**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра МО ЭВМ**

отчет

**по лабораторной работе № 5**

**по дисциплине «Объектно-Ориентированное Программирование»**

Тема: Шаблонные классы, генерация карты

|  |  |  |
| --- | --- | --- |
| Студент гр. 1384 |  | Алиев Д.А. |
| Преподаватель |  | Жангиров Т.Р. |

Санкт-Петербург

2022

**Цель работы.**

Разработать классы, позволяющие задать клавиши для управления. Разработать классы, обрабатывающие команды пользователя таким образом, чтобы без изменения существующего кода можно было внедрить иной способ ввода команд.

**Задание.**

Реализовать шаблонный класс генерирующий игровое поле. Данный класс должен параметризироваться правилами генерации (расстановка непроходимых клеток, как и в каком количестве размещаются события, расположение стартовой позиции игрока и  выхода, условия победы, и.т.д.). Также реализовать набор шаблонных правил (например, событие встречи с врагом размещается случайно в заданном в шаблоне параметре, отвечающим за количество событий)

**Требования:**

* Реализован шаблонный класс генератор поля. Данный класс должен поддерживать любое количество правил, то есть должен быть variadic template.
* Реализовано не менее 6 шаблонных классов правил
* Классы правила должны быть независимыми и не иметь общего класса-интерфейса
* При запуске программы есть возможность выбрать уровень (не менее 2) из заранее заготовленных шаблонов

**Примечания:**

* Для задания способа генерации можно использовать стратегию, компоновщик, прототип

**Выполнение работы.**

Был создан шаблонный класс-генератор карты MapGenerator, принимающий любое количество шаблонных параметров. В нём был создан публичный метод generate(), возвращающий сгенерированную карту. Создание карты сводится к следующим пунктам:

* Найти среди правил то, что может создать карту (имеет метод createMap). Использовать такое правило в первую очередь, иначе используется стандартный конструктор карты.
* Пройтись по всем правилам, умеющим модифицировать карту, и модифицировать ранее созданную карту с их помощью.
* Вернуть сгенерированную карту.

Были созданы следующие правила генерации карты:

* FieldSizeRule<int, int> определяет размеры поля. Создаёт поле заданного размера.
* WinEventPositionRule<int, int> определяет координаты события, приводящего к победе.
* PlayerPositionRule<int, int> определяет изначальное положение игрока.
* ObstaclesRule<ObstaclesVariant> определяет способ расставления препятствий.
* LoseEventPositionRule<int, int> определяет координаты события, приводящего к поражению.
* DamageEventFrequencyRule<int> определяет процент заполнения карты событиями урона.

На выбор пользователю предоставляется два уровня, каждый создан с разными правилами генерации.

Диаграмма разработанных в ходе лабораторной работы классов:

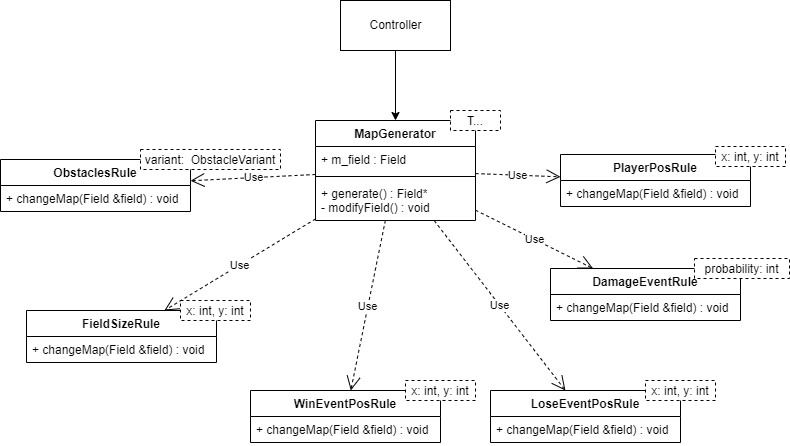


Рисунок – Диаграмма классов

**Тестирование.**

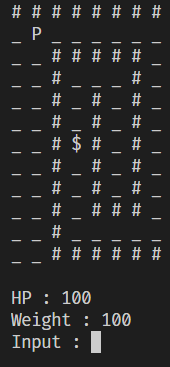
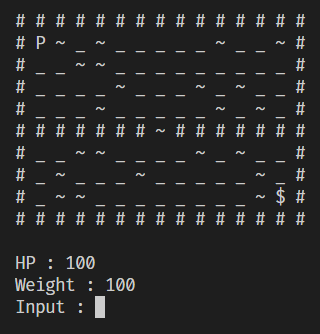
При запуске пользователю предлагается на выбор 2 карты, каждая из которых генерируется с разными шаблонными правилами.

Рисунок ,3 – варианты карты

**Выводы.**

В ходе выполнения лабораторной работы был разработан шаблонный класс – генератор карты, использующий классы-правила, передаваемые генератору в качестве шаблонных параметров. При этом правила не имеют общего базового класса и полностью независимы.