CS 2401 – Elementary data structures and algorithms

Fall 2022

Lab 3

Due Date: Wednesday September 14, 2022 – end of day.

Objective: The goal of this assignment is to practice with array of objects.

Background: We are given an input text file that contains the dimensions of boxes (i.e., length, width, height) – one box per line. We will analyze the dimension of the boxes and provide a summary of the analyses.

The file, input.txt, will have some lines. Each line will have three values of a box – length, width, height – separated by a space. Your program should read the input file and use the information to create an array of Box objects.

A **sample input** file might be as following:

8.2 17.1 2.9
3.4 4.6 3.1
1.4 19.9 1.6
4.9 6.7 5.3
18.5 2 16.3
4.0 7.2 10.8

Assignment: Boxes have three dimensions (e.g., length, width, height). These dimensions can be used to compute the surface area of a Box. Note that, boxes can have other properties (e.g., color, material), but those are not important for the purpose of this assignment.

Write a Box class in a java file named Box.java following the object-oriented programming principle. The Box class must maintain the following requirements.

1. All status variables (length, width, and height) of the Box class must be private.
2. Write no more than two constructors.
3. Must have a public method named **getSurfaceArea**() that will return the surface area of the box. If l, w, and h are the length, width, and height of a box, the surface area of that box is computed as: 2*(l*w + l*h + w*h).
4. The Box class must NOT contain any main method.
5. The Box class must have a standard **compareTo** method so that two Box objects can be compared based on their surface areas.

You can create any additional method in the Box class if you need.

The program file (the Java file that contains the main method) must be written in a file named **Lab3_Runner_Lastname.java**. **The Runner class must not have any status/field variable**. Runner must have the following functionalities.

1. Read the input text file and **create an array of Box objects**. The sequence of the lines should be used in the sequence of objects in the array (e.g., first line for the first object).
2. Write a **method compute the average surface area** of all the Box objects.
3. Write a **method** that will find the **largest Box** (based on the surface area) **in the array**. The method must report the index, dimensions, and surface area of the largest Box object. Whenever you need to compare two Box objects, you must use the **compareTo** method of one object.
4. Write a **method** that will report the following information.
    a. The average surface area of all the Box objects
    b. The indices of all the Box objects whose surface area is larger than the average surface area (4a).
    c. The index, surface area, and dimensions of the box with the largest surface area.
    d. The number of Box objects in the array whose surface area is more than the average surface area
    e. The number of Box objects in the array whose surface area is less than the average surface area

Sample output:

Average Surface Area: 310.40

Box Indices larger than average surface area: 0, 4
The largest box has index: 4, surface area: 742.3, and dimensions: length is 18.5, width is 2, height is 6.3
Number of boxes with larger than average surface area: 2

Box Indices larger than average surface area: 1,2,3,5
Number of boxes with smaller than average surface area: 4

Test cases:

Provide two test cases showing the program can compute the required information correctly. While writing the test cases, provide a case number, sample input for that case, and expected output. You should have at least one test case for correctness and the other test case that checks for say, negative values in dimensions and throws error message only. Also, give a brief description of what criteria the test case is checking. Write these cases in a txt file. Your program is expected to pass these test cases.

Deliverables: You are expected to submit two files in Blackboard:

(i) Lab3_Runner_Lastname.java, Box.java--- the 2 java files of your program (one containing the main class and the other containing the Box class).

(ii) Lab3_Testcases_Lastname.txt --- the test cases for your program.

Grading Criteria:

- [10 points] Program compiles and runs.
- [5 points] The program is indented correctly.
- [5 points] The program is documented properly.
- [5 points] The program uses correct variable types.
- [5 points] The program uses meaningful variable names.
- [60 points] The program has a correct logic and generates correct output.
  - 15 points for the Box class
  - 45 points for the Runner class
    - 15 points for each method
- [10 points] 2 test cases

- Late submission: [**-10**] points for every 24 hours after the deadline.

If you need any clarification, please ask your TA for further details.