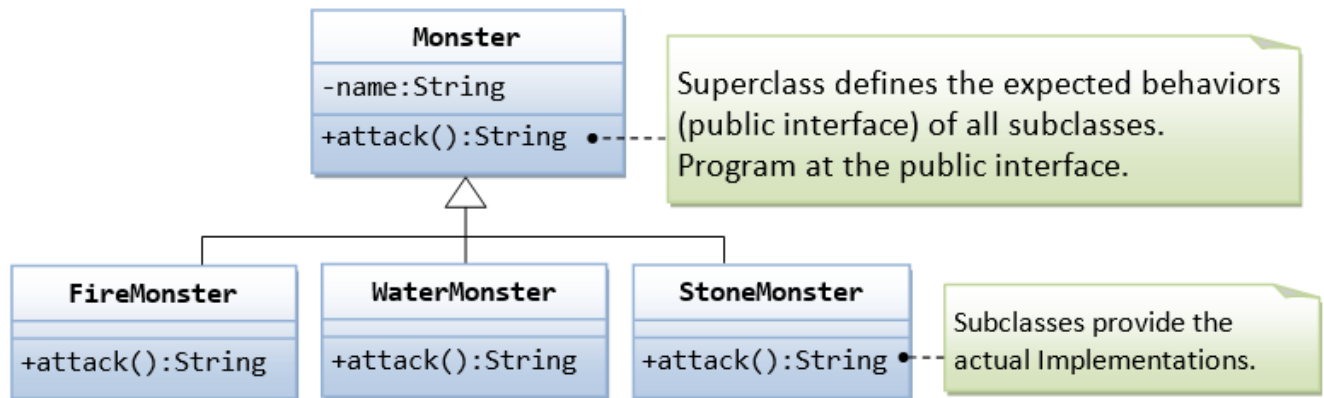


Assignment of OOPs

Monster and its Subclasses



Scenario, In our game app, we have many types of monsters that can attack. We shall design a superclass called Monster and define the method attack() in the superclass. The subclasses shall then provides their actual implementation. In the main program, we declare instances of superclass, substituted with actual subclass; and invoke method defined in the superclass.

Hint: Main method is 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, generate 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!!!
    }
}
```

output:

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