
Inheritance –
Getting Started

Inheritance – getting started

Objective

The primary objective of this lab is to enable you to derive new types and to add specialist functionality.

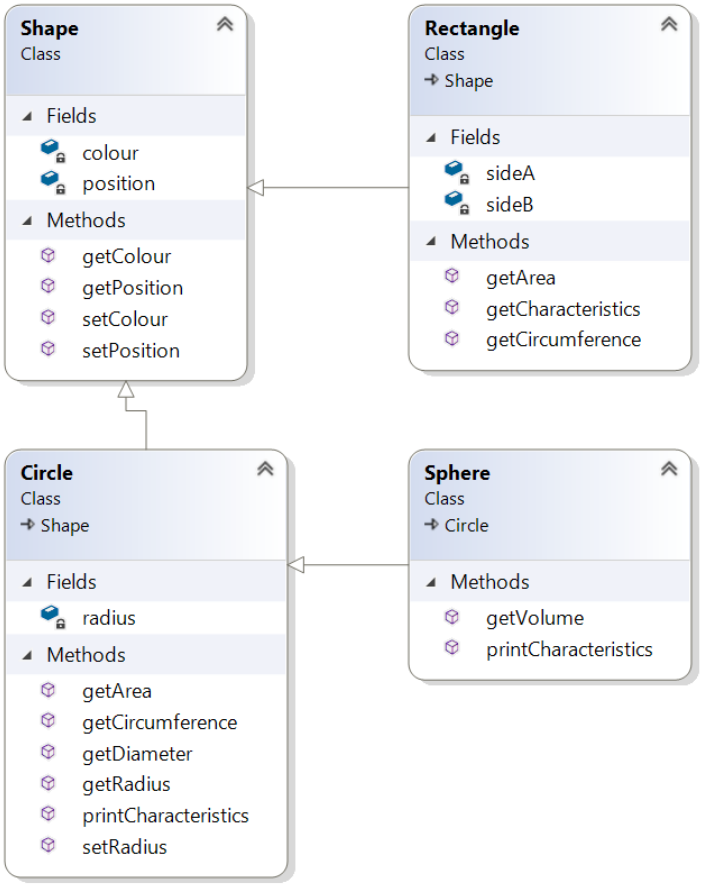
Overview

The lab introduces some of the basic concepts of the inheritance story. As mentioned in the associated session, in order to implement inheritance, you must first have a class that provides the fundamental definition or behavior you need. In this lab we will play about with circular shapes.

This practical will be built on two chapter's time.

Step by step

1. Open the labs project and then add a new package called lab04.
2. Create a new class called Program with a main() method in the lab04 package.
3. Create the following class structure.



4. Create a constructor for Shape to set its colour and position.

5. As you can see, **Circle** extends **Shape** and **Sphere** extends **Circle**.
   a. ***Position*** is of type **Point** which is a class with built-in x and y.
   b. ***Colour*** is of type **java.awt.Color**

   Tip: Use **Math.PI** to get the value of **PI**.
   You will need this to calculate the area and circumference of circle.

   The volume of a sphere is calculated as 4/3 * PI * R^3 (R to power of 3)
   You can use the Math.pow() fuction or R * R * R

6. Create getters and setters for each of the fields (colour, radius...) as indicated in the class diagram above.
7. The **getCharacteristics()** method returns a *String* containing all the attributes of the shape. It will be up to the caller how to display this information.
8. Create a few shape types in main().
9. Print the characteristics of the Rectangle, Circle, and Sphere objects which you've created.

10. Create an ArrayList<Shape> called **shapes** in the main()

11. Add the shapes which you created earlier into the *shapes* ArrayList.

12. Create an enhanced **for** loop to scroll through each shape and print its colour and position (x,y).

    How does this work?! How can we store a shape like Rectangle in a list of Shapes? All will be revealed in the next chapter.

*** End ***