CPSC 1181 - Lab 5 [60 marks]

Objectives:

- Create abstract classes
- Implement interfaces
- Use interface types to perform polymorphism

Submission:

- Zip up all of Java files and submit them to D2L prior to the due date.
- Unzipped submissions or submissions containing .class files will be automatically given ZERO

Exercise 1

Create an abstract class called GeometricShape. A GeometricShape has a color and may or may not be filled with that color. GeometricShape should contain a default constructor that creates a shape filled with the color white, as well as getters and setters for the two 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.

Exercise 2

Modify Geometric shape so that it implements the comparable interface. How should shapes be compared? How do you tell if one shape is larger than another? Probably by area.
Then create a static method called max that accepts two GeometricShapes as parameters and returns the larger of the two objects. If the objects are the same size, return the first object passed as a parameter.

Exercise 3

Create two child classes of GeometricShape, one called Circle and one called Octagon. Octagon has a side length, Circle has a radius, and each should have an appropriate constructor for initializing instance data. In each class, provide getters and setters for the side length/radius and a toString method.

In both of these classes, be sure to implement the abstract methods of the parent class.

Exercise 4

Create a static method for finding the total sum of the areas of a collection of GeometricShapes. The method signature is : `public static double sumArea(GeometricShape[] shapes)`

Exercise 5

Create a tester class called FunWithShapes. Create an Array of GeometricShapes that contains several Octagons and several Circles. Show that each of the following methods work correctly, for objects of either type:

- max
- sumArea
- compareTo (for both Circle and Octagon)
- Use Arrays.sort to demonstrate that your collection of shapes is sorted ascendingly by area

Exercise 6

Create a comparator called ShapeComparator that compares GeometricShape objects by perimeter. In FunWithShapes, use the comparator to sort the collection in descending order.

CPSC 1181 - Lab 5 [60 marks]

Marking Rubric:

Style, Convention, Documentation [6 marks]

GeometricShape.java[19 marks]

+5 implementing comparable
+4 instance data, getters/setters, toString
+1 getArea abstract
+1 getPerimeter abstract
+4 correctly implementing max
+3 correctly implementing sumArea

ShapeComparator.java[8 marks]

+1 class declaration
+5 correctly implementing compare
+2 demonstrate comparator use in FunWithShapes

Circle.java [9 marks]

+1 extends
+2 constructor and instance data
+2 getter/setter/toString
+2 getArea
+2 getPerimeter

Octagon.java [9 marks]

- see Circle.java

FunWithShapes.java[10 marks]

+3 checking different cases for Max
+3 checking differenct cases for sumArea
+3 checking different cases for compareTo
+1 showing sort works ascendingly