SIM GLOBAL EDUCATION

UOW AUSTRALIA

SCIT

School of Computing and Information Technology
Faculty of Engineering & Information Sciences

CSIT121
Object Oriented Design and Programming
Assignment 2
File name: YourName _ClassListNo_A2.java

Objectives:

Practice java programming with inheritance and polymorphism.

Task (6 marks)

Implement the Shape hierarchy shown in Fig. 1 (Class name in *italic* indicates an abstract class).

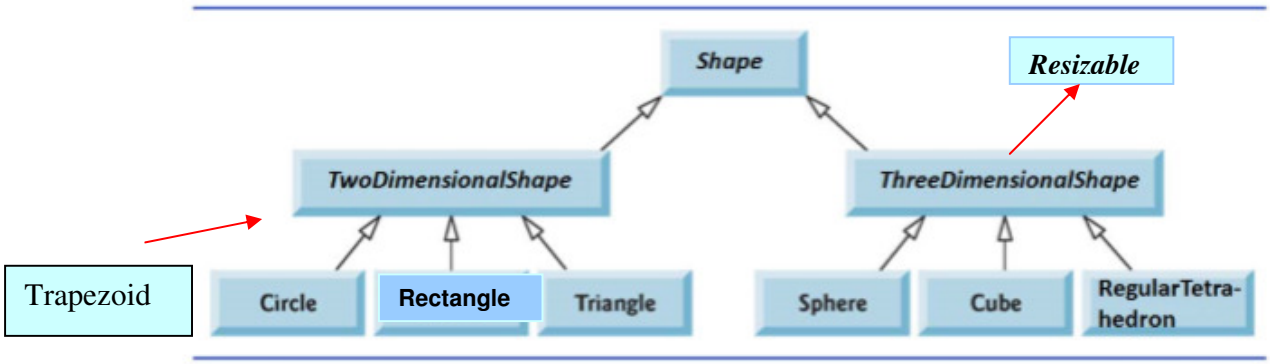


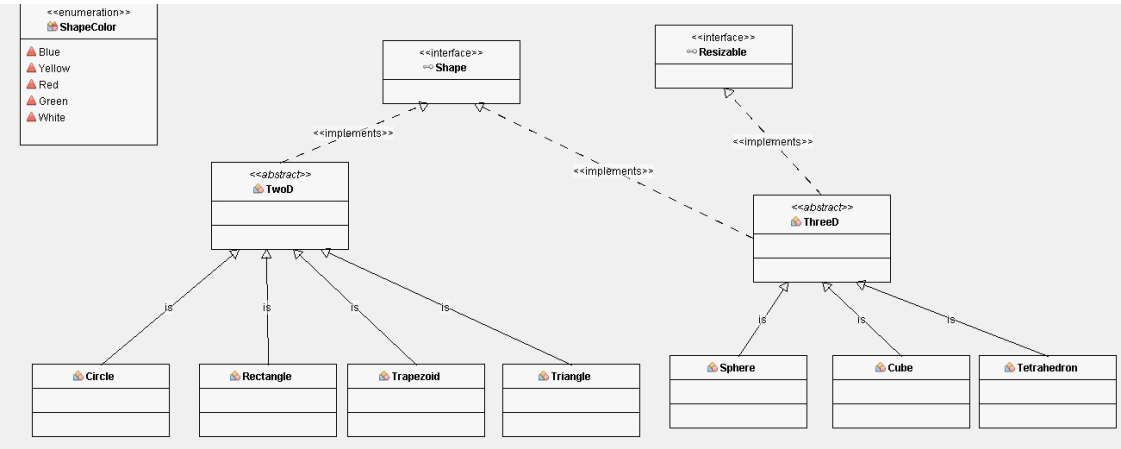
Fig 1: Inheritance hierarchy of Shapes

- Basically each `TwoDimensionalShape` has an `area` method computes and returns the area of the two-dimensional shape. Try to have a private `computeArea` method (do the computation), and area method returns the computed area; though this method was not shown in the UML diagram.

- Each `ThreeDimensionalShape` other than the `area` method we also have a `volume` method; computes and returns, respectively, the surface area and the volume of the three-dimensional shape. Likewise have two private `computeArea` and `computeVolume` methods in the design though these two methods were not shown in the UML diagram.

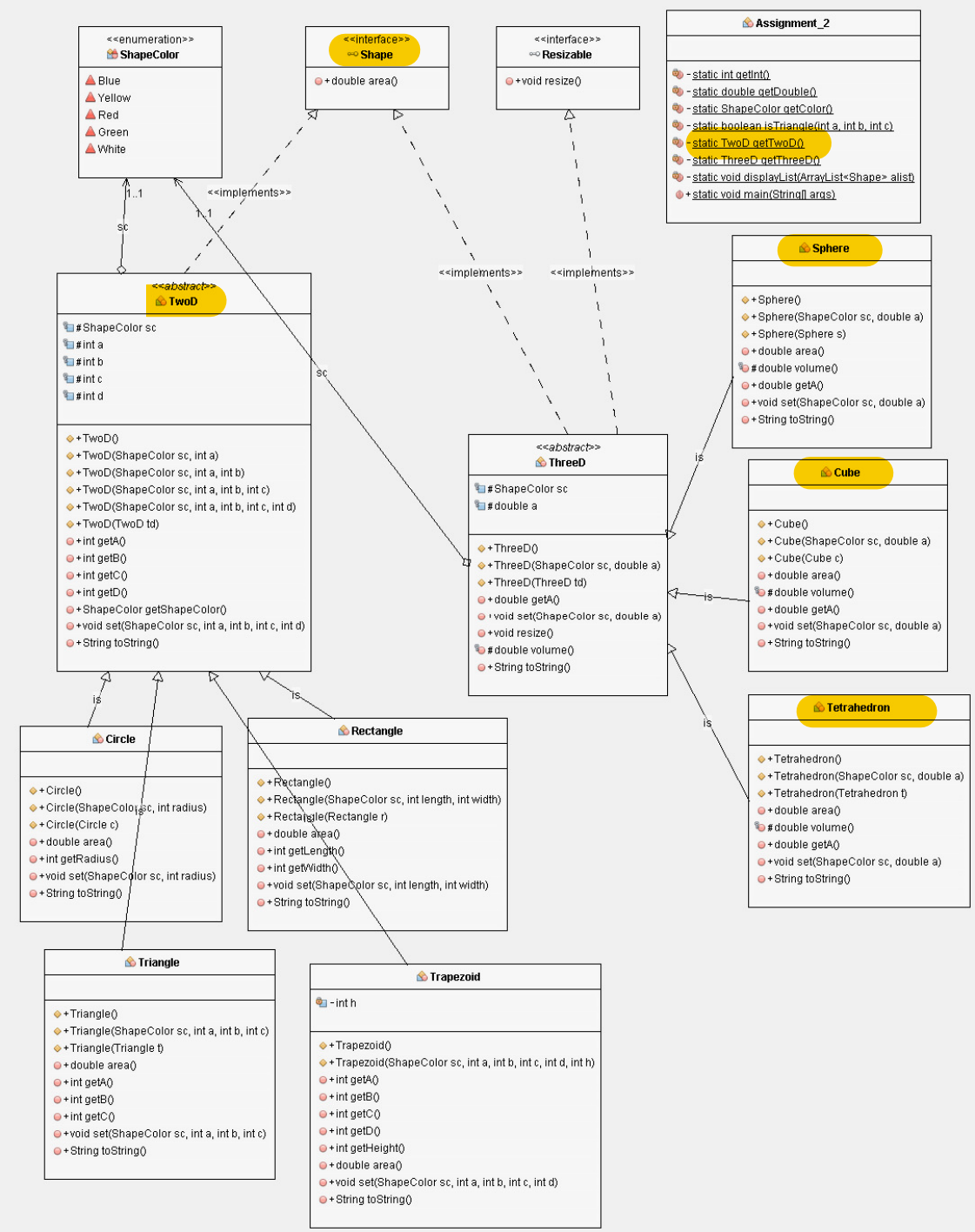
- Try to surf the internet to look for some formulas, for example, to compute the areas, the surface areas and the volumes … I did that too ☺

- The following UML diagram show briefly the overall design:



Let us break the design into 2 parts: 2D and 3D.

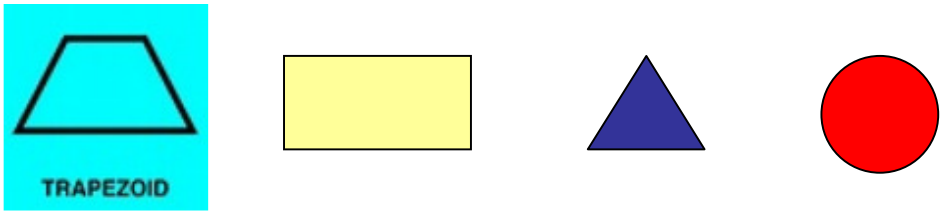
A more detailed UML diagram is shown as follow: (Note that the #'s before the instance variables and the methods' names mean "protected")



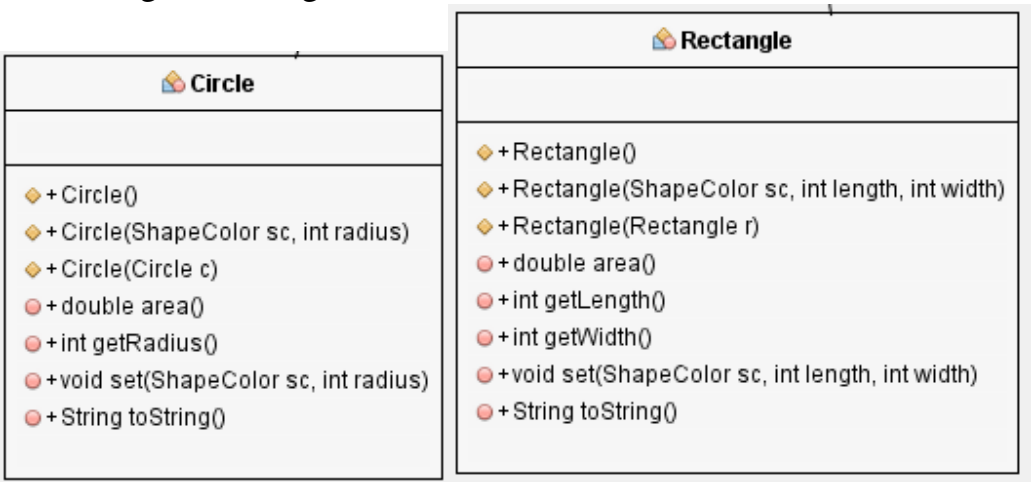
Wow … so difficult to see☺; no worry, I will break it down bit by bit and explain what you have to do …

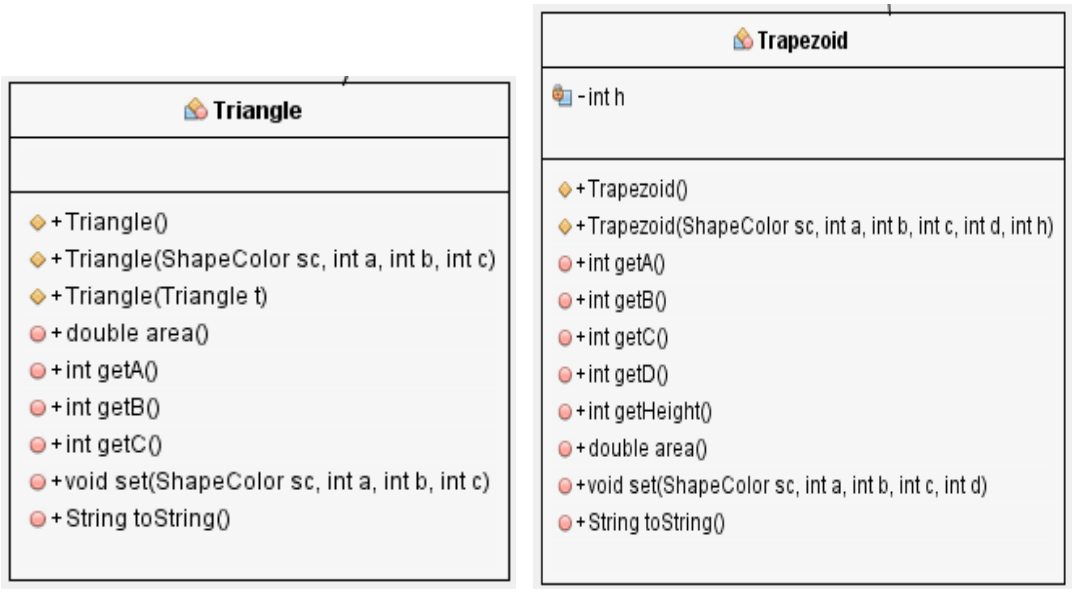
Let us explore the above UML diagram at the highest hierarchy; you can see that `Shape` is an interface class. Two abstract classes `TwoD` and `ThreeD` implement the `Shape` interface; `ThreeD` also implements `Resizable` interface and we also have an enumeration data type to describe some of the colors:

Four possible shapes for `TwoD`: one value, for example the radius, is a circle shape; two values, for example the length and the width, is a rectangle; three values for example the three sides of a triangle (provided it can be formed), five values for trapezoid (one additional value was the height, as we need that to compute the area). In this class, you can see that we have four constructors to describe the four shapes, a copy constructor, some accessor and mutator methods and a `toString` method. Each 2D shape also has a color. You can see we define an enumeration type to specify the color of the shapes.



The following UML diagram shows the four concrete subclasses of `TwoD`:





Some methods just override the super class methods (same implementations)

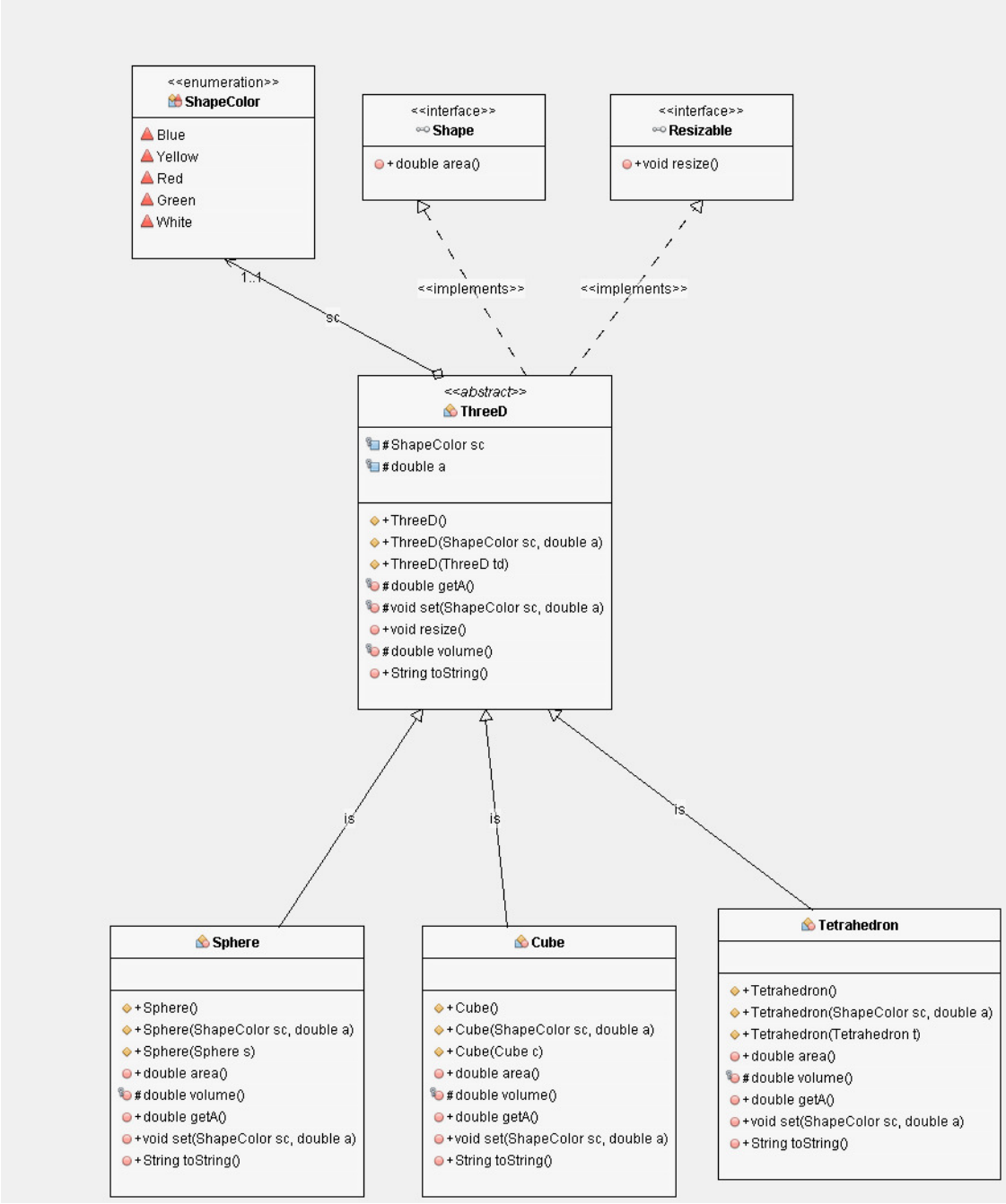
Information defined in each of the subclasses should be obvious in definitions, don't forget the private method that I have just mentioned ☺

Now, let us look at the `ThreeD` class:

Three possible shapes for `ThreeD`: Just one value can determine the shapes of a sphere, a cube and or a tetrahedron. In this class, we only also have constructors, copy constructor, accessor methods, mutator methods and a `toString` method. For a 3D object, we can compute and return the volume too. Therefore we have *one abstract method* in this abstract class `ThreeD`.

`ThreeD` class also implements `Resizable` interface class. In this Resizable interface, other than the method `resize`, to reduce the size by certain *percentage*; therefore in this class, we also have constant specifying the percentage of the size to be reduced.

Let us now look at the UML diagram for the three subclasses of `ThreeD` class:

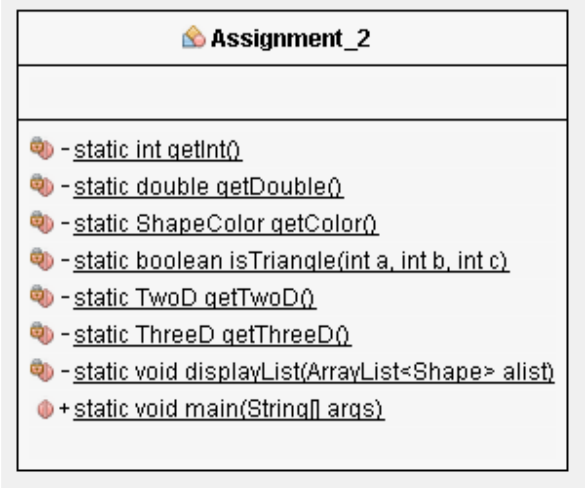


Look for the surface area and volume formulas somewhere in internet to compute and to return their values, also don't forget the two private methods I have just mentioned ☺

Note that all 3D objects need to resize by a certain percentage.

Let us now explore the main class, i.e. main method is defined in this class

All shapes (2D or 3D) should be *randomly generated* and store them in an ArrayList of `Shape's`.



You can see a few private class methods are defined in this class `YourName_A2` (remember to replace by your own name):

- a method generates and returns a positive integer, not too large
- a method generates and returns a positive real number, not too large
- a method generates and returns a color.
- a method generates and returns a TwoD shape
- a method generates and returns a ThreeD shape
- a method to display the objects stored in the ArrayList. Note that in the display method, you display the details of each shape object, i.e. the toString method for each of the classes only display a "brief" object info, display of area / volume / resizable should be done in this method.

Convenient to your design, minor updates to methods or additional methods are allowed.

In the main method, you *repeatedly* generate an integer `k` (0 or 1 or 2). If k is 1 you construct a 2D object; if k is 2, you construct a 3D object and k is 0, you end the task. The following shows one of the possible displays:

```
Shape 1: Rectangle (2D (Red, 5, 8))
Area = 40.000
I am a rectangle shape
------------------------------
Shape 2: Sphere (3D (White, 1.307))
Surface area = 21.475
Volume = 7.158
Size reduced by 10%: Sphere (3D (White, 1.177))
Updated surface area = 17.395
Updated volume = 5.798
I am a sphere shape
------------------------------
Shape 3: Cube (3D (Green, 3.951))
Surface area = 93.646
Volume = 61.661
Size reduced by 10%: Cube (3D (Green, 3.556))
Updated surface area = 75.853
Updated volume = 44.951
I am a cube shape
------------------------------
Shape 4: Trapezoid (2D (Yellow, 5, 1, 1, 4)7)
Area = 21.000
I am a trapezoid
```

Four objects were generated and stored in the array list and you displayed the list. You can see Shape 2 and Shape 3 were 3D objects, their sizes, area, volume were reduced by 10%.

Note that the list may be empty …

IMPORTANT

Put all your classes in a file called **`YourName_ClassListNo_A2.java`** and make sure that this file can be compiled and can be executed. Upload **ONLY** this file to Moodle**. ALL ZIP FILE SUBMISSION WILL BE REJECTED.**

No re-submission will be allowed after grading.

In the above file, remember to put down your name and also the following declaration (some similar contents):

// Tell me if it is your own work, and whether you
// have passed your program to your friends etc etc etc
// and willing to accept whatever penalty given to you.

- **Wrong file name -0.5 mark**
- **No declaration, no name etc -0.5 mark**
- **Failing to demo -1 mark**
- **Programs indentations and alignment of statements -0.5 mark**
- **Late penalty: -0.1 mark per hour.**