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

Студент гр. 1384 _____ Прошичев А.В.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург
2022

Цель работы.

Научиться реализовывать классы, их поля и методы, в том числе и конструкторы по умолчанию, копирования, перемещения. Научиться моделировать UML-диаграммы.

Задание.

Реализовать прямоугольное игровое поле, состоящее из клеток. Клетка - элемент поля, который может быть проходим или нет (определяет, куда может стать игрок), а также содержит какое-либо событие, которое срабатывает, когда игрок становится на клетку. Для игрового поля при создании должна быть возможность установить размер (количество клеток по вертикали и горизонтали). Игровое поле должно быть зациклено по вертикали и горизонтали, то есть если игрок находится на правой границе и идет вправо, то он оказывается на левой границе (аналогично для всех краев поля).

Реализовать класс игрока. Игрок – сущность, контролируемая пользователем. Игрок должен иметь свой набор характеристик и различный набор действий (например, разные способы перемещения, попытка избежать событие, и так далее).

Требования:

- Реализован класс игрового поля
- Для игрового поля реализован конструктор с возможностью задать размер и конструктор по умолчанию (то есть конструктор, который можно вызвать без аргументов)
- Реализован класс интерфейс события (в данной лабораторной это может быть пустой абстрактный класс)
- Реализован класс клетки с конструктором, позволяющим задать ей начальные параметры.
- Для клетки реализованы методы реагирования на то, что игрок перешел на клетку.
- Для клетки реализованы методы, позволяющие заменять событие. (То есть клетка в ходе игры может динамически меняться)
- Реализованы конструкторы копирования и перемещения, и соответствующие им операторы присваивания для игрового поля и при необходимости клетки
- Реализован класс игрока минимум с 3 характеристиками. И соответствующие ему конструкторы.
- Реализовано перемещение игрока по полю с проверкой допустимости на переход по клеткам.

Примечания:

- При написании конструкторов учитывайте, что события должны храниться по указателю для соблюдения полиморфизма
- Для управления игроком можно использовать медиатор, команду, цепочку обязанностей

Выполнение работы.

В начале выполнения программа выделяет память под классы, управляющие работой игры: Commander, CommandReader и Game.

Commander – класс, хранящий команды для управления сущностями игры: полем Field, выводом поля FieldViewer. Имеет доступ к значениям характеристик игрока Player, поля Field, клетки Cell и события Event.

Класс Field – класс, содержащее в себе поле клеток Cell и методы контролирования позиции игрока Player. Также игровая область хранит в себе поле игрока Player, и всё взаимодействие с игроком происходит через него.

Класс Cell – класс, содержащий в себе указатель на интерфейс события Event. А также имеет своё поле, показывающее наличие препятствий на клетке, mWall.

Класс Event – интерфейс события, от которого наследуются любое конкретное событие.

Класс Player – класс игрока, содержащий 3 характеристики: жажду, голод и здоровье.

Класс CellViewer – класс, который определяет вывод клеток, ориентируясь на их параметры.

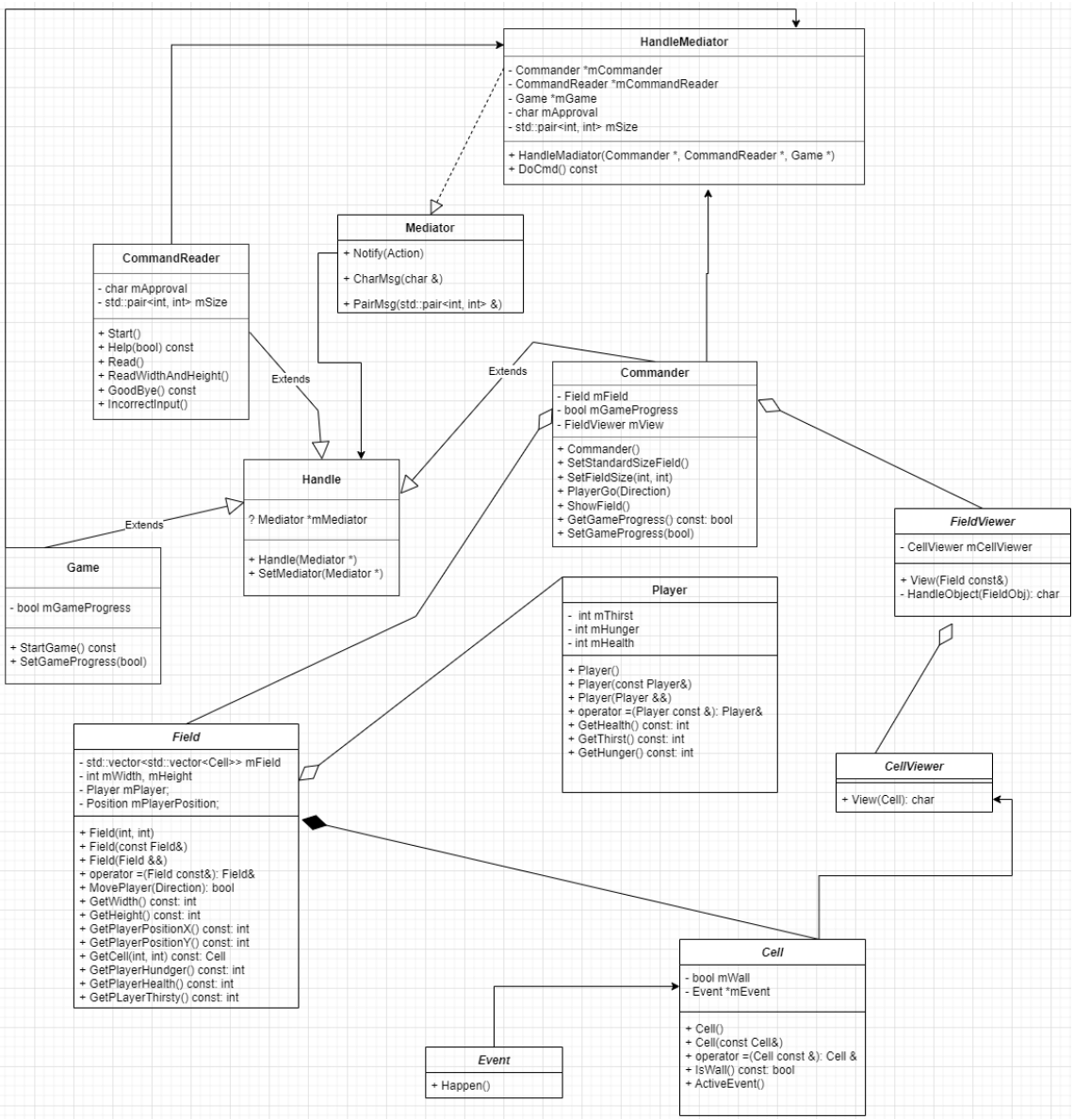
Класс FieldViewer – класс, прорисовывающие поле. Хранит в себе класс CellViewer. При прорисовывании клетки обращается именно к нему. Сам прорисовывает игрок и границы карты.

CommandReader – класс, реализующий общение с игроком. Выводит текстовые сообщения, чтобы сориентировать пользователя. Записывает введённый команды и отправляет Handle Mediator, который и регулирует дальнейшую работу игры через общение с классом Commander.

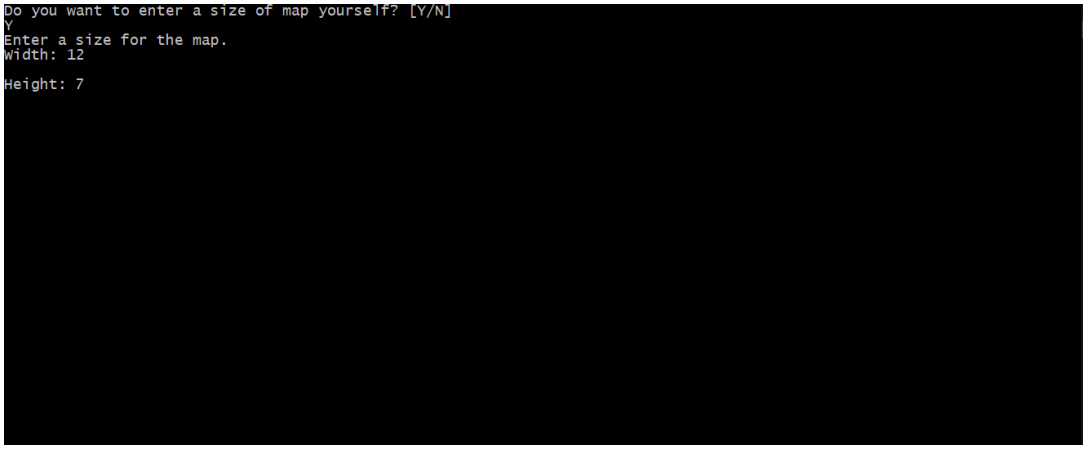
Game – класс, имеющий функцию для запуска всей игры. Сам не управляет сущностями, но через общение с медиатором следит за считыванием, прогрессом и окончанием игры.

Handle Mediator – класс, реализующий связь между Commander, CommandReader и Game. Является наследником интерфейса Mediator. Принимает от любого из этих классов «сообщения», которые обрабатывает по своей инструкции. Является элементом паттерна проектирования «Посредник».

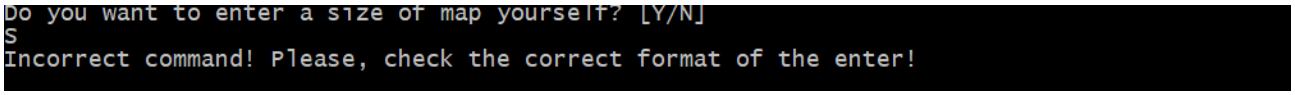
Handle – класс, от которого наследуется Commander, CommandReader и Game. Позволяет унаследовать указатель на интерфейс медиатора и метод его установки, что помогает выбирать для его детей любой медиатор, реализованный от интерфейса Mediator.



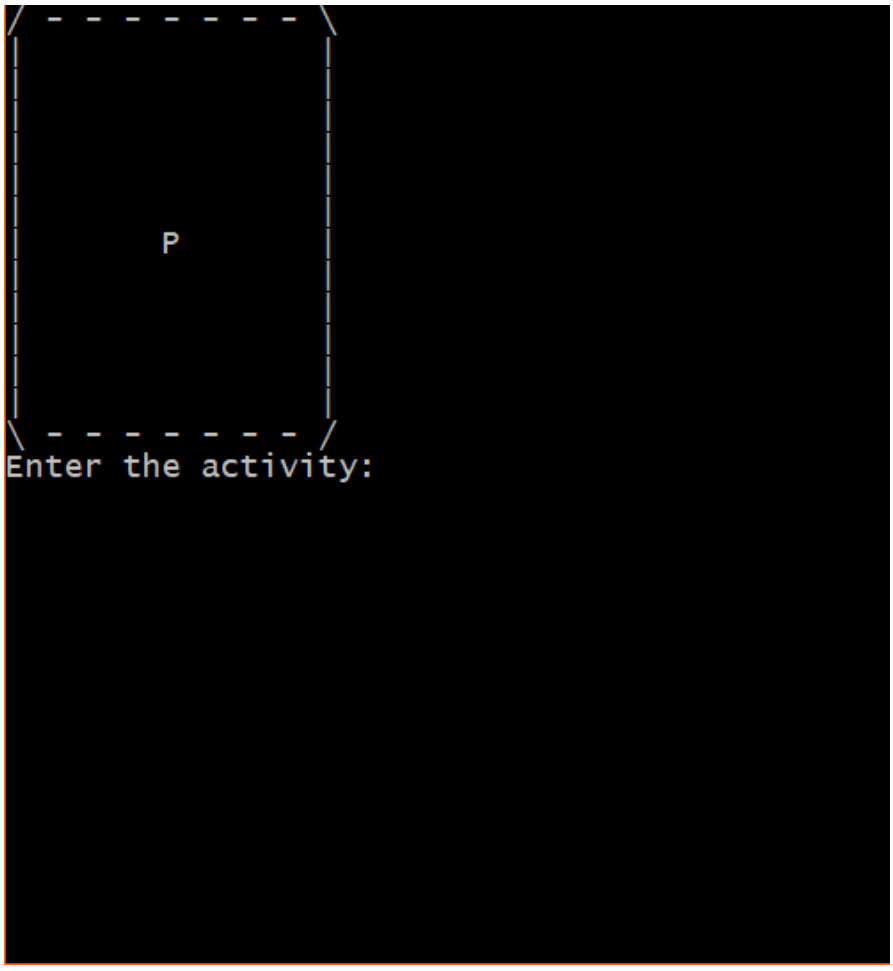
Тестирование.



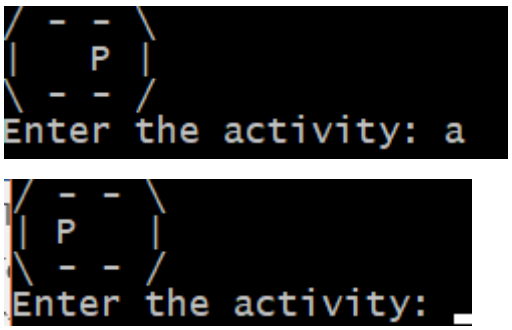
Изображение №1. Интерфейс при выборе размера поля



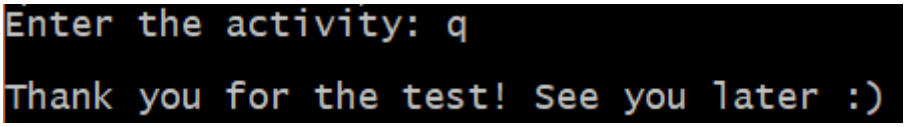
Изображение №2. Неизвестная команда.



Изображение №3. Вывод поля и считывание команд.



Изображение №4. Передвижение игрока.



Изображение №5. Выход из программы.

Выводы.

Освоен навык реализации классов, их полей и методов, в том числе и конструкторов по умолчанию, копирования, перемещения. Написания моделей UML-диаграммы.