**Langara College**

**Department of Computing Science & Information Systems**

**CPSC1181 – Object-oriented Computing**

###### **Lab6: Abstract Classes and Interfaces**

**Objectives:**

* Create abstract classes and interfaces
* Create and test classes that extend abstract classes and implement interfaces

**Instructions:**

1. Inside **IntelliJ**, create a new project named **Lab6** to store all the files for this lab. Make sure select **Maven** as the Build System and **lab6** as the **GroupId** and **ArtifactId**.
2. Add Javadoc comments and regular comments to all your code.
3. Generating JavaDoc is not required for this lab.

**Problem [65 marks]**

**Part 1: [10 Marks] Create the GeometricShape class**

Create an abstract class called GeometricShape. A GeometricShape may or may not be filled in. GeometricShape should contain a standard constructor that sets the data member and a no-argument constructor that creates a filled in shape, as well as getters and setters for the instance data and a toString method.

Create two abstract methods: one for getting the area of a shape, and one for getting the perimeter of a shape.

**Part 2: [5 Marks] Implement the Comparable interface**

Modify GeometricShape so that it implements the Comparable interface. Implement the compareTo() method, comparing shapes by their areas. **The Comparable interface is part of the Java library, do not try to create the Comparable interface yourself.**

**Part 3: [10 Marks] Create the RectagnleShape class**

Create a child class of GeometricShape, called RectangleShape. RectangleShape has a width and a height. Those values should be positive integers. RectangleShape should have an appropriate constructor for initializing instance data. Provide getters and setters for the data members and a toString method.

In RectangleShape, be sure to implement the two abstract methods of the parent class.

**Part 4: [5 Marks] Create the ASCIIDrawable interface**

Create an interface called ASCIIDrawable. The interface should have one unimplemented method, String drawAsASCII() which will return a String containing the object drawn using ASCII characters.

**Part 5: [10 Marks] Implement the ASCIIDrawable interface**

Add implements ASCIIDrawable to the GeometricShape class header, but implement the drawAsASCII () function in the RectangleShape subclass. See the examples below for the expected output.

**Note: The toString method should not be used to draw anything. It is only to display the basic info of an object.**

**Part 6: [10 Marks] Create the Billboard class**

Create a new class, Billboard, that represents a sign containing a message. It contains one String data member, message, a setter and getter for it, and a toString. It should implement the ASCIIDrawable interface. Have the message **on a single line with a space separating the message from the surrounding line**.

See the example below for the expected output.

**Note:** **A billboard does not extend any of the previous classes.** **It does not use the previous class to draw the rectangle around the message**

**Part 7: [10 Marks] Create the GeometricShapeTester class**

Create a tester class called GeometricShapeTester (not JUnit). Create an Array (not an ArrayList) of GeomtricShapes that contains several RectangleShapes.

Use **Arrays.sort** to demonstrate that your array of shapes is sorted ascendingly by area.

Create an array or ArrayList with the ASCIIDrawable interface type and use it to test the draw function of RectangleShapes and Billboards.

**Part 8: [5 Marks] Create the PerimeterComparator class**

Create a comparator class called PerimeterComparator that compares GeometricShape objects by perimeter in decreasing order. In GeometricShapeTester, use the comparator to sort the array in descending order by perimeter.

Graphical user interface, text, application

Description automatically generated

**What to hand in**

Zip the folder **Lab6** and upload it to D2L.

**When to hand in**

By 11:59pm, Wednesday, February 21, 2024.