

## Lab 11

### Requirements:

- Create a Java project named **yourStudentId\_OOP\_Lab11**
- Read instructions and create classes needed. You are supposed to add 1 interface and 5 classes (*Shape*, *Square*, *Circle*, *Triangle*, *Land* and *Tester*) to the project.
- All instance variables are private. Please use public methods to access private instance variables.

### Description:

Nowadays, the land transaction is very common in the real world, and the total value of land often depends on the land area and the value per acre. In this question, it is assumed that the land area can be measured with squares, circles, and triangles. At the same time, each shape includes its area and perimeter calculation formula and the methods to access these results. In order to make the code more reusable, you should appreciate how interfaces can be used to decouple the classes. Therefore, by using the interface to assist the interaction between classes, you can use multiple shapes to calculate the land area. Figure 1 describes the relationship between each class or interface.

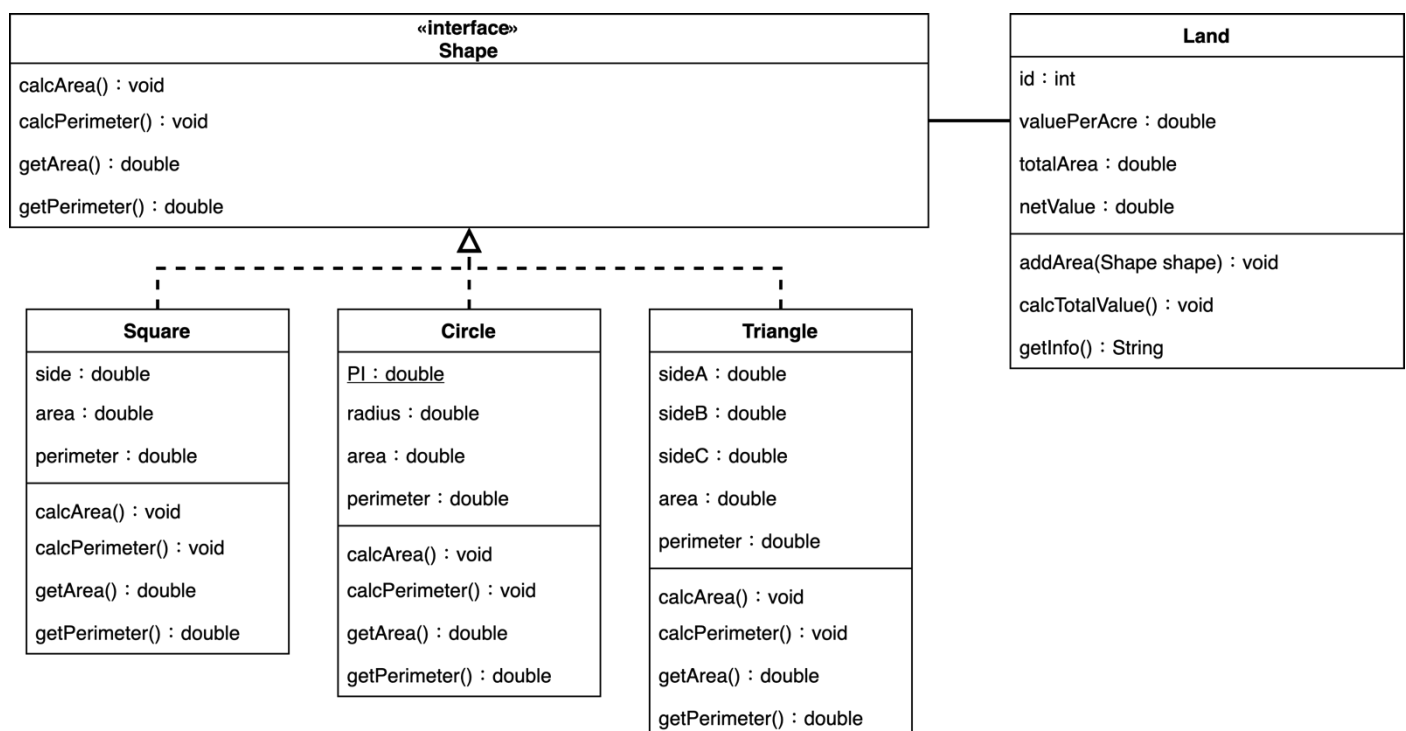


Figure 1. The UML diagram of the question

1. Create *Shape* class

Shape	
Modifier and type	Method (or Variable) and description
<b>Abstract methods</b>	
<b>void</b>	calcArea() The abstract method is used to calculate the area of a shape.
<b>void</b>	calcPerimeter() The abstract method is used to calculate the perimeter of a shape.
<b>double</b>	getArea() The abstract method is used to get the area of a shape.
<b>double</b>	getPerimeter () The abstract method is used to get the perimeter of a shape.

2. Create *Square* class

Square	
Modifier and type	Method (or Variable) and description
<b>Instance variable</b>	
<b>double</b>	side The side of the square.
<b>double</b>	area The area of the square.
<b>double</b>	perimeter The perimeter of the square.
<b>Constructor</b>	
<b>Square(double side)</b> Enable to instantiate the object of <i>Square</i> with given <i>side</i> .	
<b>Instance methods</b>	
<b>double</b>	2 getter for 2 attributes ( <i>getArea()</i> , <i>getPerimeter()</i> ).
<b>void</b>	calcArea Override the <i>calcArea</i> method that interface required.
<b>void</b>	calcPerimeter Override the <i>calcPerimeter</i> method that interface required.

3. Create *Circle* class

Circle	
Modifier and type	Method (or Variable) and description
<b>Instance variable</b>	
<b>double</b>	PI The instance variable is a final variable which cannot be changed.
<b>double</b>	radius The radius of the circle.
<b>double</b>	area The area of the circle.
<b>double</b>	perimeter The perimeter of the circle.
<b>Constructor</b>	
<b>Circle(double radius)</b> Enable to instantiate the object of <i>Circle</i> with given <i>radius</i> and set <i>PI</i> as a constant of 3.14.	
<b>Instance methods</b>	
<b>double</b>	2 getter for 2 attributes ( <i>getArea()</i> , <i>getPerimeter()</i> ).
<b>void</b>	<i>calcArea()</i> Override the <i>calcArea</i> method that interface required.
<b>void</b>	<i>calcPerimeter()</i> Override the <i>calcPerimeter</i> method that interface required.

4. Create *Triangle* class

Triangle	
Modifier and type	Method (or Variable) and description
<b>Instance variable</b>	
<b>double</b>	sideA A side of the triangle.
<b>double</b>	sideB A side of the triangle.
<b>double</b>	sideC A side of the triangle.
<b>double</b>	area The area of the triangle.

<b>double</b>	perimeter The perimeter of the triangle.
<b>Constructor</b>	
<b>Triangle(double sideA, double sideB, double sideC)</b> Enable to instantiate the object of <i>Triangle</i> with given three sides.	
<b>Instance methods</b>	
<b>double</b>	2 getter for 2 attributes ( <i>getArea()</i> , <i>getPerimeter()</i> ).
<b>void</b>	calcArea() Override the <i>calcArea</i> method that interface required. We suggest using Hero's formula to calculate the area of a triangle when the length of all three sides are known.
<b>void</b>	calcPerimeter() Override the <i>calcPerimeter</i> method that interfaces required.

## 5. Create *Land* class

<b>Land</b>	
<b>Modifier and type</b>	<b>Method (or Variable) and description</b>
<b>Instance variable</b>	
<b>int</b>	id The unique number of the land.
<b>double</b>	valuePerAcre The lot value per acre.
<b>double</b>	totalArea The total area.
<b>double</b>	netValue The value of land.
<b>Constructor</b>	
<b>Land(int id, double unitLandValue)</b> Enable to instantiate the object of <i>Land</i> with given <i>id</i> and <i>unitLandValue</i> .	
<b>Instance methods</b>	
<b>void</b>	addArea(Shape shape) Add the calculated area of the shape to <i>totalArea</i> .
<b>void</b>	calcTotalValue() Multiply the total land area by the unit value per acre and assign the calculated result to the <i>netValue</i> .

<b>String</b>	<pre>getInfo()</pre> <p>Return land information as the following example. The result should be rounded off to the 2nd decimal place.</p> <p><b><u>Example:</u></b></p> <p>The area ID: 1</p> <p>Value per acre: \$30000.00</p> <p>Total area: 184.50</p> <p>Total value: \$5535000.00</p>
---------------	---

## 6. The *main* method in *Tester* class

In this section, you should create two instances of *Land* named *florida* and *indiana*. The value per acre of *florida* is US\$30,000 and is numbered as 1. The area is composed of a square with a side length of 10, a circle with a radius of 5, and a right triangle with sides 3, 4, and 5. Moreover, the value per acre of *indiana* is \$17,000 and the number is 2. The land area consists of a square with side length of 5, a circle with a radius of 7, and a regular triangle with a side length of 6. Then, call *calcTotalValue()* of each land to calculate the total value of the land. Finally, call and print the return value of *getInfo()* to the console. The result must be the same as the sample output.

### Sample output

```
<<Florida>>
The area ID: 1
Value per acre: $30000.00
Total area: 184.50
Total value: $5535000.00

<<Indiana>>
The area ID: 2
Value per acre: $17000.00
Total area: 194.45
Total value: $3305623.77
```

**Submission:** Submit your project as “.zip file” via Moodle. No other submissions will be graded.

**Reminder:** Please zip **the whole project**

**Deadline:** Tomorrow’s midnight (for both Mon56 and Tue23)