Software Engineering and Programming Basics - WS2021/22
Exercise 9

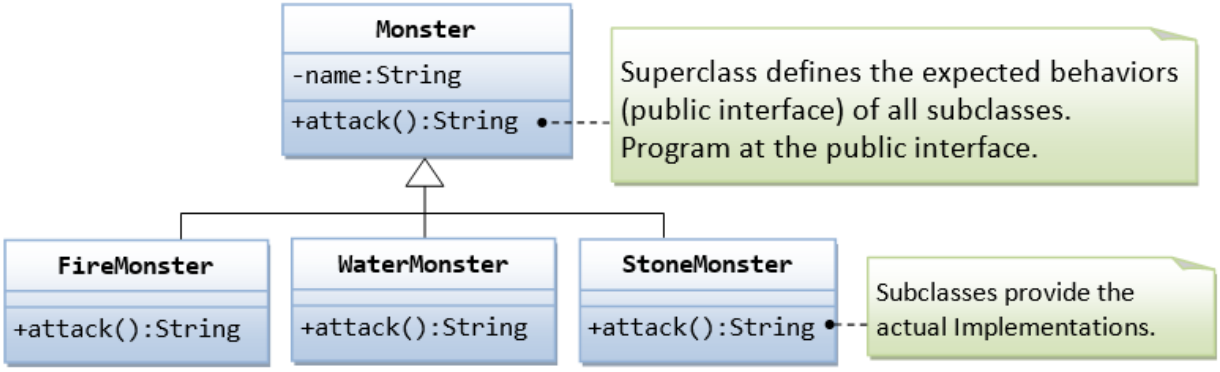
TECHNISCHE UNIVERSITÄT
CHEMNITZ

Professorship of Software Engineering

12| 2021

Task 1

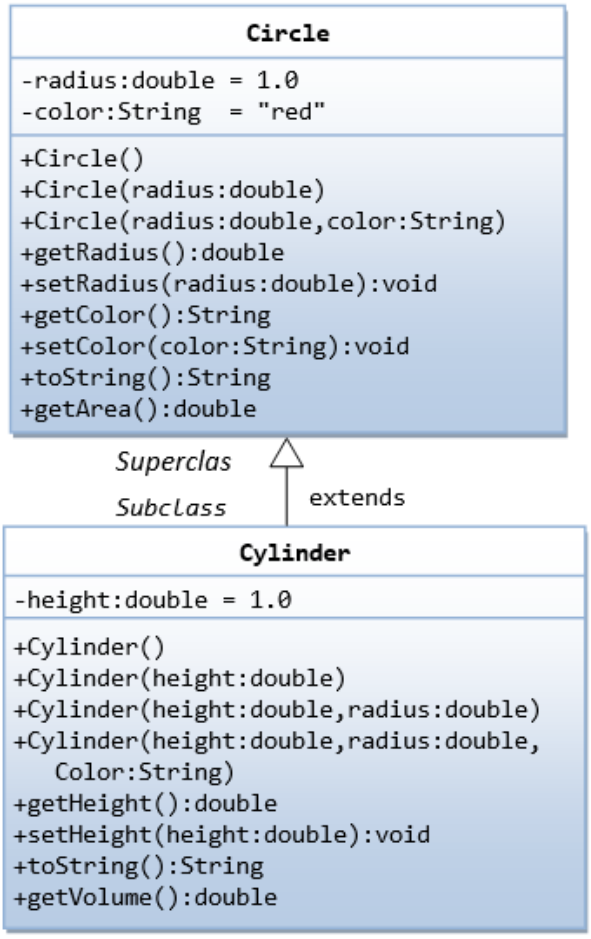
Let us consider a game app, where we have many types of monsters that can attack. We shall design a superclass called Monster and define the method attack() in the superclass. The subclasses shall then provide their individual implementation (overriding). In the main program, we declare references of superclass, substituted with instances of subclass; and invoke method defined in the superclass (polymorphism). Below UML diagram gives more details about the class hierarchy.



Note: Overriding is a feature that allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its super-classes or parent classes. When a method in a subclass has the same name, same parameters or signature, and same return type(or sub-type) as a method in its super-class, then the method in the subclass is said to override the method in the super-class.

Task 2

In this example, we derive a subclass called Cylinder from the superclass Circle, which we have created in the previous exercise(03). It is important to note that we reuse the class Circle. Reusability is one of the most important properties of OOP. (Why reinvent the wheels?) The class Cylinder inherits all the member variables (radius and color) and methods (getRadius(), getArea(), among others) from its superclass Circle. It further defines a variable called height, two public methods - getHeight() and getVolume() and its own constructors, as shown:



Note : Overloading allows different methods to have the same name, but different signatures where the signature can differ by the number of input parameters or type of input parameters or both.