Activity 7.3 - 2D Array - Soda Survey

Overview

In this activity, you will read table data from a file to populate a 2D array. You will then access all table values in loops to calculate row and column sums and averages.

The Battle Bottle Soda Company is conducting market testing on some new soda flavors like Nuclear Nectarine, Samurai Strawberry, Bananarang, Mixed Martial Mango, etc. They've gathered feedback from passing patrons at several state fairs, where people are generally willing to try anything.

The data from each survey site is in table form in text files, where each row is one person's rating of sodas on a scale of 1 to 10, where 1 corresponds to "Call 911..." and 10 is "Take my money, now!" Each column represents one of the sodas. The first line of the file has two integers, giving the number of rows (people) and columns (sodas) in the file to facilitate reading the table into a 2D integer array for processing.

For example, the following file represents the subsequent table:

```
3 4
2 4 1 7
3 3 2 4
1 5 2 5
```

	Soda 0	Soda 1	Soda 2	Soda 3
Person 0	2	4	1	7
Person 1	3	3	2	4
Person 2	1	5	2	5

The company can't afford a professional statistician who knows fancy terms like 'variance' and 'normalization', so you will be doing some initial processing of the data to get simple averages.

Program output should closely resemble:

Person 0 Mean: 3.50
Person 1 Mean: 3.00
Person 2 Mean: 3.25

Soda 0 Mean: 2.00
Soda 1 Mean: 4.00
Soda 2 Mean: 1.67
Soda 3 Mean: 5.33

Instructions

Getting Started

1. Create an "`Activity7.3`" directory in VS Code.
2. Import `SodaSurveyInterface.java`
3. Import sample input files `Survey1.dat` and `Survey2.dat`.
4. Create a `SodaSurvey` class that implements `SodaSurveyInterface`.
5. Create a `SodaSurveyDriver` class containing the `main()` method.

Part 1: Write SodaSurvey class

The `SodaSurvey` class will contain and process a 2D int **array** with rows corresponding to people and columns corresponding to sodas, as read from an input text file. The 2D array should be the ONLY instance variable. Use `.length` in all code to get array dimensions. There should be no ArrayLists anywhere in this program.

`SodaSurvey` must implement `SodaSurveyInterface` with its class header as follows:

```
public class SodaSurvey implements SodaSurveyInterface { }
```

Implement all methods as described in the `SodaSurveyInterface` method javadocs.

`SodaSurvey` must have one constructor that takes in an input filename as a `String` argument:

```
public SodaSurvey( String filename ) { }
```

Use a `Scanner` to read in each row. You have a choice for how to process each line. You can use a second `Scanner` or you can use `String .split()` to break each row into a `String[]` of column values. Convert `String`s to ints with `Integer.parseInt()`.

The `toString()` method should return a `String` formatted as shown in the Overview, above, and in the interface javadoc for `toString()`. Format averages to two decimal places.

Part 2: Write SodaSurveyDriver class

`SodaSurveyDriver` contains the program's `main()` method. It must expect an input filename passed as a command-line argument to `String[] args`. There should be NO other user interaction via prompts or input from keyboard `Scanner`s.

Validate that you received a command line argument in `args`. If you did not get a command line argument, report the problem and exit.

Create a `SodaSurvey` object using the filename. Test the functionality of all `SodaSurveyInterface` methods.

Sample Sessions

$ java SodaSurveyDriver
Usage: java SodaSurveyDriver filename
$

$ java SodaSurveyDriver Survey1.dat
Testing rowAvg(0): 3.5
Testing rowAvg(1): 3.0
Testing colAvg(0): 2.0
Testing colAvg(1): 4.0

```
Testing toString():
Person 0 Mean: 3.50
Person 1 Mean: 3.00
Person 2 Mean: 3.25

Soda 0 Mean: 2.00
Soda 1 Mean: 4.00
Soda 2 Mean: 1.67
Soda 3 Mean: 5.33
$
```

```
$ java SodaSurveyDriver Survey2.dat
Testing rowAvg(0): 2.0
Testing rowAvg(1): 3.3333333333333335
Testing colAvg(0): 2.0
Testing colAvg(1): 4.0

Testing toString():
Person 0 Mean: 2.00
Person 1 Mean: 3.33
Person 2 Mean: 3.33
Person 3 Mean: 1.33
Person 4 Mean: 3.00

Soda 0 Mean: 2.00
Soda 1 Mean: 4.00
Soda 2 Mean: 1.80
$
```

Terminology Identification

In your code add comments identifying examples of the following: two-dimensional array, row, column, iterating over 2D array. These should be identified with an @keyterm tag within the comment.

Code Review

When you are finished with this activity, pair up with a classmate and review each other's code to make sure it meets all the requirements.

Submission

After completing the assignment, use the assignment link in Canvas and follow the submission instructions there. You will upload your .java files and put your reflection in the "Comments" box.

Reflection Requirements

Write a one paragraph reflection describing your experience with this activity. The reflection should also include the name of your code review partner AND something interesting you found in their code. Please review the activity rubric for more details.