

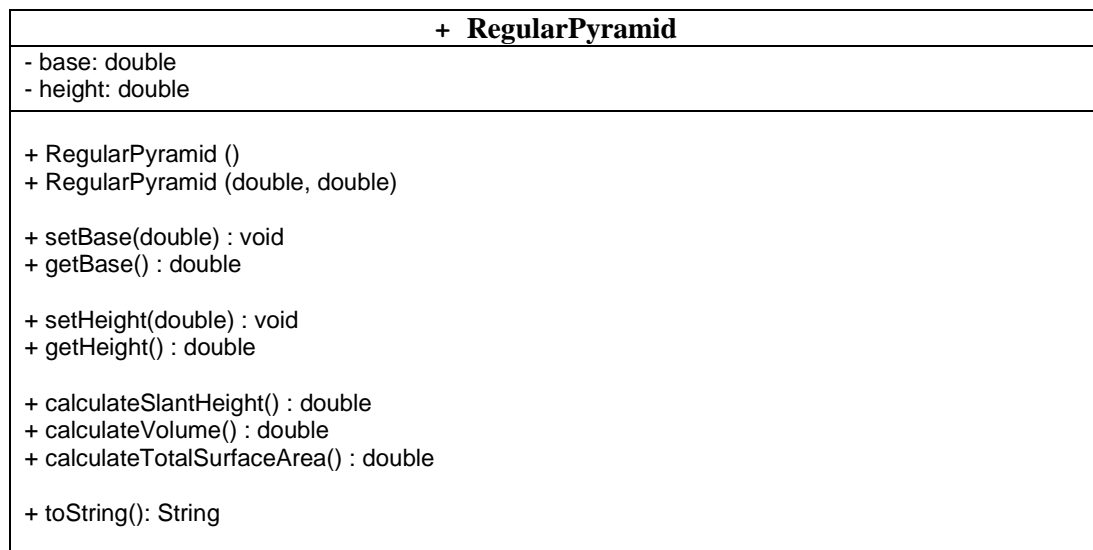
Do not use the reserved keyword “**this**” in the constructors and setters.

1. Rewrite the “**Regular Pyramid**” program using **Object-Oriented Programming** approach.

Step 1: Define **RegularPyramid** class (RegularPyramid.java):

- Two private instance data members: base and height
- Two constructors (one default, one overloaded)
- Get and set methods for each private data member
- An instance method that calculates and returns the slant height
- An instance method that calculates and returns the volume
- An instance method that calculates and returns the total surface area
- toString method

The class diagram is shown as below:



Note: The class diagram is the design document. Your program should match the class name, data members, methods, constructors, variable names, data types, method signatures, and access modifiers (private or public) in the class diagram.

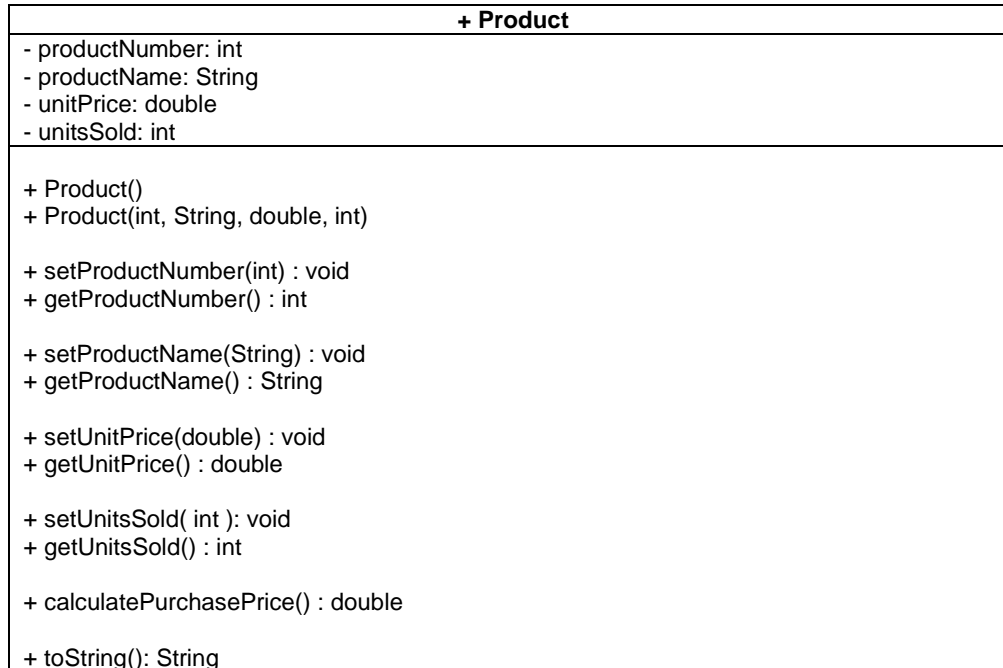
Step 2: Write a client program as follows (RegularPyramidClient.java):

- Prompt the user to enter the base and height.
- Create a **RegularPyramid** object based on the user input.
- Call the methods on this object to get the slant height, volume, and total surface area.
- Print the object information including the base, height, slant height, volume, and total surface area.
- The results should be formatted.

2. Using **Object-Oriented Programming** approach, write a program to calculate the purchase price for purchasing product in retail store.

A product has four instance data members: product number, product name, unit price, and units sold. An instance method can calculate and return the purchase price (unit price * units sold).

a) Define **Product** class based on the following class diagram (Product.java)



Note: The class diagram is the design document. Your program should match the class name, data members, methods, constructors, variable names, data types, method signatures, and access modifiers (private or public) in the class diagram.

b) Write a client program (RetailStore.java) as follows:

- Create a **Product** objects based on the user input.
- Call `calculatePurchase()` method on the object to get the purchase price.
- Print the result (include the product number, product name, unit price, units sold, and purchase price)

Sample Run:

```
Enter Product Number: 1100900
Enter Product Name: Chair
Enter Unit Price: 85.5
Enter Units Sold: 2

*****
Product Number: 1100900
Product Name: Chair
Unit Price: $85.50
Units Sold: 2

Purchase Price: $171.00
```

Due: Wednesday, 11/9/22

- To receive full credit, the source code files must be submitted to D2L by the due date.
- Late submissions will incur a penalty of 5% per day.

Style, form, documentation, naming convention, and more

<p>Each program should have a file header section.</p> <pre> /* * Author: Your name * Date: Date of completion * Assignment: Assignment # NameOfSourceCode.java * Description: The program description */ </pre>	<p>Up to 5% deduction</p>
<p>Each program should be written with the appropriate form and style. Use indentation, blank line, and comments to make the source code easy to read.</p>	<p>Up to 5% deduction</p>
<p>Use Java naming convention and meaningful names to name the classes, methods, variables, constants, and other identifiers in the programs.</p>	<p>Up to 5% deduction</p>
<p>Format the output appropriately</p>	<p>Up to 5% deduction</p>