

TCSS142 Introduction to Object-Oriented Programming

Programming Assignment 7

DUE: Monday, August 18, 2014 by 12 midnight

This assignment will give you practice with indexing arrays, passing arrays as parameters, and using multiple arrays of unlike data types in parallel.

You will read from the supplied input file "Clients.txt" information about your clients (ID, Weight, & Name), store this information in parallel arrays (just means the same index used in unlike arrays will refer to data that is related to one particular client), perform updates on selected clients' weights and while doing so display the current data on the console, and finally, write the data and some statistics to an output file.

You are supplied with the input file and the main method that will drive this application, i.e. everything in your main is already written and given to you. You will only have to add all the needed methods that main calls plus a few helper methods that these methods will call. Details on main and these methods are as follows:

// DON'T FORGET PROPER DOCUMENTATION HERE

```
import java.io.*;
import java.util.Scanner;
// AND DON'T FORGET PROPER DOCUMENTATION HERE
public class Assign7 {
    public static void main(String[] args) {
        Scanner console = new Scanner(System.in); // For keyboard input
        Scanner input = null; // For file input
        PrintStream output = null; // For file output
        String[] clientNames = null;
        double[] clientWeights = null;
        boolean filesOK = false;
        try {
            input = new Scanner(new File("Clients.txt"));
            output = new PrintStream(new File("ClientOut.txt"));
            filesOK = true;
        }
        catch (FileNotFoundException e) {
            System.out.println(e);
        }
        if (filesOK) {
            // Create the correct number of elements in the clientNames array
            // Note the input.nextInt() used to acquire the size of the array.
            clientNames = new String[input.nextInt()];
            // Create the correct number of elements in the clientWeights array
            clientWeights = new double[clientNames.length];
            readClientData(input, clientNames, clientWeights);
            input.close();
            updateWeights(console, clientNames, clientWeights);
            outputStatistics(output, clientNames, clientWeights);
        }
    }
    // YOUR METHODS WILL GO HERE. You have to use the method headers I have
    // listed on the following pages. You fill in the body of each.
}
```

All lines (except the first) in the input file contain the same type of data: ID Weight Name

The first line simply has a single integer representing how many clients there are and thus, the size of each array. This first integer is read by main and the file marker will begin on the second line of input when execution enters the following method:

```
/**
 * This method reads from the input file starting on the second line, the clients'
 * ID, their weight, and their name. The ID is used as an index into the names and
 * weights arrays.
 *
 * @param in Scanner to the file being read.
 * @param names is an array to hold each client's name.
 * @param weights is an array to hold each client's weight.
 */
public static void readClientData(Scanner in, String[] names, double[] weights) {
}

/**
 * This method allows the user to change the current weights of each client.
 *
 * @param console used for keyboard input
 * @param names is an array to hold each client's name.
 * @param weights is an array to hold each client's weight.
 */
public static void updateWeights(Scanner console, String[] names, double[] weights) {
}

/**
 * This method writes to a file statistic regarding each client. All the client's ID,
 * Name, and Weight are listed in a table followed by statistics on how many clients
 * there are, the Average weight, the highest, and lowest weight.
 *
 * @param output file to which the statistics are written.
 * @param names is an array to hold each client's name.
 * @param weights is an array to hold each client's weight.
 */
public static void outputStatistics(PrintStream output, String[] names,
                                   double[] weights) {
}

// The following methods are helper methods for outputStatistics:
/**
 * This method returns the highest weight found in the array.
 *
 * @param weights is an array to hold each client's weight.
 * @return highest weight found in the array.
 */
public static double getHighest(double[] weights) {
}

/**
 * This method returns the lowest weight found in the array.
 *
 * @param weights is an array to hold each client's weight.
 * @return lowest weight found in the array.
 */
public static double getLowest(double[] weights) {
}

/**
 * This method returns the average of all the weights found in the array.
 *
 * @param weights is an array to hold each client's weight.
 * @return average weight found in the array.
 */
public static double getAverage(double[] weights) {
}
```

Be sure to include proper documentation in the proper places as described on all previous assignments. Not following proper documentation will incur a heavier penalty with this assignment.

SAMPLE RUN

Input file "Clients.txt" (note the first line is consumed by main. You will start reading on the second line):

```
6
2 152 Lea Dea
3 243 Sam Spam
0 148 Ziggy Zip
1 206 Sally Solaride
4 104 Mia Dea
5 125 Tom Tom
```

During execution some prompts and interactions will take place on the console. For example:

```
----jGRASP exec: java Assign7
```

ID.	NAME	WEIGHT
0.	Ziggy Zip	148.00
1.	Sally Solaride	206.00
2.	Lea Dea	152.00
3.	Sam Spam	243.00
4.	Mia Dea	104.00
5.	Tom Tom	125.00

To change a client weight, type the ID (-1 to terminate): 3

Enter a new weight for Sam Spam: 222

ID.	NAME	WEIGHT
0.	Ziggy Zip	148.00
1.	Sally Solaride	206.00
2.	Lea Dea	152.00
3.	Sam Spam	222.00
4.	Mia Dea	104.00
5.	Tom Tom	125.00

To change a client weight, type the ID (-1 to terminate): 0

Enter a new weight for Ziggy Zip: 165

ID.	NAME	WEIGHT
0.	Ziggy Zip	165.00
1.	Sally Solaride	206.00
2.	Lea Dea	152.00
3.	Sam Spam	222.00
4.	Mia Dea	104.00
5.	Tom Tom	125.00

To change a client weight, type the ID (-1 to terminate): 4

Enter a new weight for Mia Dea: 101

ID.	NAME	WEIGHT
0.	Ziggy Zip	165.00
1.	Sally Solaride	206.00
2.	Lea Dea	152.00
3.	Sam Spam	222.00
4.	Mia Dea	101.00
5.	Tom Tom	125.00

To change a client weight, type the ID (-1 to terminate): -1

```
----jGRASP: operation complete.
```

See next page for output file produced "ClientOut.txt" :

Contents of "ClientOut.txt" (BE SURE IT SHOWS THE AVERAGE, HIGHEST, AND LOWEST FIRST BEFORE IT SHOWS THE ENTIRE LIST OF CLIENTS) :

Statistics on our clients:

Total number of clients: 6
Average weight: 161.83
Highest weight: 222.00
Lowest weight: 101.00

ID.	NAME	WEIGHT
0.	Ziggy Zip	165.00
1.	Sally Solaride	206.00
2.	Lea Dea	152.00
3.	Sam Spam	222.00
4.	Mia Dea	101.00
5.	Tom Tom	125.00