CS 2401 – Elementary data structures and algorithms

Lab: 3

Spring 2023

Due Date: Sunday, February 12 – end of the day.

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

Background: An e-commerce company asked us to write software to help with shipping. The company sells balls of different diameter (e.g., soccer ball, basketball, tennis ball). When an order for a ball is placed, the company needs to find out the most suitable box for shipping.

The box dimensions are stored in a text file. Each line in the file contains the **width, height, and length** of a box. The dimensions are separated by spaces. A sample file is shown below.

```
11.46 23.62 27.31
21.1 23.73 28.83
10.91 18.21 21.36
29.24 27.71 21.99
16.53 17.04 22.64
22.74 20.6 29.12
10.28 19.85 13.66
19.15 11.91 10.32
12.6 12.73 14.06
23.92 10.61 11.22
```

Assignment: Your program should assume that the content is written in a file named input.txt. Each box must be considered an object. To achieve this, you must write a class named `Box`. The Box class is provided below. Do not change the **Box** class. The Box.java file is provided with the assignment too.

```
public class Box {
  private double width, height, length;

  Box(double w, double h, double l){
    width=w;
    height=h;
    length=l;
  }

  public double getWidth(){
        return width;
  }
  public double getHeight(){
        return height;
  }
```

```
  public double getLength(){
        return length;
  }

  public void setWidth(double w){
        width = w;
  }

  public void setHeight(double h){
        height = h;
  }

  public void getLength(double l){
        length = l;
  }

  public double getVolume(){
    return width*height*length;
    }

  public String toString(){
    return "width: "+width+
           "\theight: "+height+
            "\tlength: "+length+
            "\tVolume: "+getVolume();
    }
}
```

The partially written program file (the Java file containing the main method) is provided in a file named Lab3.java. **The instructions are provided as comments in the code. The** Lab3.java file with the following content is provided with the assignment too.

```
import java.io.File;
import java.util.Scanner;
import java.lang.Math;

public class Lab3 {
  public static void main(String[] args){

    Box[] boxes;

    boxes = getBoxArrayFromDataFile(fileName);

    printAllBoxes(boxes);

    //randomly generate the diameter of the ball to be shipped.
    int range = 20;
    int diameter = (int) (Math.random() * range) + 2;


    canContain(boxes, diameter);

    smallestBox(boxes, diameter);

    int l_index = largestBox(boxes, diameter);

    if(l_index >= 0){
            int count = shipHowMany (diameter, boxes[l_index]);
            System.out.println(count+" ball(s) can be shipped in the largest box.");
            System.out.println("---------------------------------------\n");
        }
    else{
            System.out.println("Found no such box");
            System.out.println("---------------------------------------\n");
        }
 }

  /**
   * Complete this method to print the Box
```

```
 * objects in the array parameter theBoxes
 * @param theBoxes is the array of Box objects
 */
public static void printAllBoxes (Box[] theBoxes) {

  // You are not allowed to change the header.
  // Change the body of this method.

  System.out.println("I have not yet implemented the printBoxes method.");

}

/**
 * Change the body of this method to print the box
 * objects in the array parameter that can hold/store
 * the ball whose diameter is given in the parameter.
 * @param theBoxes is the array of Box objects
 * @param diameter of the ball
 */
public static void canContain (Box[] theBoxes, int diameter) {
  // You are not allowed to change the header.
  // Change the body of this method.

  System.out.println("I have not yet implemented the canContain method.");

}

/**
  * Change the body of this method to print the information
  * of the smallest Box object that can hold the ball of
  * the given diameter
  * @param theBoxes is the array of Box objects
  * @param diameter of the ball
  */
public static void smallestBox (Box[] theBoxes, int diameter) {
  // You are not allowed to change the header.
  // Change the body of this method.

  System.out.println("I have not yet implemented the smallestBox method.");

}


/**
 * Change the body of this method to (a) print the information
 * of the largest Box object that can hold the ball of
 * the given diameter, and (b) return the index of that object.
 * @param theBoxes is the array of Box objects
 * @param diameter of the ball
 * @return index of the largest box, -1 if no such box is found.
 */

public static int largestBox (Box[] theBoxes, int diameter) {
  // You are not allowed to change the header.
  // Change the body of this method.

  System.out.println("I have not yet implemented the largestBox method.");

}


/**
* Change the body of this method to return "the number of
* balls" with the given diameter that can be shipped in the largest box
* @param diameter of the ball to be shipped
* @param largestBox is the box Object with largest volume
* @return the number of ball(s) that can be shipped in the
* largest Box object
*/
public static int shipHowMany (int diameter, Box largestBox) {
```

```
    // You are not allowed to change the header.
    // Change the body of this method.

    System.out.println("I have not yet implemented the shipHowMany method.");

  }


   /**
   * Change the body of this method to return an array
   * of Box objects, created after reading the file.
   * @param filename
   * @return the array of Box objects created from the fileName file
   */

  static Box[] getBoxArrayFromDataFile (String fileName){
    // You are not allowed to change the header.
    // Change the body of this method.

    System.out.println("I have not yet implemented the method to \nconstruct the array
from the input file.");

    return null;
  }

}
```

Sample output: The output of a correctly written code for the input provided earlier, and a ball of diameter 13, is shown below.

```
Printing box dimensions.
index: 0 width: 11.46  height: 23.62    length: 27.31    Volume: 7392.412812
index: 1 width: 21.1   height: 23.73    length: 28.83    Volume: 14435.26749
index: 2 width: 10.91  height: 18.21    length: 21.36    Volume: 4243.614696000001
index: 3 width: 29.24  height: 27.71    length: 21.99    Volume: 17817.186396
index: 4 width: 16.53  height: 17.04    length: 22.64    Volume: 6377.035968
index: 5 width: 22.74  height: 20.6     length: 29.12    Volume: 13641.08928
index: 6 width: 10.28  height: 19.85    length: 13.66    Volume: 2787.43228
index: 7 width: 19.15  height: 11.91    length: 10.32    Volume: 2353.74948
index: 8 width: 12.6   height: 12.73    length: 14.06    Volume: 2255.19588
index: 9 width: 23.92  height: 10.61    length: 11.22    Volume: 2847.537264
---------------------------------------
Checking for boxes that can hold the ball.
The diameter of the ball: 13
index: 1 width: 21.1   height: 23.73    length: 28.83    Volume: 14435.26749
index: 3 width: 29.24  height: 27.71    length: 21.99    Volume: 17817.186396
index: 4 width: 16.53  height: 17.04    length: 22.64    Volume: 6377.035968
index: 5 width: 22.74  height: 20.6     length: 29.12    Volume: 13641.08928
The number of boxes that can hold the ball is: 4
---------------------------------------
Finding the smallest box to ship the ball.
Diameter of the ball to ship:  13
Information of the smallest box
  index: 4 width: 16.53 height: 17.04 length: 22.64
---------------------------------------
Finding the largest box to ship the ball.
Largest box that can hold the ball is in index 3 with volume: 17817.186396
---------------------------------------
Finding how many ball(s) can be shipped in the largest box.
4 ball(s) can be shipped in the largest box.
---------------------------------------
```

Requirements: Your task is to change the code in Lab3.java as instructed in the comments to receive the necessary output. You must follow the following requirements**.**

1. You must NOT change Box.java at all.
2. You must NOT change the main method.
3. Do NOT change any of the method headers provided in Lab3.java.
4. Do not use any package other than the ones already imported in Lab3.java.
5. Feel free to write as many new methods as you need in Lab3java.

Note:

- If there is no box that can hold the ball, the ***canContain*** method should display "No box matches the dimension of the ball."
- If there are no smallest box or largest box found, the methods should display "Found no such box"
- The ball will be circular, with a uniform diameter (e.g., there will be no football).
- The diameter of the ball will be between 1 to 20, which will be randomly generated.

Deliverables: You are expected to submit only one Java files (`Lab3.java`) via Blackboard. Please do not submit the input.txt file or the Box.java file.

Grading Criteria:

- [20 points] The Program **compiles and runs**.
- [10 points] The program is **indented** and **documented** properly.
- [10 points] The program uses the correct **variable types** and **names**.
- [60 points] The six methods are implemented correctly.

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

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