

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

ОТЧЕТ
по лабораторной работе №2
по дисциплине «ООП»
Тема: Создание элементов поля.

Студент гр. 9304

Попов Д.С

Преподаватель

Размочаева Н.В.

Санкт-Петербург

2020

Цель работы.

Создать элементы игрового поля, игрока, логику взаимодействия.

Задание.

Создан класс игрока, которым управляет пользователь. Объект класса игрока может перемещаться по полю, а также взаимодействовать с элементами поля. Для элементов поля должен быть создан общий интерфейс и должны быть реализованы 3 разных класса элементов, которые по разному взаимодействуют с игроком. Для взаимодействия игрока с элементом должен использоваться перегруженный оператор (*Например, оператор +*). Элементы поля могут добавлять очки игроку/замедлять передвижения/и.т.д.

Обязательные требования:

- Реализован класс игрока
- Реализованы три класса элементов поля
- Объект класса игрока появляется на клетке со входом
- Уровень считается пройденным, когда объект класса игрока оказывается на клетке с выходом (и при определенных условиях: например, набрано необходимое кол-во очков)
 - Взаимодействие с элементами происходит через общий интерфейс
 - Взаимодействие игрока с элементами происходит через перегруженный оператор

Дополнительные требования:

- Для создания элементов используется паттерн **Фабричный метод/Абстрактная фабрика**
- Реализовано динамическое изменение взаимодействия игрока с элементами через паттерн **Стратегия**. Например, при взаимодействии с

определенным количеством элементов, игрок не может больше с ними взаимодействовать

Выполнение работы.

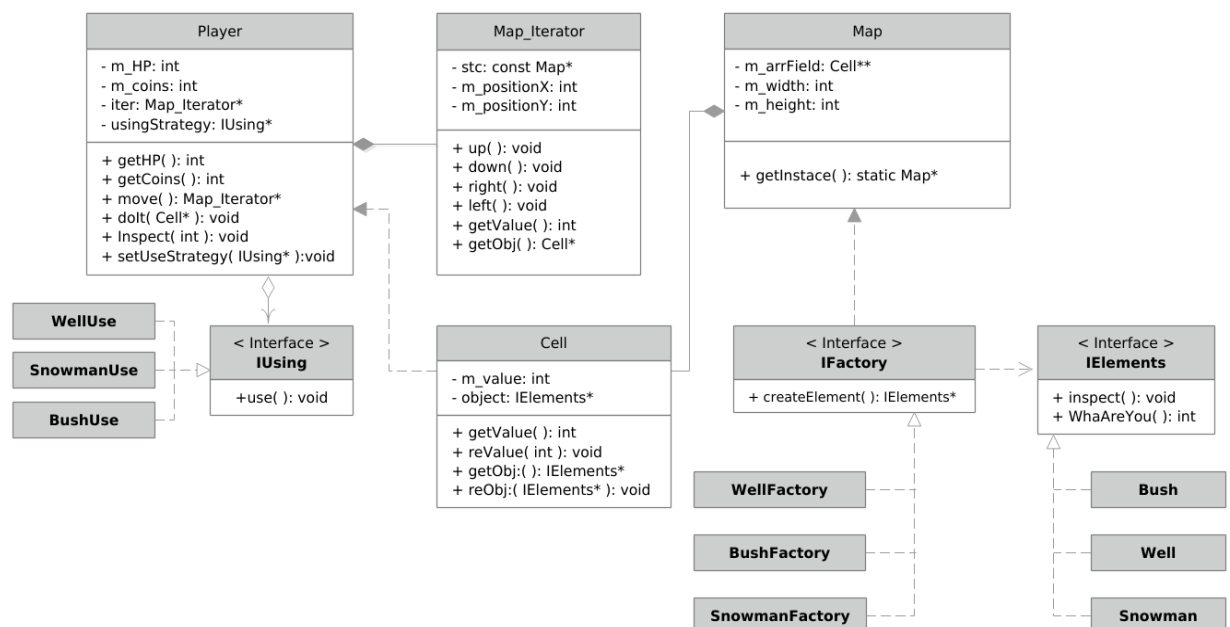


Рис. 1 - UML — диаграмма.

Создан класс игрока *Player*, который содержит в себе информацию о кол-ве собранных монет *m_coins*, собственный итератор для перемещения по полю *iter*, стратегию взаимодействия с элементами поля *usingStrategy*. Класс игрока может передвигаться благодаря методу *move*, осматривать элементы поля — *inspect*, взаимодействовать с ними — *doIt* и выводить информацию о кол-ве уже собранных монет — *getCoins*, а так же менять стратегию

поведения в зависимости от предмета с которым происходит действие — *setStrategy*. Игрок появляется на клетке со входом, а монетки добываются путем взаимодействия с элементами игрового поля, такими как: Куст(*Bush*), колодец(*Well*) и снеговик(*Snowman*). Все три элемента имеют общий интерфейс *IElements* и порождаются с помощью паттерна «Фабричный метод». Для взаимодействия с каждым из элементов поля был реализован паттерн «Стратегия», который изменяет стратегию поведения игрока.

Выводы.

Мы реализовали класс игрока, элементы игрового поля, а так же логику взаимодействия. Научились перегружать операторы. Освоили паттерны «Фабричный метод» и «Стратегия».