан абстрактный класс *Cell_Interface*(интерфейс клетки), с чисто виртуальными методами *content()* и *get_pos()*. От него наследуется класс *Cell*(клетка). Класс клетка имеет поля *position*, для хранения позиции клетки, *inf*, для хранения информации о том, что на ней находится. Во избежание дублирования кода в конструкторах копирования и перемещения, и в перегрузке

операторов присваивания с копированием и присваивания с перемещением, были написаны методы *clone()* и *move()*.

Класс *Field* хранит двумерный массив клеток(*Cell*), ее размер, начальные расположения героя, врагов, вещей, а также координаты клетки выхода . Для класса были перегружены операторы: присваивания(оба), индексирования. Для этого также были написаны метода геттеры.

Класс *Iterator* предоставляет способ последовательного доступа ко всем элементам поля(клеткам). Он получает указатель на поле и обходит все его клетки.

UML-диаграмма классов.



Выводы.

Были исследованы способы реализации наследуемых классов, а также их конструкторов и деструкторов. Были реализованы классы *Cell*, *Field*, *Iterator*. Для хранения парных значений (таких как координата, размер) был использован класс *sf::Vector2i()*, из библиотеки SFML. Задание было выполнено при помощи основных управляющих конструкций языка.