1. Create a Hero class that represents a hero and contains the name field.

Add a constructor that accepts the hero's name and a getter for the name (no setter is needed).

Add the attackEnemy() method, which outputs a message to the console that the hero is attacking the enemy.

Create a TrainingGround class containing a main method. Test the creation of the hero and his attack.

2. Create the Warrior, Mage and Archer classes, which are the successors of the Hero class

Override the attackEnemy() method in them to display an attack message specialized for this class.

Test the creation of heroes of different classes and their attacks in the TrainingGround classroom.

3. Create a class Enemy, which is an enemy and contains the field health (amount of health).

Add a constructor that takes the amount of health as well as a setter

and getter

Add a method takeDamage(int damage) that reduces the amount of health according to the damage received.

Rewrite the attackEnemy method of the Hero class by adding a parameter of type Enemy to it.

The method should call the enemy's takeDamage method and transfer a certain amount of damage to it.

Override the method in the Warrior, Mage, and Archer subclasses so that each hero deals a different amount of damage to the enemy.

4. Make the Hero class and its attackEnemy method abstract.

5. Create a Mortal interface containing the isAlive() method. Make the Enemy class implement the Mortal interface.

Define the isAlive method in the Enemy class so that it returns true if the enemy's health is greater than 0.

6. Create a BattleGround class with the main method in which to create a simulation of a hero attacking the enemy.