

LAB ASSIGNMENT № 1

CS 111 - Spring 2017

Assigned: 01/09/2017

Due: In One Week

Problem Description

Your task is to design a program to help a hospital analyze the flow of patient arrivals in their emergency room. A text file contains patient arrivals 7 days a week, 24 hours a day for 4 weeks. Each value represents the number of patients that entered the emergency room during a given hour. Each row in the data file contains 24 values, one for each hour in the day, for all 7 days of the week. Each row contains 24×7 or 168 values. The first set of values in each row is for day zero or Sunday of each week. Each hour is based on military time with hour 0 being 12am, and hour 23 being 11 pm.

Requirements

Read data from a text file into a 3 dimensional array. From the data, compute:

1. the total number of patients that entered the emergency room per week
2. the total number of patients that entered the emergency room per day for each week
3. the average number of daily patient visits per week
4. the number of patient visits per day, averaged over all weeks
5. the busiest day for each week
6. the least busy day for each week

Each of these requirements directly corresponds to a function you must implement. See the following section on Specifications for more details.

Specifications

Implement the `ERDataReader.java` and `ERDataAnalyzer.java` modules according to the predefined class methods. These predefined class methods serve as the public API (application program interface) for your modules. You are not permitted to change the existing skeleton of class and method declarations. However, you are encouraged to create additional private methods to help simplify the process of implementing the public interface. The `ERDataReader` and `ERDataAnalyzer` classes exist in the `er_data` package.

Reading the data

Implement the `readData()` function of the `ERDataReader` module. The `readData` function does not need to perform any exception handling. In its function declaration, it specifies what exceptions it may throw. It will

be the responsibility of other programs which use, or "consume," this function to handle those exceptions. The ERDataReader module must have the following signature:

Listing 1: ERDataReader Module

```
1 public class ERDataReader {
2     public static int[][][] readData(String dataFile)
3         throws FileNotFoundException, NoSuchElementException, IllegalStateException ↔
4         {}
5 }
```

Analyzing the data

Implement ERDataAnalyzer class interface with the following public methods:

Listing 2: ERDataAnalyzer Module

```
1 public class ERDataAnalyzer {
2     public static int[] patientsPerWeek(int[][][] data) {}
3     public static int[][] patientsPerDayPerWeek(int[][][] data) {}
4     public static double[] averagePatientsPerWeek(int[][][] data) {}
5     public static double[] averagePatientsPerDayAcrossWeeks(int[][][] data) {}
6     public static int[] busiestDayPerWeek(int[][][] data) {}
7     public static int[] leastBusyDayPerWeek(int[][][] data) {}
8 }
```

API Consumption

Note that the class methods are all declared as static. This design decision has certain implications. The primary implication is that the classes do not need to be "instantiated" in order for their methods to be accessed. Notice in the following code block on lines 13 and 20 where the *readData()* and *patientsPerWeek()* methods are invoked directly on the class instead of on an instance of the class. Therefore, you can expect your modules to be used in the following manner:

Listing 3: API Consumption: Example Use Case

```
1 import er_data.ERDataReader;
2 import er_data.ERDataAnalyzer;
3 import java.io.FileNotFoundException;
4
5 public class Assignment1 {
6     public static void main(String[] args){
7         String dataFile = "path/to/data";
8
9         int[][][] data = // initialize array
10        try {
```

```

11     data = ERDataReader.readData(dataFile);
12 } catch (FileNotFoundException e) {
13     System.err.println("File cannot be found or accessed: " + dataFile);
14 } catch (Exception e) {
15     e.printStackTrace();
16 }
17
18 int[] patientsPerWeek = ERDataAnalyzer.patientsPerWeek(data);
19 System.out.println("Patients per week: " + patientsPerWeek);
20 // Calls to other ERDataAnalyzer functions...
21 }
22 }

```

Remember, as the ERDataReader and ERDataAnalyzer modules will be used in the scope of a larger program (which you do not have to write), it is critical to adhere to the API specifications.

Further Instructions

For this lab assignment, you may choose to work with **one** partner.

Deliverables

I will write a program which calls your API and checks the correctness of the output. You are also expected to write readable, well-structured code to the best of your ability.

Each student must submit via ECampus the following:

1. ERDataReader.java - the code for the ERDataReader module
2. ERDataAnalyzer.java - the code for the ERDataAnalyzer module

If you are working in a group, both students should individually submit their versions of the two files. Also, include your teammate's name in a comment at the top of each file.

Submission Deadlines

For Section 11 (Monday Lab): The assignment is due by 6:00pm, Monday January 16th.

For Section 12 (Tuesday Lab): The assignment is due by 3:30pm, Tuesday January 17th.

Late assignments are not accepted.

For assistance, feel free to post questions on Slack (for simpler questions), or email me for more in-depth help.