Gebze Technical University
Department of Computer Engineering
CSE 241/505
Object Oriented Programming
Fall 2018
Homework # 7
Your First Java Program

Due date Jan 20rd 2019 (No late submission for this HW)

In this homework, you will write a Java Shape hierarchy that you wrote already for HW5 in C++.

Interface **`Shape`** defines at least the following method.

- **`area`** that returns the area of the shape
- **`perimeter`** that returns the perimeter
- Functions **`increment`** and **`decrement`** for incrementing and decrementing the shape positions by 1.0.
- This interface implements the **`Comparable`** interface to compare shapes with respect to their areas.
- Draw takes a **`Graphics`** object as parameter and draws the shape. This method will be called from the **`paintComponent`** method of a **`JPanel`** object.

Class **`Rectangle, Triangle, Circle`** and **`ComposedShape`** all implement the **`Shape`** interface. They behave like the classes in HW3. Class ComposedShape keeps an array of Shape references for the shape elements.

`Polygon` is an abstract class that implements. **`PolygonVect and PolygonDyn`** are two concrete classes that derive from **`Polygon`** class. One of them uses Collection class ArrayList vectors to keep the 2D points, the other uses Java arrays to keep the 2D points.

We also define the following static methods in a separate class

- Method drawAll takes an array of Shape references and draws all shapes to an JPanel
- Method convertAll takes an array of Shape references, converts all shapes to Polygons and returns a new array with the new shapes.
- Method sortShapes takes an array of Shapes and increasingly sorts them with respect to their areas.

Use the program that our TA Ahmet Soyyiğit showed you during the PS to draw this shape hierarchy In UML diagrams and submit the results.

Notes:

- Do error and range checking for any parameters. Throw exceptions and test them in your client code. Do not forget to define the throw lists for your functions.
- For each class you will write appropriate class documentation for Javadoc. You will also submit the Javadoc files.

- As expected, you should follow all object-oriented programming principles.
- You should submit your work to the moodle page.
- You should submit the images of drawn shapes.