BIM224- Object-Oriented Programming – Homework 1

Note: Students are encouraged to submit the homework as groups consisting of 1 or 2 members. Please make a single submission and send a “group.txt” file indicating group members if the group has 2 members. All of the homeworks will be checked against plagiarism by a specialized software.

1) (30 pts) You will implement the inheritance hierarchy described in Figure 1. Classes and methods shown in italics are abstract.

Here’s a description of what each concrete method should do:

- **draw()**: print a message about “type of the shape”, “x-y coordinates”, “color”, and other shape specific properties.

- **getArea()**: calculate the area of the shape and return to the caller.

- **getPerimeter()**: calculate the perimeter of the shape and return to the caller.

- **resize(double s)**: resizes the shape according to this coefficient. For example, entering 0.5 as parameter will reduce its dimensions half the original.

Also, write a test class namely ShapeTester.java in order to test all of the methods in concrete classes Square and Circle. Shapes must be comparable because of implementing Comparable interface. It is necessary to write tests for methods of the Comparable interface. It is necessary to send Shape.java, Square.java, Circle.java and ShapeTester.java.
2) (70 pts) Design new ComparableCircle2D and ComparableRectangle2D classes that extend the abstract GeometricObject2D class. Implement them as follows:

**ComparableCircle2D**: Two double data fields named x and y that specify the center of the circle with get methods.
- A data field radius with a get and set methods.
- A no-arg constructor that creates a default circle with (0, 0) for (x, y) and 1 for radius.
- A constructor that creates a circle with the specified x, y, and radius.
- A method getArea() that returns the area of the circle.
- A method getPerimeter() that returns the perimeter of the circle.
- A method contains(double x, double y) that returns true if the specified point (x, y) is inside this circle (see Figure 2).
- A method contains(ComparableCircle2D circle) that returns true if the specified circle is inside this circle (see Figure 2).
- A method overlaps(ComparableCircle2D circle) that returns true if the specified circle overlaps with this circle (see Figure 2).
- A method that implements the compareTo method defined in Comparable interface.
- A method that overrides toString() method.

**ComparableRectangle2D**: Two double data fields named x and y that specify the center of the rectangle with get and set methods. (Assume that the rectangle sides are parallel to x or y-axes.)
- The data fields width and height with get and set methods.
- A no-arg constructor that creates a default rectangle with (0, 0) for (x, y) and 1 for both width and height.
- A constructor that creates a rectangle with the specified x, y, width, and height.
- A method getArea() that returns the area of the rectangle.
- A method getPerimeter() that returns the perimeter of the rectangle.
- A method contains(double x, double y) that returns true if the specified point (x, y) is inside this rectangle (see Figure 3).
- A method contains(ComparableRectangle2D r) that returns true if the specified rectangle is inside this rectangle (see Figure 3).
- A method overlaps(ComparableRectangle2D r) that returns true if the specified rectangle overlaps with this rectangle (see Figure 3)
- A method that implements the compareTo method defined in Comparable interface.
- A method that overrides toString() method.

Figure 3: A point is inside the rectangle. (b) A rectangle is inside another rectangle. (c) A rectangle overlaps another rectangle. (d) Points are enclosed inside a rectangle.

It is necessary to send ComparableCircle2D.java, ComparableRectangle2D.java, GeometricObject2D.java and Shape2DTester.java.

Simple test file for these classes is given below:

```java
public class Shape2DTester {
    public static void main(String[] args) {
        // Declare and initialize two geometric objects
        GeometricObject2D geoObject1 = new ComparableCircle2D(0, 2, 5);
        GeometricObject2D geoObject2 = new ComparableRectangle2D(0, 1, 5, 3);

        System.out.println("The two objects have the same area? " +
                           equalArea(geoObject1, geoObject2));
        displayGeometricObject(geoObject1);
        displayGeometricObject(geoObject2);

        GeometricObject2D geoObject3 = new ComparableCircle2D(0, 2, 3);
        GeometricObject2D geoObject4 = new ComparableRectangle2D(0, 3, 5, 3);

        System.out.println();
        System.out.println("geoObject1 contains (4,5) : " + geoObject1.contains(4,5));
        System.out.println("geoObject1 contains geoObject3: " +
                           geoObject1.contains(geoObject3));
        System.out.println("geoObject1 overlaps geoObject3: " +
                           geoObject1.overlaps(geoObject3));

        System.out.println();
    }
}
```
System.out.println("geoObject2 contains (4,5) : " + geoObject2.contains(4,5));
System.out.println("geoObject2 contains geoObject4: " + 
geoObject2.contains(geoObject4));
System.out.println("geoObject2 overlaps geoObject4: " +
geoObject2.overlaps(geoObject4));

GeometricObject2D[] rectangles = {
    new ComparableRectangle2D(0, 0, 3.4, 5.4),
    new ComparableRectangle2D(2, 3, 13.24, 55.4),
    new ComparableRectangle2D(),
    new ComparableRectangle2D(5, 7, 1.4, 25.4)};
java.util.Arrays.sort(rectangles);
for (GeometricObject2D rectangle: rectangles) {
    System.out.print(rectangle + " ");
    System.out.println();
}

GeometricObject2D[] circles = {
    new ComparableCircle2D(0, 0, 3.4),
    new ComparableCircle2D(2, 3, 13.24),
    new ComparableCircle2D(),
    new ComparableCircle2D(5, 7, 1.4)};
java.util.Arrays.sort(circles);
for (GeometricObject2D circle: circles) {
    System.out.print(circle + " ");
    System.out.println();
}

/** A method for comparing the areas of two geometric objects */
public static boolean equalArea(GeometricObject2D object1, 
GeometricObject2D object2) {
    return object1.getArea() == object2.getArea();
}

/** A method for displaying a geometric object */
public static void displayGeometricObject(GeometricObject2D object) {
    System.out.println();
    System.out.println("The area is " + object.getArea());
    System.out.println("The perimeter is " + object.getPerimeter());
}