**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. 1303 |  | Чубан Д.В. |
| --- | --- | --- |
| Преподаватель |  | Жангиров Т.Р. |

Санкт-Петербург

2022

**Цель работы:**

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.
* Класс генератор создает поле, а не принимает его.
* Класс генератор не должен принимать объекты классов правил в каком-либо методе, а должен сам создавать (в зависимости от реализации) объекты правил из шаблона.
* 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
* Классы правила не должны быть только “хранилищем” для данных.
* Так как используются шаблонные классы, то в генераторе не должны быть dynamic\_cast

**Описание архитектурных решений и классов:**

В данной лабораторной работе реализована система генерации поля игры со случайным наполнением с возможностью задать его характеристики. Пользователю предлагаются на выбор два уровня - размером 5✕5 клеток и 10✕10, отличающиеся количеством событий, находящихся на поле.

Для этого реализован класс FieldGen, который имеет поля Level level, LevelType\* type и FieldScheme scheme.

Первые два поля отвечают за определение уровня, который будет сгенерирован. Поле scheme является объектом класса Field scheme, который является **схемой** игровой карты, по которой в итоге будет построена сама карта.

Класс LevelType является интерфейсом, от которого наследуются его реализации FirstLevel и SecondLevel. Каждый из них представляет собой класс с полем generator шаблонного класса LevelGenerator и методом generate(FieldScheme&), в котором с помощью метода generate() у поля generator к схеме поля применяются правила генерации, заложенные в выбранный пользователем уровень.

Так как класс LevelGenerator является шаблонным, то мы можем передать в него любое количество правил генерации, которые поочередно применяются к схеме поля. Порядок передачи правил не влияет на работоспособность системы.

Правила в контексте данной лабораторной работы - это шаблонные классы, каждый из которых отвечает за расположение элементов игровых систем, располагающихся на поле. Чтобы избежать коллизий при расположении элементов, у FieldScheme существует поле freeCells, которое является массивом с типом данных pair, в котором хранятся координаты всех свободных клеток. При установке элемента на поле, в массиве freeCells удаляется элемент, имеющий значения, соответствующие координатам установленного события. Это нужно для того, чтобы в правилах со случайной генерацией координат событий всегда имелся доступ к свободным клеткам.

Для того, чтобы создать игровое поле, созданному заранее объекту класса FieldGen сначала нужно установить уровень, по которому будет проводиться генерация с помощью метода setLevel(Level), после чего требуется вызвать метод create(), который вернет указатель на созданное поле.

Метод create() сначала создает схему поля, к которой применяет правила выбранного уровня, затем создает поле с размерами схемы, после чего проходя по каждому элементу схемы заполняет соответствующую ему клетку на поле. В конце, функция вернет указатель на созданное игровое поле.

Система уровней реализована паттерном “Стратегия”.

Отношения классов представлены на UML-диаграмме (рисунок 1).

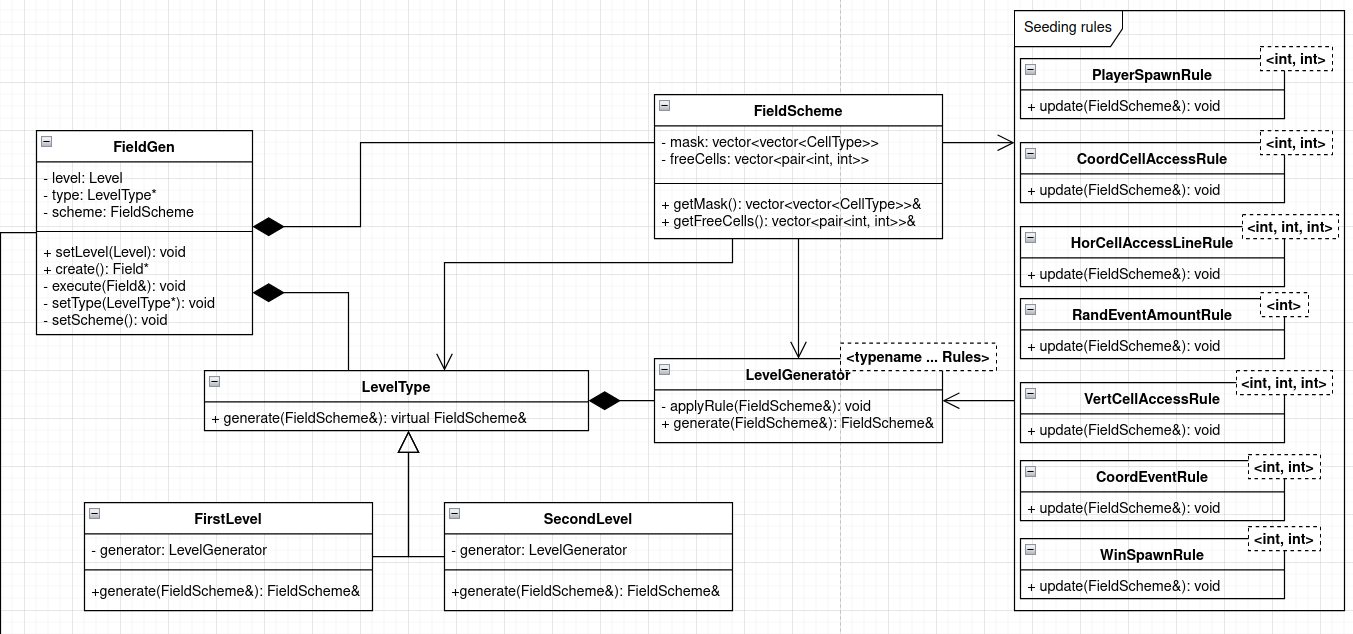


Рисунок 1 – UML-диаграмма отношений классов

**Вывод:**

Реализован класс, генерирующий игровое поле. Данный класс параметризиуется правилами генерации. Также реализован набор шаблонных правил. Создана UML-диаграмма.