

Introduction

Purpose: This application will permit a user to track rainfall amounts for an entire year. It will track rainfall by day and will provide some summary functions as well. Input will be via the keyboard and output will go to the screen console.

Background

A number of areas of the country, including much of the Southeast, are in the middle of droughts. Rainfall is important to maintaining healthy community water supplies. In addition, farmers and others working in agriculture are concerned about their growing seasons and the amount of rainfall and the patterns of rainfall to facilitate growing crops. This application will begin to build an application that would enable us to track and report rainfall amounts and summaries for a given period of time.

Existing program components

Your application will be written from scratch with no existing components. Your application will consist of 3 Java classes, [RainfallDriver](#), [RainfallIO](#) and [Rainfall](#). These classes are described below.

General Program Behavior

When this application begins, it will first provide a brief introduction, "Welcome to the Rainfall Tracker". This statement will be followed by two newline characters (or one full blank line).

Next, you will prompt the user to begin entering rainfall amounts. Input is in the form of MM, DD, R.RR where MM is the month number (1 based), DD is the day (1 based) and R.RR is the rainfall amount for that day. Rainfall amount is entered on a different line. There is one prompt and then the user will enter lines until finished. The sentinel value for this application is a 0 for the month. A 0 month should cause the input to stop and the processing and output to begin.

After the input is entered, the application will display a report which will consist of the following parts:

1. Annual rainfall: followed by the annual rainfall amount.
2. Monthly rainfall. On each subsequent line will be a month name and the rainfall total for that month.
3. Wettest month. On the subsequent line a list of all months with the same highest total will be printed (in month number order) followed by the rainfall amount.
4. Driest month: On the subsequent line a list of all months with the same lowest total will be printed (in month number order) followed by the rainfall amount.
5. Highest rainfall: On the subsequent line the month name and day with the highest rainfall amount should be displayed.

Output

There will be three kinds of output, introductory prompts, error prompts and the final rainfall report.

Initial Prompts: The application should display: "Welcome to the Rainfall Tracker" as the first line of the output. This should be followed by a blank line.

Then the application should display: "Enter rainfall amounts (m, d, total) where m is

the month number, d is the day, and total is the rainfall amount." On the next line, "Enter 0 to end." This will display only once (at the beginning or when an error is encountered) and then the user will enter each day's entry on a single line.

Error Messages: There can be several errors:

1. If at any point, a user enters a non-integer value for the month and day or a non-numeric value for the total, you should display the message: "Your entry " followed by the full entry line followed by " contains an invalid character." This message should be followed on a new line by the initial prompt line.
2. If at any point, a user enters a month that is out of range, you should display the message: "Your entry " followed by the full entry line followed by "contains and invalid month. Months must be in the range of 1 - 12 or 0 to end the entry." This message should be followed on a new line by the initial prompt line.
3. If at any point, a user enters a day that is out of range for that month, you should display the message: "Your entry " followed by the full entry line followed by "contains an invalid day. " followed by the name of the month followed by " contains " followed by the number of days followed by "days. Please reenter." This message should be followed on a new line by the initial prompt line.
4. If at any point, a user enters a rainfall amount that is negative for the rainfall amount, you should display the message: "Your entry " followed by the full entry line followed by "contains a negative rainfall amount. Please reenter." This message should be followed on a new line by the initial prompt line.

Rainfall Report: The rainfall report should print as follows:

0th line - blank
1st line - "Annual Rainfall Report"
2nd line - blank
3rd line - "Annual Rainfall: " followed by the total rainfall.
4th line - blank
5th line - "Monthly Rainfall"
6-17th lines - tab space followed by the month name followed a colon followed by a space followed by the rainfall total.
18th line - blank
19th line "Wettest Month: " followed by the name of the month with the highest total followed by a space and the rainfall amount.
20th line "Driest Month: " followed by the name of the month with the lowest total followed by a space and the rainfall amount.
21st line "Highest Rainfall: " followed by the name of the month and the numeric day of the day with the highest rainfall amount followed by that amount.
22st line blank
23rd line - "End of Rainfall Report" this line should be followed by a single newline character.

Specifications for each of the classes

RainfallDriver - This class's role is to manage the activity of the other two classes. It will have the main method and will call appropriate methods in each of the classes.

Attributes: none

Methods: main (standard header) This method should control the action of the application.

RainfallIO- RainfallIO's role will be to do all of the input and output processing. All input will be done by this class and all output will be performed by this class. **IMPORTANT**, this class should declare a single Scanner object for keyboard input and then instantiate it only once inside the constructor.

Attributes:

- a keyboard Scanner object. Optionally, you may use a formatting object.
- A Rainfall object.

Methods: Minimum required methods

- enterRainfallData - will allow user to enter data and update the Rainfall object appropriately.
- printRainfallReport - will print the information required by the application.

You may choose to add other methods for error detection and correction and may want to break down the report into submethods.

Rainfall

Attributes:

- A rainfall matrix (two dimensional array). The rows represent the individual months, the columns represent the days in each month. Note: this will be a "ragged" array since each month will have a different number of days. For our purposes, we will assume we are dealing with 2008 which is a leap year.
- Optionally, you may want attributes for the names of the months and the number of days in each month.

Methods:

- a constructor
- an updateDailyRainfall method (will take in a month, day, and rainfall amount and will update the rainfall for that day. Note it will also replace any previously entered amount for that day. Returns true if the month, day, and rainfall values were okay...false otherwise.)
- getAnnualRainfall (will produce the sum of all rainfall amounts for the entire year).
- getMonthlyRainfall(will take in a parameter representing a month and will return the sum of that month's rainfall. Return -1 if the month number is invalid.)
- getWettestMonth(will return a String representing the names of all the months with the same highest rainfall amount separated by commas).
- getWettestAmount(will return a double representing the wettest rainfall amount for a month.)
- getDriestMonth(will return a String representing the names of all of the months with the same lowest rainfall amount separated by commas).
- getDriestAmount(will return a double representing the driest rainfall amount for a month.)
- getHighestDay(will return a string representing the day with the highest single rainfall amount.)
- getHighestAmount(will return a double representing the single highest rainfall amount.)

You may add any additional helper methods that you find useful.

Additional Program Requirements

1. You may add additional methods, but these should be private "helper" or utility methods. See the professor if you feel the need to add additional public methods.
2. Rainfall amounts should always print with three decimal positions.
3. Month numbers and days are 1 based. In other words a user will enter January as month 1 and the first day of the month as day 1.
4. Your program must include an acknowledgement section acknowledging help received from TAs or reference sources. You do not need to reference receiving help from the instructor.
5. If you received no help your acknowledgement section should have a statement to that effect.
6. Your program must conform to standard Java programming standards and the additional requirements for this class. See the [Style Guide](#) for this class.
7. You must use a two dimensional array for this program.
8. You do NOT need to do any exception handling in this program. Use PREVENTION instead (see Scanner methods). I will take off if you handle bad input with exception handling.

Hints

- Begin early. Class time will be provided to begin the design process. USE IT and then get the stubs done timely.
- The Scanner class provides options to scan a String or other input type. See the lab from Monday for an example. This Scanner can be separate from the keyboard Scanner and will not affect input to submit.
- Be sure to review your final program for conformance to the Style Guide which is available under the Resource tab.