```
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 then writes them into a new file while calculating the
7     reimbursement amounts.
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
11    Zachary Stall
12    Program #6, CS 1050, Section 2
13    jGRASP, Custom PC, Windows 10
14
15    Attenuate - to make thin or slender.
16
17    "The best way to predict the future is to create it."
18    -Peter Drucker (1909 - 2005)
19 */
20
21 import java.util.Scanner;  // For console input
22 import java.io.*;          // Access PrintWriter and related classes
23
24
25 public class ZacharyStall_2_06 {
26
27    static Toolkit tools   = new Toolkit();
28    static Scanner console = new Scanner(System.in);
29
30    public static void main (String [] args) throws IOException {
31
32       // Access the input/output method
33       final String INPUT_FILE  = "ZacharyStall_2_06_Input.txt";
34       final String OUTPUT_FILE = "ZacharyStall_2_06_Output.txt";
35
36       int totalPosVal      = 0;    // Number of positive mileages
37       double dataValue     = 0;    // Number of data values
38       double mileDriven    = 0.0;  // Mileage driven
39       double base          = 0.0;  // Base amount for reimbursement
40       double rate          = 0.0;  // Rate per mile to reimburse
41       double overage       = 0.0;  // Mileage overage which to calc rate per mile
42       double reimbMoney    = 0.0;  // Any money owed over base rates
43
44       double totalMileage  = 0.0;  // Sum of all the miles
45       double totalReimb    = 0.0;  // Sum of all reimbersement
46
47       String mileReimbStr;
48
49       // Access the input/output files
50       File inputDataFile = new File(INPUT_FILE);
51       Scanner inputFile  = new Scanner(inputDataFile);
52
53       FileWriter outputDataFile = new FileWriter(OUTPUT_FILE);
54       PrintWriter outputFile = new PrintWriter(outputDataFile);
55
56       // Begin program execution
57       System.out.println("Reading  file " + INPUT_FILE  + "\r\n" +
58                          "Creating file " + OUTPUT_FILE + "\r\n");
59
60       // Prints the headers for the table
61       displayHeader(outputFile);
62
63       // dataValue is the first number in the input file, number of data values
64       dataValue = inputFile.nextDouble();
65
66       // While loop to go through input file and create data table
67       while(inputFile.hasNext()) {
68
```

```
 69          // Gets the next value in the input file and stores it
 70          mileDriven = inputFile.nextDouble();
 71
 72          // Checks to see if mileage is less than zero...
 73          // if it is, will print five stars for reimbursement
 74          if(mileDriven <= 0) {
 75             System.out.println(tools.leftPad(mileDriven, 10, "##,##0.0") +
 76                                tools.padString("*****", 20, " ", ""));
 77             outputFile.println(tools.leftPad(mileDriven, 10, "##,##0.0") +
 78                                tools.padString("*****", 20, " ", ""));
 79             continue;
 80          }
 81
 82          // Checks for miles and assigns appropriate values for base, rate, and overage
 83          else if (mileDriven <  400) {base =   0; rate = 0.18; overage = mileDriven;}
 84          else if (mileDriven <  900) {base =  65; rate = 0.15; overage = mileDriven - 400;}
 85          else if (mileDriven < 1300) {base = 115; rate = 0.12; overage = mileDriven - 900;}
 86          else if (mileDriven < 1900) {base = 140; rate = 0.10; overage = mileDriven - 1300;
}
 87          else if (mileDriven < 2600) {base = 165; rate = 0.08; overage = mileDriven - 1900;
}
 88          else
 89             {base = 195; rate = 0.06; overage = mileDriven - 2600;}
 90
 91          // Calculate and output the reimbursement amount and calculate running totals
 92          reimbMoney = base + (rate * overage);
 93          totalMileage += mileDriven;
 94          totalReimb += reimbMoney;
 95          totalPosVal++;
 96
 97          // Output the table of data to the counsole and the output file
 98          mileReimbStr =
 99             tools.leftPad(mileDriven, 10, "##,##0.0") +
100             tools.leftPad(reimbMoney, 20, "$#,##0.00");
101
102          System.out.println(mileReimbStr);
103          outputFile.println(mileReimbStr);
104       } // End while loop
105
106       // Using methods to output formatted data to the console and output file
107       outputData(outputFile, totalReimb, totalMileage, dataValue, totalPosVal);
108
109       inputFile.close();
110       outputFile.close();
111
112       System.exit(0);
113    } // End Main
114
115    // *********************************************************************
116
117    // Method for headers
118    public static void displayHeader(PrintWriter output) {
119       String str;
120       str = tools.padString("Mileage", 10, " ", "") +
121                        tools.padString("Reimbursement", 20, " ", "") +
122                        "\r\n" +
123                        tools.padString("-------", 10, " ", "") +
124                        tools.padString("-------------", 20, " ", "") +
125                        "\r\n";
126
127      System.out.print(str);
128      output.println(str);
129    } // End headers
130
131   // *********************************************************************
132
133    // Method for output
134    public static void outputData(
```

```
135                                 PrintWriter output,
136                                 double sumRiemb,
137                                 double sumMile,
138                                 double sumValue,
139                                 int sumPosVal)
140       {
141       String str2;
142       str2 =  "\r\n" + "Total amount of reimburesment: " +
143                tools.leftPad(sumRiemb, 10, "$#,##0.00") +
144                "\r\n" + "Total amount of mileage: " +
145                tools.leftPad(sumMile, 15, "##,##0.0") +
146                "\r\n" + "Total values processed: " +
147                tools.leftPad(sumValue, 14, "##0") +
148                "\r\n" + "Total positive (mi) values: " +
149                tools.leftPad(sumPosVal, 10, "##0");
150
151       System.out.print(str2);
152       output.println(str2);
153
154       } // End outputData
155 } // End Class
```