**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе №2**

**по дисциплине «Объектно-ориентированное программирование»**

Тема: **Конструкторы и деструкторы**

|  |  |  |
| --- | --- | --- |
| Студент гр. 2381 |  | Рыжиков И.С. |
| Преподаватель |  | Жангиров Т.Р. |

Санкт-Петербург

2023

## Цель работы

Изучить методы работы с классами. Реализовать работу конструктора и деструктора в С++, а также конструкторы перемещения и конструкторы копирования.

## Задание

а) Создать класс клетки игрового поля. Клетка игрового поля может быть проходимой или нет, тем самым определяя возможность игрока встать на эту клетку. Возможность задать проходимость клетки должна быть реализована через конструктор и через метод клетки. В будущем в клетке будет храниться указатель на интерфейс события.

б) Создать класс игрового поля. Игровое поле представляет собой прямоугольник из клеток (двумерный массив). **В учебных целях, клетки хранятся как чистый массив на указателях (использовать контейнеры stl запрещено в этой лаб. работе)**. Размер поля передается в конструктор поля, в котором динамически выделяется память под массив клеток. Также должна быть возможность вызвать конструктор поля без аргументов. Так как происходит выделение память, то необходимо реализовать деструктор в котором будет происходить очистка память.

Также добавить в игровое поле добавьте информацию о входе (где в начале появляется игрок) и выходе (куда игрок должен дойти)

в) В класс управления игрока добавить взаимодействия с полем. При перемещении игрока должна быть проверка на проходимость клетки, если клетка непроходима, то перемещение не должно производиться.

Примечания:

* Так как в клетке будет храниться указатель, то  при копировании и перемещении должен быть предусмотрен механизм копирования объекта хранящегося по указателю.
* Убедитесь, что присутствует проверка контроля размера поля, чтобы его нельзя было сделать слишком маленьким или с отрицательными размерами.
* В конструкторе перемещения и соответствующем ему операторе присваивания не должно происходить никакого копирования данных.
* Через класс поля должен быть доступ к каждой индивидуальной клетке. Создавать метод, который возвращает указатель на весь массив или указатель на каждую клетку, плохая практика, так как появляется возможность очистки памяти.

## Выполнение работы

Созданы два класс FieldCell — клетка поля и *Field —* поле, которое хранит двумерный массив клеток.

Добавлена структура Coordinate для хранения пары координат. Теперь Player хранит свои координаты в данной структуре. Для Player добавлены методы getCoordinate и setCoordinate.

Класс PlayerController теперь также принимает Field при создании. Добавлен метод changeField для смены поля. Добавлен метод canMove для определения может ли игрок передвинуться на указанную клетку.

### Класс *FieldCell*

Класс *клетки поля*. Может быть проходимой или нет. Так же будет хранить указатель на *Event*, который будет реализован в дальнейшем.

Имеет методы isOccupied, setOccupied, removeOccupied.

Метод isOccupied возвращает true, если *клетка занята* и false, если свободна.

Метод setOccupied добавляет метку *«занята».*

Метод removeOccupied *удаляет* у клетки метку «занята», делая её *доступной для перемещения*.

### Класс Field

Класс поля. Создает и хранит в себе двумерный массив клеток. А также координаты старта и финиша.

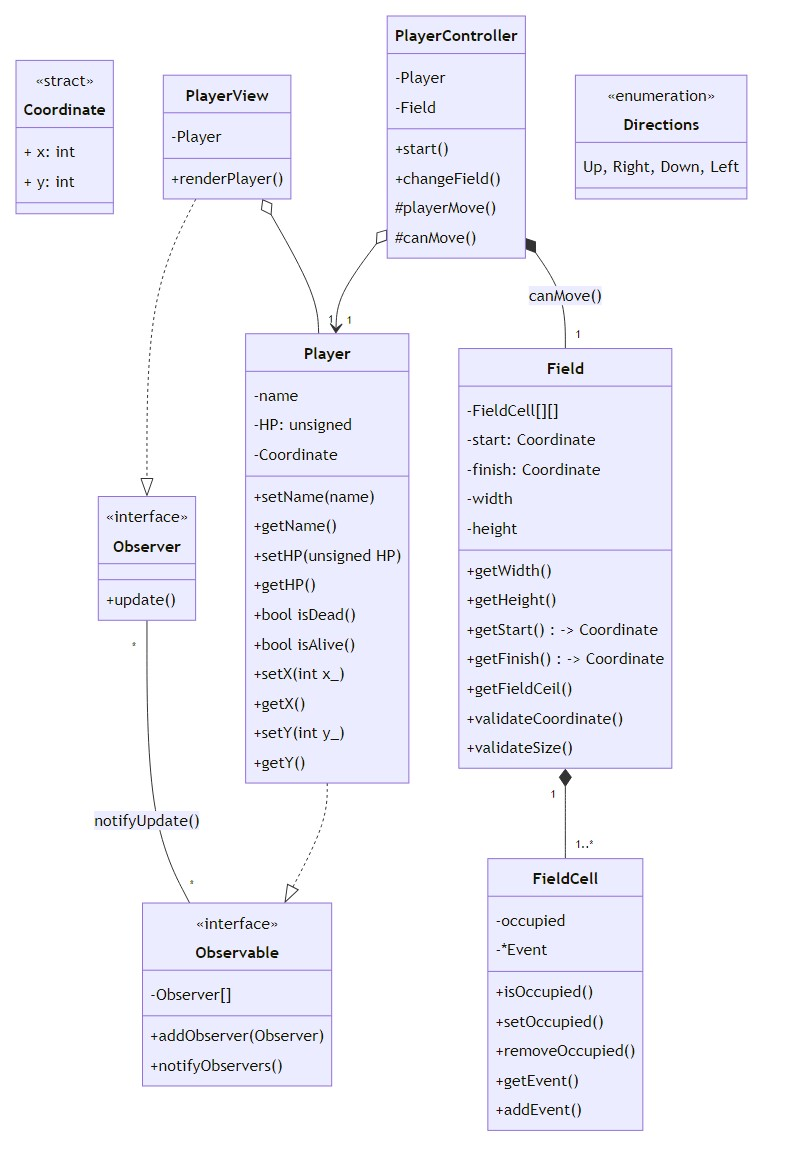
Методы getWidth и getHeight возвращают высоту и ширину поля.

Методы getStart и getFinish возвращают координаты старта и финиша.

Методы getFieldCeil возращает копию клетки поля.

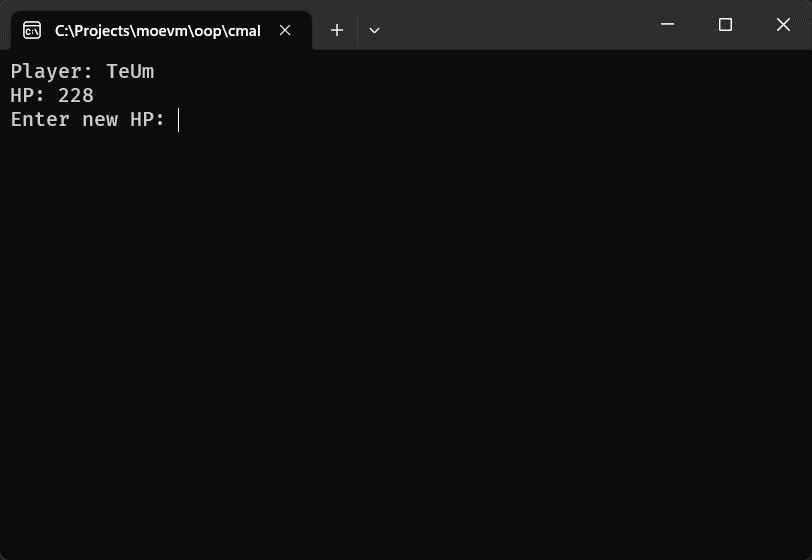
Метод validateSize проверяет является ли находятся ли размеры поля в допустимом диапазоне.

Метод validateCoordinate проверяет входят ли координаты клетки во внутрь поля.



1. Диаграмма классов (полная)

## Работа программы



1. Пример работы программы в терминале

## Выводы

Был реализован класс поля и клетки. Реализованы конструкторы копирования, перемещения, а также деструктор.