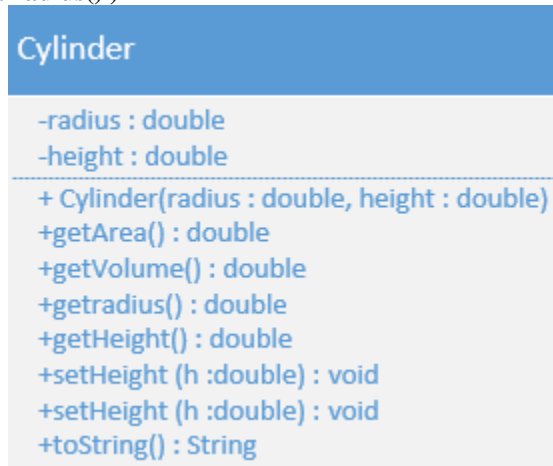


Assignment 3:

- Q1:** Implement the class called *Cylinder* shown in UML below. The constructor accepts and initializes the radius and height for the Cylinder, while accessors and mutators allow them to be changed after object construction. The class also include methods that calculate and return the volume and surface area of the Cylinder. Lastly, it contains a toString method that returns the name of the shape, its radius, its height, its Area and its Volume. Create a main method which instantiates 4 Cylinder objects (any parameters), display them with toString(), change one parameter (your choice) in each, and display them again. [10 points] (**Note: Corrections in diagram. (1) It should have a setRadius method instead of an extra setHeight method. (2) getradius() method needs to be named getRadius())**



- Q2:** Using the Die class (see `Base_A03Q2.java`), design and implement a new class called `PairOfDice`, which uses two Die objects. Include methods to set and get the individual die values, a method to roll the dice, and a method that returns the current sum of the two die values. Rewrite the `SnakeEyes` program (see `Base_A03Q2.java`) using a `PairOfDice` object. [10 points]
- Q3:** Design and implement an application that creates a *histogram* that allows you to visually inspect the frequency distribution of a set of values. The program should read in an arbitrary number of integers that are in the range 1 to 10 inclusive, stopping when the user enters a 0, and warning the user when the input is less than 0 or greater than 10 (i.e., invalid). Once the data has been read, produce a chart similar to the one below that indicates how many input values were equal to 1, 2, 3, and so on. Print one hash symbol for each value entered. [15 points]

Sample output 1:

```
Enter a value to plot: 4
Enter a value to plot: 5
Enter a value to plot: 5
Enter a value to plot: 4
Enter a value to plot: 5
Enter a value to plot: 9
Enter a value to plot: 0

1 |
2 |
3 |

4 |  ##
5 |  ###
6 |
7 |
8 |
```

```
9 | #
10 |
```

Sample output 2 (input values omitted):

```
1 | #####
2 | ##
3 | #####
4 |
5 | ###
6 | #####
7 | ##
8 | #####
9 | #####
10 | #####
```

- **Q4:** Design and implement a set of three classes that define shapes: RoundShape, Sphere, Cone. For each class, store fundamental data about its size and provide methods to access and modify this data. In addition, provide appropriate methods to compute the area, and volume, for Sphere and Cone. In your design, consider how shapes are related and thus where inheritance can be implemented. Don't create duplicate instance variables. Create a main method which instantiates 2 Sphere objects (any parameters), 2 Cone objects (any parameters), display them with ToString(), change one parameter (your choice) in each, and display them again. [15 points]