Наследование и Полиморфизм

1. Создать класс Hero, представляющий собой героя и содержащий поле

name.

Добавить конструктор, принимающий имя героя и геттер для имени (сеттер не нужен).

Добавить метод attackEnemy(), выводящий в консоль сообщение о том,

что герой атакует врага.

Создать класс TrainingGround, содержащий метод main. Протестировать

создание героя и его атаку.

2. Создать классы Warrior, Mage и Archer, представляющие собой

наследников класса Hero

Переопределить в них метод attackEnemy() для вывода

специализированного для этого класса сообщения об атаке.

Протестировать создание героев различных классов и их атаки в классе

TrainingGround.

3. Создать класс Enemy, представляющий собой врага и содержащий поле

health (количество здоровья).

Добавить конструктор, принимающий количество здоровья, а также сеттер

и геттер.

Добавить метод takeDamage(int damage), который уменьшает количество

здоровья в соответствии с полученным уроном.

Переписать метод attackEnemy класса Hero, добавив ему параметр типа

Enemy.

Метод должен вызывать у врага метод takeDamage и передавать в него

определённое количество урона.

Переопределить метод в подклассах Warrior, Mage и Archer так, чтобы

каждый герой наносил врагу разное количество урона.

4. Сделать класс Hero и его метод attackEnemy абстрактными.

5. Создать интерфейс Mortal, содержащий метод isAlive().

Сделать так, чтобы класс Enemy реализовывал интерфейс Mortal.

Определить метод isAlive в классе Enemy так, чтобы тот возвращал true,

если количество здоровья врага больше 0.

6. Создать класс BattleGround с методом main, в котором создать симуляцию

героя, атакующего врага.

7\*. Добавить герою показатель здоровья и возможность погибнуть.

Добавить возможность врагу атаковать героя в ответ.

Создать несколько видов врагов (наследников класса Enemy) с разными

способностями (например, Zombie - имеет шанс воскреснуть при гибели)

Дать героям уникальные способности.

Продемонстрировать сражение героя с несколькими соперниками.