Homework 2

Baoye Chen

November 2, 2020

Exercise 1

Describe all parts of this class:

12

```
public class Cube {
  // depth, width, and height are three private data fields of the Cube class
  // all of them have the type double
  private double depth = 10;
  private double width = 10;
  private double height = 10;

  // numOfCubes is also a private data field with type int
  private static int numOfCubes = 0;

  // this is a public static method with return type int
  // it returns numOfCubes
  public static int getNoOfCubes() {
      return numOfCubes;
  }

  // this is the constructor of Cube class
  // the numOfCubes would plus one after the constructor is called
  public Cube() {
      numOfCubes++;
  }

  // this is a method that scales the depth, width, and height by a factor
  // the depth, width, and height would be changed when this method is called
  public void scale(double scaling) {
    depth *= scaling;
    width *= scaling;
    height *= scaling;
  }

  // this is a method that gets the volume of the cube
  public double getVolume() {
    return depth * width * height;
  }

  // this is a method that gets the depth of the cube
  public double getDepth() {
    return depth;
  }
```

```
40
    // this is a method that sets the depth of the Cube to the value "depth"
42  // it uses "this." to explicitly reference the data field "depth" of the
      object constructed
    public void setDepth(double depth) {
44    this.depth = depth;
    }
46 }
```

Ex1.java

Exercise 2

Draw an UML diagram of the previous class.

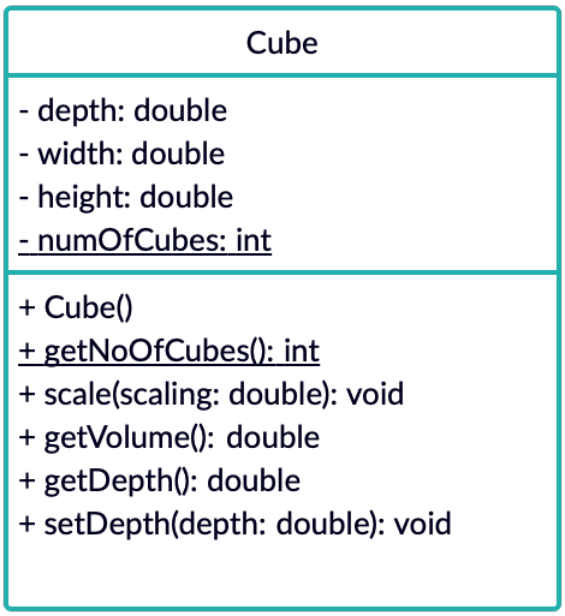

Figure 1: UML of Cube class

Exercise 3

Write a program that reads positive numbers until the user inserts a negative one (Note: the negative number does not count).

After having read the number the program should print:

- Their average
- Their minimum and maximum
- Their standard deviation (square root of the sum of the squared differences from the mean divided by total minus 1)
- Their median (the value separating the higher half of a data)

Note: for computing the last two statistics you need to store an arbitrary number of number. To do that you will need to implement the InfinitArray class.

This class should contain a normal array (double[]) as storage and provide an add(double value) method which insert values inside the storage.

When the storage is full, the add method should create a new storage twice as big and copy all values from the old storage to the new one.
This effects into having the same values inside the storage but its size will be larger. (Hint: store also the current length of the InfinitArray which can be different from storage.length)

You can add other fields, getters (e.g., at(int index)), or methods (e.g., sort()).

Please provide also an UML diagram of your class.

Source code is submitted separately.

Here is the UML of this program:

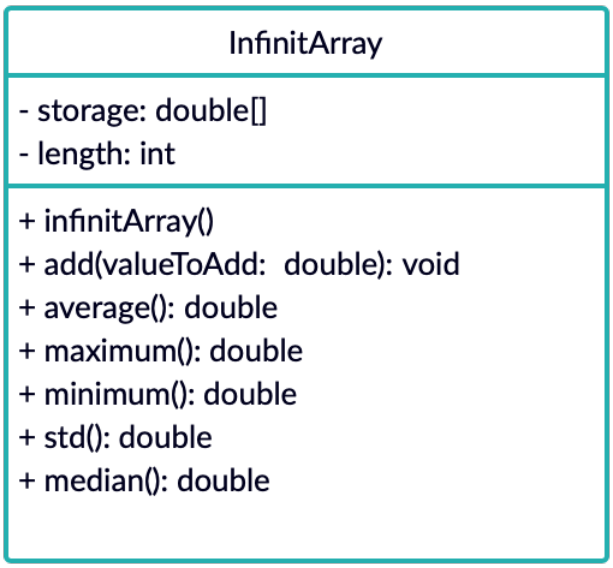

Figure 2: UML of InfinitArray class

Here is the output of the program when the sample input is used:

12

```
baoyechen@BaoyedeMacBook-Air cs101_hw2 % javac InfinitArray.java
baoyechen@BaoyedeMacBook-Air cs101_hw2 % java InfinitArray
Insert numbers (terminate with negative number):
2.1
2.6
1
-1


Their average is 1.9000000000000001
Their min/max is 1.0/2.6
Their standard deviation is 0.6683312551921141
Their median is 2.1
baoyechen@BaoyedeMacBook-Air cs101_hw2 %
```

Trial with sample input.txt

Some trials with other inputs:

12

```
baoyechen@BaoyedeMacBook-Air cs101_hw2 % java InfinitArray
Insert numbers (terminate with negative number):
1
3
2
4.3
1
2.5
33.33
1009.23
33.4
22.1
33.4
15.29
-1


Their average is 96.71250000000002
Their min/max is 1.0/1009.23
Their standard deviation is 275.4404720965131
Their median is 9.795
baoyechen@BaoyedeMacBook-Air cs101_hw2 % java InfinitArray
Insert numbers (terminate with negative number):
-1


Their average is 0.0
Their min/max is 0.0/0.0
Their standard deviation is 0.0
Their median is 0.0
baoyechen@BaoyedeMacBook-Air cs101_hw2 %
```

other trials.txt