

```

1  /* Mileage reimbursement program for Mathematical Association of America
2
3  This programs purpose is to take the data in a file starting
4  with how many data values are in the file followed by that number
5  of mileages data values. The program reads these data values
6  and stores them into an array. It will perform calculations on
7  the array to create a reimbursement array.
8  We are given the base amount, and rates that should be reimbursed
9  to drivers on a scale for how far they drove.
10 The program uses methods to do all calculations, input/output to
11 files, and output to the console.
12
13 Zachary Stall
14 Program #8, CS 1050, Section 2
15 jGRASP, Custom PC, Windows 10
16
17 Aplomb - Self-confident assurance, skill, and poise - especially
18 in difficult or challenging circumstances.
19
20 "If you don't think every day is a good day, just try missing one."
21 -Cavett Robert (1907 - 1997)
22 */
23
24 import java.util.Scanner; // For console input
25 import java.io.*;         // Access PrintWriter and related classes
26
27
28 public class ZacharyStall_2_08 {
29
30     static Toolkit tools = new Toolkit();
31     static Scanner console = new Scanner(System.in);
32
33     public static void main (String [] args) throws IOException {
34
35         // Access the input/output method
36         final String INPUT_FILE = "ZacharyStall_2_08_Input.txt";
37         final String OUTPUT_FILE = "ZacharyStall_2_08_Output.txt";
38
39         int totalPosVal = 0; // Number of positive mileages
40         int dataValue = 0; // Number of data values
41         int nRead = 0; // Number of values read from input file
42
43         double[] miles; // Mileage array to calculate Reimb
44         double[] reimb; // Reimb array to stroe calculated reimb
45         double milesAverage = 0.0; // mileage average
46         double reimbAverage = 0.0; // reimburseement average
47
48         double totalMileage = 0.0; // Sum of all the miles
49         double totalReimb = 0.0; // Sum of all reimbersement
50
51         // Access the input/output files
52         File inputDataFile = new File(INPUT_FILE);
53         Scanner inputFile = new Scanner(inputDataFile);
54
55         FileWriter outputDataFile = new FileWriter(OUTPUT_FILE);
56         PrintWriter outputFile = new PrintWriter(outputDataFile);
57
58         // Begin program execution
59         System.out.println("Reading file " + INPUT_FILE + "\r\n" +
60                             "Creating file " + OUTPUT_FILE + "\r\n");
61
62         // dataValue is the first number in the input file, number of data values
63         dataValue = inputFile.nextInt();
64
65         // Establish the length of the arrays
66         miles = new double [dataValue];
67         reimb = new double [dataValue];
68

```

```

69         // Get the mileages into the mileage array and check how many data points
70         // are read from the input file.
71         nRead = getMileageArray(inputFile, miles);
72
73
74         // Ensure the # of mileages read is the same as the number of pairs
75         // indicated at the start of the input file.
76         if (dataValue != nRead) {
77             System.out.println(
78                 "The # of entries in the input file doesn't match header value." +
79                 "\nProgram terminated.");
80             System.exit(0);
81         } // End if
82
83
84         // Calculate the reimb amount and store in in the reimb array
85         calcReimbAmount(miles, reimb, nRead);
86
87         // Prints the headers for the table
88         displayHeader(outputFile);
89
90         // Prints the detail lines in the table
91         displayTableData(outputFile, miles, reimb, nRead);
92
93         // Calculate the average miles driven and average reimbursement
94         milesAverage = getAverages(miles, nRead);
95         reimbAverage = getAverages(reimb, nRead);
96
97         // Calculate the sums of miles and reimbursements
98         totalMileage = getSum(miles, nRead);
99         totalReimb    = getSum(reimb, nRead);
100
101         // Caluclate total postive mileage values processed
102         totalPosVal = getPositiveMiles(miles, nRead);
103
104         // Using methods to output formatted data to the console and output file
105         outputData(outputFile,
106             totalReimb,
107             totalMileage,
108             reimbAverage,
109             milesAverage,
110             dataValue,
111             totalPosVal);
112
113
114         inputFile.close();
115         outputFile.close();
116
117         System.exit(0);
118     } // End Main
119
120     // *****
121
122     // Method for headers
123     public static void displayHeader(PrintWriter output) {
124         String str;
125         String str2;
126         str = "This program uses an input file of mileage and " +
127             "the reimbursement amount." + "\r\n" +
128             "It stores the data into arrays." + "\r\n" +
129             "The program uses methods to perform operations on " +
130             "the arrays." + "\r\n" +
131             "Then it prints the information to the " +
132             "user on the console," + "\r\n" +
133             "and into an output file.\r\n\r\n";
134
135         str2 = tools.padString("Mileage", 10, " ", "") +
136             tools.padString("Reimbursement", 20, " ", "") +

```

```

137         "\r\n" +
138         tools.padString("-----", 10, " ", "") +
139         tools.padString("-----", 20, " ", "") +
140         "\r\n";
141
142     System.out.print(str);
143     output.print(str);
144
145     System.out.print(str2);
146     output.print(str2);
147 } // End headers
148
149 // *****
150
151 // Method to read and store the mileages into an array
152 public static int getMileageArray(Scanner input, double[] array) {
153     int len = array.length;
154     int nRead = 0; // number of values read
155
156     while (input.hasNext() && nRead < len) {
157         array[nRead] = input.nextDouble();
158         nRead++;
159     }
160     return nRead;
161 }
162
163 // *****
164 // Method to Calculate reimbursement amount
165 public static void calcReimbAmount(double[] arrayMiles,
166                                   double[] arrayReimb,
167                                   int nElements) {
168
169     double mileDriven = 0.0;
170     double reimbMoney = 0.0;
171     double rate = 0.0;
172     double base = 0.0;
173     double overage = 0.0;
174
175     for (int i = 0; i < nElements; i++) {
176
177         mileDriven = arrayMiles[i];
178
179         // Checks to see if mileage is less than zero...
180         // if it is, will print five stars for reimbursement
181         if (mileDriven <= 0) {
182             arrayReimb[i] = 0;
183             continue;
184         }
185
186         // Checks for miles and assigns appropriate values for base, rate, and overage
187         else if (mileDriven < 400) {
188             base = 0; rate = 0.18; overage = mileDriven;
189         }
190         else if (mileDriven < 900) {
191             base = 65; rate = 0.15; overage = mileDriven - 400;
192         }
193         else if (mileDriven < 1300) {
194             base = 115; rate = 0.12; overage = mileDriven - 900;
195         }
196         else if (mileDriven < 1900) {
197             base = 140; rate = 0.10; overage = mileDriven - 1300;
198         }
199         else if (mileDriven < 2600) {
200             base = 165; rate = 0.08; overage = mileDriven - 1900;
201         }
202         else {
203             base = 195; rate = 0.06; overage = mileDriven - 2600;
204         }

```

```

205
206         // Calculate and output the reimbursement amount and calculate running totals
207         reimbMoney = base + (rate * overage);
208
209         // Stores the Reimb amount into the reimb array
210         arrayReimb[i] = reimbMoney;
211     } // End loop
212
213 } // End calcReimbAmount method
214
215
216 // *****
217 // Method to display table
218 public static void displayTableData(PrintWriter output,
219                                     double[] mileage,
220                                     double[] reimbAmount,
221                                     int nElements) {
222
223     double mileDriven = 0.0;
224     double reimbMoney = 0.0;
225     String mileReimbStr;
226
227     for(int i = 0; i < nElements; i++) {
228
229         mileDriven = mileage[i];
230         reimbMoney = reimbAmount[i];
231
232         if(mileDriven <= 0) {
233             System.out.println(tools.leftPad(mileDriven, 10, "##,##0.0") +
234                               tools.padString("*****", 20, " ", ""));
235             output.println(tools.leftPad(mileDriven, 10, "##,##0.0") +
236                           tools.padString("*****", 20, " ", ""));
237             continue;
238         } // End if statement
239
240         // Output the table of data to the counsole and the output file
241         mileReimbStr =
242             tools.leftPad(mileDriven, 10, "##,##0.0") +
243             tools.leftPad(reimbMoney, 20, "$#,##0.00");
244
245         System.out.println(mileReimbStr);
246         output.println(mileReimbStr);
247     } // End for loop
248
249 } // End displayTableData
250
251 // *****
252 // Calculate the average of miles and reimb
253 public static double getAverages(double[] array, int nElements) {
254
255     int dataPoints = 0;
256     double sum = 0.0;
257     double average = 0.0;
258
259     for(int i = 0; i < nElements; i++) {
260         if(array[i] > 0) {
261             dataPoints++;
262             sum += array[i];
263         } // End if statement
264
265         if (dataPoints != 0) {
266             average = (double)sum / dataPoints;
267         } // End if statement
268     } // End for loop
269     return average;
270 } // End getAverages method
271
272

```

```

273 // *****
274 // Calculate the average of miles and reimb
275 public static double getSum(double[] array, int nElements) {
276     double sum = 0.0;
277
278     for(int i = 0; i < nElements; i++) {
279         if(array[i] >= 0) {
280             sum += array[i];
281         } // End if
282     } // End for
283     return sum;
284 } // End getSum
285
286 // *****
287 // Sum the number of postive mileage values in the mile array
288 public static int getPositiveMiles(double[] array, int nElements) {
289     int sum = 0;
290
291     for(int i = 0; i < nElements; i++) {
292         if(array[i] > 0) {
293             sum ++;
294         } // End if
295     } // End for
296     return sum;
297 } // End getSum
298
299 // *****
300 // Method for output
301
302 public static void outputData(
303     PrintWriter output,
304     double sumRiemb,
305     double sumMile,
306     double avgReimb,
307     double avgMile,
308     double sumValue,
309     int sumPosVal)
310 {
311     String str2;
312     str2 = "\r\n" + "Total amount of reimburesment: " +
313         tools.leftPad(sumRiemb, 10, "$#,##0.00") +
314         "\r\n" + "Total amount of mileage: " +
315         tools.leftPad(sumMile, 15, "##,##0.0") +
316         "\r\n" + "Average of reimburesment: " +
317         tools.leftPad(avgReimb, 15, "$##0.00") +
318         "\r\n" + "Average of mileage driven: " +
319         tools.leftPad(avgMile, 13, "##0.0") +
320         "\r\n" + "Total values processed: " +
321         tools.leftPad(sumValue, 14, "##0") +
322         "\r\n" + "Total positive (mi) values: " +
323         tools.leftPad(sumPosVal, 10, "##0");
324
325     System.out.print(str2);
326     output.println(str2);
327
328 } // End outputData
329 } // End Class

```