

PA 303.10.1 - Practice Assignment

Polymorphism and Inheritance

Assignment Overview

This practice assignment will help you understand how to create class hierarchy using the OOPS concept. In this assignment, you will create a battle game.

Introduction

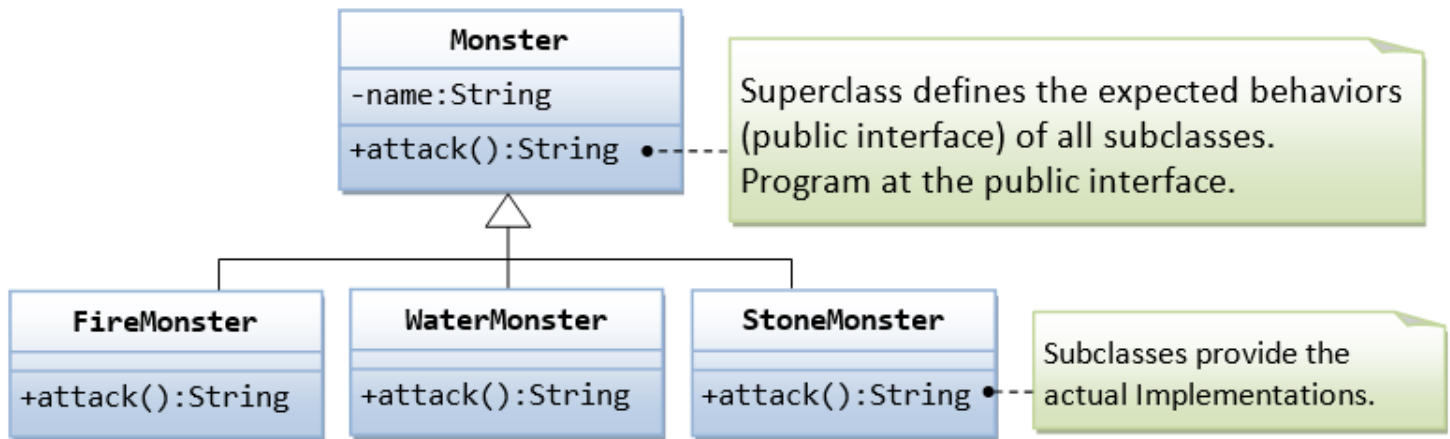
Inheritance: In the Java language, classes can be derived from other classes, thereby inheriting fields and methods from those classes. A class that is derived from another class is called a subclass (also a derived class, extended class, or child class). The class from which the subclass is derived is called a superclass (also a base class or a parent class).

Polymorphism: When a superclass reference is used to refer to a subclass object.

Requirements

Scenario: In our game app, we have many types of monsters that can attack. You will design a superclass called **Monster** and define the method of **attack()** in the superclass. You will design subclasses called **FireMonster**, **WaterMonster**, and **StoneMonster**. The subclasses will then provide their actual implementation. In the main program, we will declare instances of the superclass, substitute them with the actual subclass, and invoke the method defined in the superclass.

Monster and its Subclasses illustration



Hint: The **main()** method is the following:

```

public class TestMonster {
    public static void main(String[] args) {
        // Program at the "interface" defined in the superclass.
        // Declare instances of the superclass, substituted by subclasses.
        Monster m1 = new FireMonster("r2u2"); // upcast
        Monster m2 = new WaterMonster("u2r2"); // upcast
        Monster m3 = new StoneMonster("r2r2"); // upcast

        // Invoke the actual implementation
        System.out.println(m1.attack()); // Run FireMonster's attack()
        System.out.println(m2.attack()); // Run WaterMonster's attack()
        System.out.println(m3.attack()); // Run StoneMonster's attack()

        // m1 dies, generates a new instance and re-assign to m1.
        m1 = new StoneMonster("a2b2"); // upcast
        System.out.println(m1.attack()); // Run StoneMonster's attack()

        // We have a problem here!!!
        Monster m4 = new Monster("u2u2");
        System.out.println(m4.attack()); // garbage!!!
    }
}
  
```

Expected Output:

```
Attack with fire!
Attack with water!
Attack with stones!
Attack with stones!
!^_&^$@+%$* I don't know how to attack!
```

Submission Instructions:

Include the following deliverables in your submission -

- Submit your source code using the Start Assignment button in the top-right corner of the assignment page in Canvas.

CANVAS STAFF USE ONLY: Canvas Submission Guideline:

Instructions for Canvas Assignment Creation
<p>Assignment Name: PA 303.10.1 - Practice Assignment - Polymorphism and inheritance</p> <p>Points: 100</p> <p>Assignment Group: Module 303: Java SE Review (Not Graded)</p> <p>Display Grade As: Complete/Incomplete</p> <p>Do not count this assignment towards the final grade: Checked</p> <p>Submission Types: Website URL</p> <p>Everything else is the default.</p>