

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: Интерфейсы, полиморфизм

Студент гр. 0303

Болкунов В.О.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург 2021

Цель работы.

Создать классы игрока, врагов, предметов, соблюдая принципы полиморфизма.

Задание.

Могут быть три типа элементов располагающихся на клетках:

1. Игрок - объект, которым непосредственно происходит управление. На поле может быть только один игрок. Игрок может взаимодействовать с врагом (сражение) и вещами (подобрать).

2. Враг - объект, который самостоятельно перемещается по полю. На поле врагов может быть больше одного. Враг может взаимодействовать с игроком (сражение).

3. Вещь - объект, который просто располагается на поле и не перемещается. Вещей на поле может быть больше одной.

Требования:

- Реализовать класс игрока. Игрок должен обладать собственными характеристиками, которые могут изменяться в ходе игры. У игрока должна быть прописана логика сражения и подбора вещей. Должно быть реализовано взаимодействие с клеткой выхода.

- Реализовать три разных типа врагов. Враги должны обладать собственными характеристиками (например, количество жизней, значение атаки и защиты, и.т.д. Желательно, чтобы у врагов были разные наборы характеристик). Реализовать логику перемещения для каждого типа врага. В случае смерти врага он должен исчезнуть с поля. Все враги должны быть объединены своим собственным интерфейсом.

- Реализовать три разных типа вещей. Каждая вещь должна обладать собственным взаимодействием на ход игры при подборе. (например, лечение игрока). При подборе, вещь должна исчезнуть с поля. Все вещи должны быть объединены своим собственным интерфейсом.

- Должен соблюдаться принцип полиморфизма

Основные теоретические положения.

Полиморфизм — возможность объектов с одинаковой спецификацией иметь различную реализацию.

Язык программирования поддерживает полиморфизм, если классы с одинаковой спецификацией могут иметь различную реализацию — например, реализация класса может быть изменена (или в случае с реализацией интерфейса - определена) в процессе наследования. Кратко смысл полиморфизма можно выразить фразой: «Один интерфейс, множество реализаций».

Выполнение работы.

Интерфейс **Entity** — содержит виртуальный метод **interact** для влияния (взаимодействия) на другой объект реализующий данный интерфейс.

Классы реализующие интерфейс **Entity**:

- **Creature** — является базовым для класса **Enemy** и **Player**, имеет параметры здоровья, атаки и брони.
- **Item** — предмет с определённой функцией воздействия на объект класса **Creature**.

Фабрики:

- **EntityFactory** — базовая (абстрактная фабрика) для всех сущностей.
- **CreatureFactory** — фабрика для существ.
- **ItemFactory** — фабрика предметов.
- **EnemyFactory** — фабрика врагов.

Различные типы предметов и врагов реализованы как продукт соответствующих фабрик

- Для врагов: [LightFactory](#), [MediumFactory](#), [HeavyFactory](#)
- Для предметов: [HealerFactory](#), [ArmorFactory](#), [WeaponFactory](#)

Разработанный программный код см. в директории */game_lib*.

Разработанную диаграмму классов UML см. в *Lab1_UML.pdf*.

Тестирование.

Разработанные тесты см. в директории */tests*.

Выводы.

Были реализованы классы игрока, врага, предмета, объединённые общим интерфейсом для взаимодействия.