

Lab 12

Before you start:

1. Download Lab12-Problem.zip from the course website.
2. Unzip the file and you will find four java files `GeometricObject.java`, `Triangle.java`, `Circle.java`, `Rectangle.java`.
3. Add these files into the `src` folder in your project. Refresh (right click) in Eclipse and you should see the files listed in the `src` in the default package in Eclipse.

In your project, we now have a superclass `GeometricObject` and three subclasses `Circle`, `Triangle` and `Rectangle`.

Problem 1: Follow the instructions below:

1. Use an `ArrayList` to store randomly generated `Circle` or `Triangle` objects.
2. While generating these objects, assign radius with a random number between 0 and 1 for `Circle` objects and three sides with random numbers between 1 and 2 for `Triangle` objects.
3. While adding the objects into the `ArrayList`, you should reject those with area less than the existing objects in the `ArrayList` and only add those with area greater than the existing elements.
4. Stop the addition when you have five elements in the `ArrayList`.

Below is a sample run:

An `ArrayList` with five `Circle` or `Triangle` objects

```
Triangle: side1 = 1.2189088018581828 side2 = 1.5671204270520565 side3 =  
1.9501348396659737 created on Sun May 11 11:44:53 CST 2014  
color: white and filled: false  
Area: 0.9544598828342201
```

```
Circle: radius = 0.5636217274005368 created on Sun May 11 11:44:53 CST 2014  
color: white and filled: false  
Area: 0.9979880154100654
```

```
Circle: radius = 0.6286005571859208 created on Sun May 11 11:44:53 CST 2014  
color: white and filled: false  
Area: 1.2413647129586758
```

Triangle: side1 = 1.6762841346844843 side2 = 1.8995391672494362 side3 = 1.6589402779554627 created on Sun May 11 11:44:53 CST 2014
color: white and filled: false
Area: 1.301812125846416

Circle: radius = 0.8714794337718632 created on Sun May 11 11:44:53 CST 2014
color: white and filled: false
Area: 2.385965489770585

Problem 2: Follow the instructions below:

1. Design an interface named `Colorable` with a void method named `howToColor()`.
2. Every class of a colorable object must implement the `Colorable` interface.
3. Design a class named `Square` that extends `GeometricObject` and implements `Colorable`.
4. Implement `howToColor` to display the message: `Color all four sides`.

A sample client code and sample run look like:

```
public static void main(String[] args) {  
    Square square = new Square(2);  
    square.howToColor();  
}
```

Color all four sides

Problem 3: Given an abstract class named `GraphicsObject` and two interfaces named `Object2D` and `Object3D` as follows:

```
public abstract class GraphicsObject {  
    public abstract String getName();  
}  
  
interface Object2D {  
    public abstract double getArea();  
}  
  
interface Object3D {  
    public abstract double getVolume();  
}
```

Please complete the following steps:

1. Design a subclass of `GraphicsObject` named `Rectangle` that implements the `Object2D` interface, and
2. Design another subclass of `GraphicsObject` named `Box` that implements the `Object3D` interface.
3. The `Rectangle` class has two data fields named `width` and `height`, while
4. the `Box` class has three data fields named `length`, `width` and `height`.
5. Complete the following test program:

```
public class TestGraphicsObject {  
  
    public static void main(String[] args) {  
        GraphicsObject[] go = new GraphicsObject[3];  
        go[0] = new Rectangle(1,2);  
        go[1] = new Box(2,3,4);  
        go[2] = new Rectangle(3,4);  
  
        for (int i=0; i<go.length; i++) {  
            //to be completed  
        }  
    }  
}
```

So the following output is obtained:

Name: Rectangle; Area = 2.0

Name: Box; Volume = 24.0

Name: Rectangle; Area = 12.0