CS 1050, Mr. Kramer Programming Assignment #8 # points: 70

Purpose Calculate mileage reimbursements using arrays and methods.

Due Date Per the Course at a Glance. Cannot be resubmitted.

Submissions In this order: printed copies of the source code with line numbers, the input file, and the output file.

Java Topics Arrays, Java I/O, while loops, if-else-if, methods, summing/averaging data.

Use of leftPad in the Toolkit class.

References Textbook – use the index for relevant topics.

The programming standards and guidelines as discussed in class.

Toolkit.java

Specification

Convert the previous assignment using the mileage table to a program that uses arrays. You will use two parallel one-dimensional arrays of length n, where n is the number of mileage values to process. Note that n is determined by reading the first entry in the input file which has the number of entries that follow.

The main purpose of this assignment is to practice using arrays in methods. Therefore, no arrays or related counts and averages should be declared globally (static before the main program begins.) You can declare the Toolkit and constants like tab and new line as global objects if you want.

**All output is written to the output file and to the console.**

One array will hold the mileage and the other the corresponding reimbursement amount. Use the same input file as for the previous mileage program assignment but rename it to this assignment number. Your program must satisfy the requirements of the previous program, plus the items below. The main program should be mostly (but not necessarily only) method calls with appropriate parameters.

1. A method that explains the program to the user.

2. A method with the mileage array as a parameter that reads the mileage values and stores them in the appropriate array. (Use a ‘for’ loop that includes a test for the end of file and the number of elements allocated to the array.) This method doesn’t calculate the reimbursement; it only reads the data. From this method, determine the number of elements in the possibly partially-filled array and return that number.

3. A method with the two arrays and the number read as parameters that calculates the array of reimbursement amounts. A ‘for’ loop must be used for the calculations. If the mileage value is negative or 0, the corresponding reimbursement amount should be zero.

4. A method prints the heading and another method prints the detail lines in the table. The latter method will have the two arrays and the number read as parameters. Use a ‘for’ loop. Detail lines are as specified in the previous mileage program assignment.

Each detail line of the table contains the number of miles (real, print with one decimal place) and the reimbursement amount (real, print with two decimal places). If the input value is <= 0, output five stars instead of the reimbursement amount. As in the earlier assignment, use the leftPad method to format the numbers to one or two decimal places as needed.

5. After the array of reimbursements has been calculated, a method or methods calculate the average reimbursement and the average number of miles traveled. A ‘for’ loop must be used in doing the calculations. These averages will be the averages for mileage values which are > 0. In other words, you might divide by a number less than the number of elements (which, by the way, could be 0, so check for it before you calculate the average).

6. A method outputs the summary information at the end of the table. This output should include all the output specified in the previous mileage assignment (the total of the reimbursement values, the number of mileage values processed, and the number of mileage values that were > 0). In addition, output the total of the mileage values that were > 0, and the two averages calculated in the previous step, all with appropriate messages.