

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

**ОТЧЕТ**  
**по лабораторной работе №1**  
**по дисциплине «ООП»**  
**Тема: Создание игрового поля**

Студент гр. 9303

\_\_\_\_\_

Дюков В.А.

Преподаватель

\_\_\_\_\_

Размочаева Н.А.

Санкт-Петербург

2020

### **Цель работы.**

Научиться использовать классы в языке C++ , изучить основные методы работы с классами, а также изучить способы взаимодействия классов.

### **Задание.**

Написать класс игрового поля, которое представляет из себя прямоугольник (двумерный массив). Для каждого элемента поля должен быть создан класс клетки. Клетка должна отображать, является ли она проходимой, а также информацию о том, что на ней находится. Также, на поле должны быть две особые клетки: вход и выход.

#### Обязательные требования:

- Реализован класс поля
- Реализован класс клетки
- Для класса поля написаны конструкторы копирования и перемещения, а также операторы присваивания и перемещения
- Поле сохраняет инвариант - из любой клетки можно провести путь до любой другой
- Гарантированно отсутствует утечки памяти

#### Дополнительные требования:

- Поле создается с использованием паттерна **Синглтон**
- Для обхода по полю используется паттерн **Итератор**. Итератор должен быть совместим со стандартной библиотекой.

## Выполнение работы.

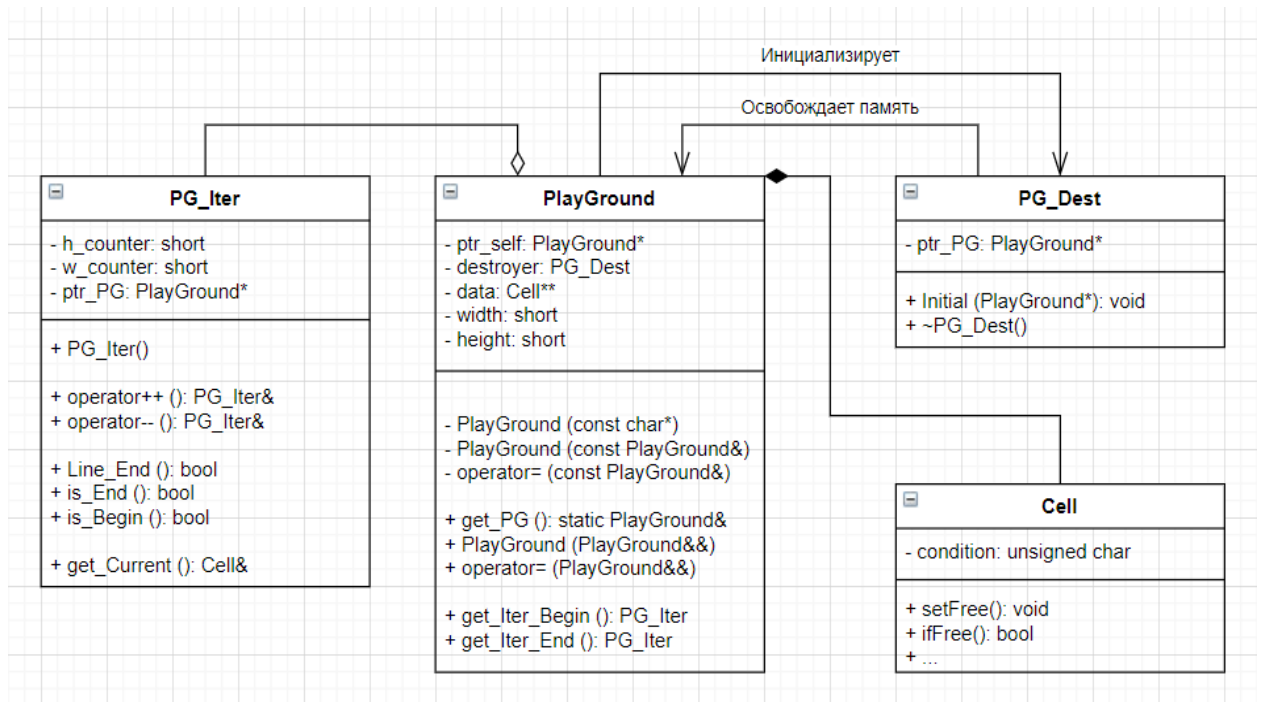


Рисунок 1. UML – диаграмма классов

В ходе работы был создан класс PlayGround, имеющий поля:

- ptr\_self – статическая переменная, указатель на объект класса PlayGround.
- destroyer – статическая переменная, хранящая объект класса PG\_Dest.
- data – переменная, указатель на двумерный массив объектов класса Cell.
- width – переменная, хранящая ширину двумерного массива.
- height – переменная, хранящая высоту двумерного массива.

Класс PG\_Dest имеет следующие поля:

- ptr\_PG – указатель на объект класса PlayGround.

Класс Cell имеет следующие поля:

- condition – переменная, содержащая состояние в котором находится сама клетка.

Класс PG\_Iter имеет следующие поля:

- h\_counter – переменная, содержащая данные о вертикальной координате на которой в данный момент находится итератор.
- w\_counter – переменная, содержащая данные о вертикальной координате на которой в данный момент находится итератор.
- ptr\_PG – переменная, содержащая состояние в котором находится сама клетка.

Чтобы сделать объект поля единственным в программе, был применён паттерн Синглтон. Поэтому конструктор, конструктор копирования и оператор присваивания находятся в приватном праве доступа, а объект этого класса можно получить при вызове метода get\_PG.

Конструктор класса PlayGround принимает на вход строку — адрес файла « схемы » поля. Конструктор сам определяет высоту и ширину поля и инициализирует переменную data двумерным массивом из объектов класса Cell. Если высота и ширина меньше допустимого минимума, то переменная data не инициализируется. Также конструктор инициализирует переменную destroyer.

Класс PG\_Dest создан дружественным к классу PlayGround. Его задача состоит в очищении памяти, занятой в результате работы объекта класса PlayGround.

Класс Cell содержит методы позволяющие получить информацию о состоянии клетки поля. Данные о состоянии содержатся в переменной condition.

Для прохождения по клеткам поля был создан класс PG\_Iter. Итератор обходит двумерный массив привычным образом - по порядку в соответствии с индексами. Метод get\_Current возвращает ссылку на текущую клетку поля, перегруженный оператор инкремента переходит к следующей, а

перегруженный оператор дикремента переходит к предыдущей клетке. С помощью методов класса `PlayGround` можно получить объект класса `PG_Iter` с положением либо в начале, либо в конце поля.

## Тестирование.

Ниже представлены изображения с тестированием программы.

*Рисунок 2.*

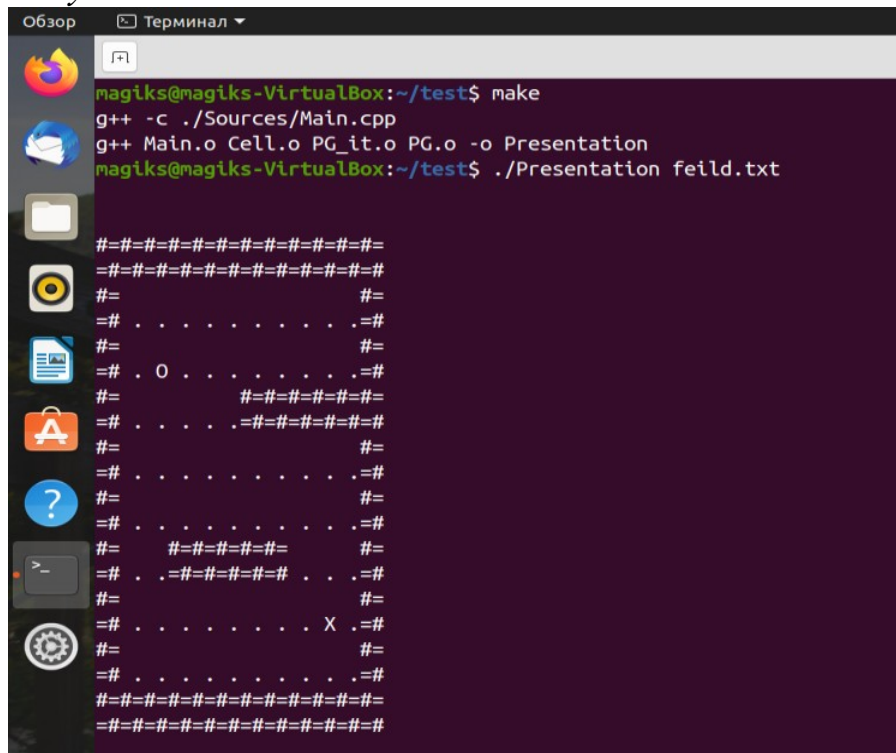
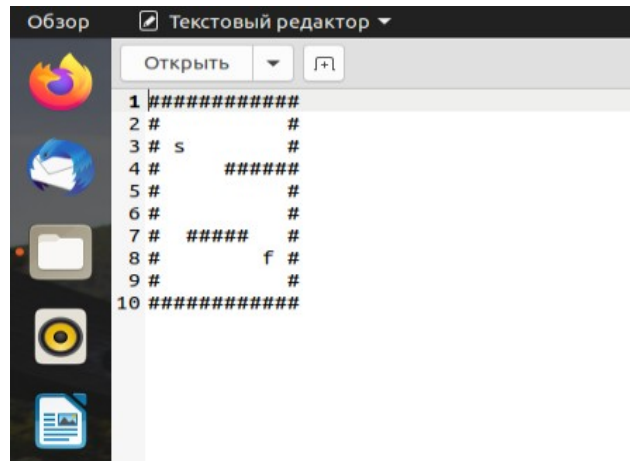


Рисунок 2 — изображение вывода поля с помощью перегруженного оператора вывода класса *ostream*:

```
std::ostream& operator<< (std::ostream&, const Playground&).
```

Рисунок 3 — изображение файла «схемы» с которого идёт считывание конструкции поля.

Рисунок 3.



### Выводы.

Был реализован класс поля, для которого были изучены паттерны Синглтон и Итератор. Для реализации паттерна Синглтон в классе *PlayGround* конструктор копирования и оператор копирования были помещены в область видимости `private`. Итератор представляет собой класс для прохождения по массиву из клеток в классе *PlayGround*.