CS102- Algorithms and Programming II

Lab 03

Lab Objectives: Inheritance, Abstract Classes, Polymorphism

Notes:

- For all labs in CS 102, your solutions must conform to these CS101/102 style guidelines.
- Create a Java Project named Lab03. Put all of your classes in this project.
- Remember to include **javadoc comments** for each class and method.
- Upload your solution **as a single .zip file** to the Lab03 assignment by the end of your section's lab session in the week of March 1. You must use the following naming convention: Lab03_Surname_FirstName.zip where Surname is your family name and FirstName is your first name. You may upload multiple times; the last upload will be considered.

Create an abstract class called **`Shape2D`**. A 2D shape has two properties: x and y coordinates of its center. Its constructor takes two parameters: x and y coordinates. You may assume that both coordinates are integers.

This class has the following methods:

- **`public abstract double calculatePerimeter()`** - returns perimeter of a 2D shape. This method will be an abstract method.

- **`public abstract double calculateArea()`** - returns area of a 2D shape. This method will be an abstract method.

- **`public double calculateDistance( Object anyShape )`** – returns the euclidean distance between the centers of two **`Shape2D`** objects. If **`anyShape`** parameter is not a **`Shape2D`** instance, it should return -1.

- **`public String toString()`** – returns a string representation of two properties: x and y coordinates of its center. Make sure to use **`@Override`** annotation before the method definition, because this method overrides the **`toString()`** method in our superclass **`Object`**.

- **`public boolean equals( Object o )`** – returns true if given object is a **`Shape2D`** object that is equal to this one; false otherwise. Make sure to use **`@Override`** annotation before the method definition, because this method overrides the **`equals()`** method in our superclass **`Object`**.

After creating the **`Shape2D`** class, create the following three classes.

- Create a class **`Circle`** which extends **`Shape2D`** class. Circle class has an additional property, **radius**. Circle is a 2D shape, therefore its constructor takes three parameters, x and y coordinates and radius.

- Create a class **`Rectangle`** which extends **`Shape2D`** class. Rectangle class has additional properties, **height** and **width**. Its constructor takes four parameters; x, y coordinates, height and width.

- Create a class **`Square`** which extends **`Shape2D`** class. Square class has an additional property, **sideLength**. Its constructor takes three parameters, x and y coordinates and sideLength.

All classes will also implement **`calculatePerimeter()`** and **`calculateArea()`** methods, and override **`toString()`** and **`equals()`** methods. Reuse the **`toString()`** and **`equals()`** implementations in the super class when implementing these methods in the subclasses with **`@Override`** annotation before the method definitions.

Write a test class, **`ShapeTest.`** Create an array of three shape objects: one rectangle, one square and one circle. Then, find a 2D Shape object with the longest perimeter and the object with the largest area to the screen and print whether these objects are Square, Circle or Rectangle. You should do this by implementing and using the following methods.

- **`public static Shape2D findLargestArea() -`** Write a method which takes an array of **`Shape2D`** objects as parameter and returns the shape with the largest area in the shape list.

- **`public static Shape2D findLongestPerimeter() -`** Write a method which takes an array of **`Shape2D`** objects as a parameter and returns the shape with the longest perimeter in the shape list.

Demonstrate the uses of your `toString()` and `equals()` methods from these classes. Also, calculate the distances between these three **`Shape2D`** objects.

Sample Run

```
[class Rectangle]x = 2, y = 3 height = 8 and width = 15
[class Circle]x = 13, y = 15 and radius = 3.0
sq: [class Square]x = -2, y = -5 and side = 5
sq2: [class Square]x = -2, y = -5 and side = 5
sq3: [class Square]x = -1, y = -5 and side = 5
sq4: null
sq.equals(  sq 2 ) is true
sq.equals(  sq3  ) is false
sq.equals(  sq4  ) is false
sq.equals(  circl  ) is false
The shape array:[[class Rectangle]x = 2, y = 3 height = 8 and width = 15, [class Circle]x = 13, y = 15 and
radius = 3.0, [class Square]x = -2, y = -5 and side = 5]
Circle has largest area
Rectangle has longest perimeter
Distance between shape 1 shape 2 is 16.278820596099706
Distance between shape 1 shape 3 is 8.94427190999916
Distance between shape 2 shape 3 is 25.0
```