Quiz 2 Part 3- JUNIT and MOCKITO

Instructions: Create the corresponding program based on the specifications below. Provide the necessary JUNIT tests for your methods.

General Description of the Program:

The program will accept values that represent quantity of students based from four different university degrees categories: B.Tech, M.Tech, M.S and Ph.D. It will also generate a random number of 1 to 4 representing the four categories. Whichever number is generated, the program will display that that category will be hosting the yearly fair. The program will do the following:

a. Accept the quantity per category
b. Compute total quantity
c. Compute the percentage each of the numeric input got based on the total quantity
d. Generate a random number of 1 to 4 that represent the categories
e. Based on the result, a message will be displayed that states which university degree category was chosen to host the yearly fair.

Program Specifications

1. Create a class that is a simple plain old java object (POJO) that store the data that will be used by other classes. This class contains getter and setter methods. Name this class Degrees
2. Create an interface that will contain two abstract methods which is “add” and “randomize”. These methods will be implemented in another class. The purpose of the add method is to return the total student quantity. The purpose of the randomize is to return a randomize number from 1 to 4. You can provide your own name for this interface.
3. Create a class that will contain the methods on how to implement the program requirements. Name this class DegreeService
4. Create a class that will implement your interface and will contain the implementation logic of the methods. The methods here is the one that you will use in the DegreeService specifically the functionalities of getting the total of all inputted student quantity and the randomize number from 1 to 4.
5. Make sure to complete your DegreeService Class. This is the class where most of the “business logic” of the problem is implemented. This will also be the class that will become your system under test (SUT)
6. Provide some exception handling technique in your DegreeService. One exception handling is enough.

7. Create a class that will have the main method. This class will integrate all the other classes and provide the actual output on the console.
8. Test your program in the main and make sure it is running properly.
9. Your system under test (SUT) will be the DegreeService. Since this class will use the methods that will return the total of students and the generation of random number, which is implemented in another class, you will mock this class using MOCKITO. Please do not stub or use the real class. You will not get any point on the MOCKITO part if you do this. You should have the following tests:
    a) The method that returns the total student quantity.
    b) Method or methods that will return the percentage of each student quantity.
    c) The method that determines who is the host of the year's fair.
    d) A method that will test the exception handling utilized in the SUT.
    e) At least one method that will test an input
10. All tests should be passing.
11. Codes should be aptly documented
12. Program should be running properly and there is no logic or syntax error.
13. Numerical outputs should be shown in two decimal points formats.
14. There should be ample prompts to guide user for inputs.

Sample output

```
Enter the number of B.Tech students
10
Enter the number of M.Tech students
15
Enter the number of M.S students
60
Enter the number of Ph.D students
15
The total student population is 100.0
B.Tech percentage from the total voting population is 10 percent
M.Tech percentage from the total voting population is 15 percent
MS percentage from the total voting population is 60 percent
Phd percentage from the total voting population is 15 percent
MS students will host this year's fair.
```

Submission Instructions:

1. Please name your project with your first name and student number. If you do not follow this requirement, you will have a deduction of one point.
2. Zip and submit your entire project folder.
3. The professor should be able to run the program in one click. The professor will not debug the program if it is not running. Programs should be running as specified by the requirements. Extra movements such as pressing the enter twice to make the output appear will not be considered as valid output.