

```

1 package misc;
2
3 import java.io.File;
4 import java.io.FileNotFoundException;
5 import java.util.HashMap;
6 import java.util.Scanner;
7 import java.util.Set;
8
9 public class RecordProcessor {
10     private static String [] firstNames;
11     private static String [] lastNames;
12     private static int [] employeeAge;
13     private static String [] employeeType;
14     private static double [] employeePay;
15
16     public static String processFile(String f) {
17         StringBuffer output = new StringBuffer();
18
19         Scanner fileIn = null;
20         try {
21             fileIn = new Scanner(new File(f));
22         } catch (FileNotFoundException e) {
23             System.err.println(e.getMessage());
24             return null;
25         }
26
27         int numOfRecords = 0;
28         while(fileIn.hasNextLine()) {
29             String l = fileIn.nextLine();
30             if(l.length() > 0)
31                 numOfRecords++;
32         }
33
34         firstNames = new String[numOfRecords];
35         lastNames = new String[numOfRecords];
36         employeeAge = new int[numOfRecords];
37         employeeType = new String[numOfRecords];
38         employeePay = new double[numOfRecords];
39
40         fileIn.close();
41         try {
42             fileIn = new Scanner(new File(f));
43         } catch (FileNotFoundException e) {
44             System.err.println(e.getMessage());
45             return null;
46         }
47
48         numOfRecords = 0;
49         while(fileIn.hasNextLine()) {
50             String fileLine = fileIn.nextLine();
51             if(fileLine.length() > 0) {
52
53                 String [] fileLineData = fileLine.split(",");
54
55                 int loopCount = 0;
56                 for(;loopCount < lastNames.length; loopCount++) {
57                     if(lastNames[loopCount] == null)
58                         break;
59
60                     if(lastNames[loopCount].compareTo(fileLineData[1]) > 0) {
61                         for(int i = numOfRecords; i > loopCount; i--) {
62                             firstNames[i] = firstNames[i - 1];
63                             lastNames[i] = lastNames[i - 1];
64                             employeeAge[i] = employeeAge[i - 1];
65                             employeeType[i] = employeeType[i - 1];
66                             employeePay[i] = employeePay[i - 1];
67                         }
68                         break;
69                     }
70                 }
71
72                 firstNames[loopCount] = fileLineData[0];
73                 lastNames[loopCount] = fileLineData[1];
74                 employeeType[loopCount] = fileLineData[3];
75
76                 try {
77                     employeeAge[loopCount] = Integer.parseInt(fileLineData[2]);
78                     employeePay[loopCount] = Double.parseDouble(fileLineData[4]);
79                 } catch (Exception e) {
80                     System.err.println(e.getMessage());
81                     fileIn.close();
82                     return null;
83                 }
84
85                 numOfRecords++;
86             }
87         }
88
89         if(numOfRecords == 0) {
90             System.err.println("No records found in data file");

```

```

91         fileIn.close();
92         return null;
93     }
94
95     output.append(String.format("# of people imported: %d\n", firstNames.length));
96     output.append(String.format("\n%-30s %s  %-12s %12s\n", "Person Name", "Age"
97         , "Emp. Type", "Pay"));
98     for(int i = 0; i < 30; i++) output.append(String.format("-"));
99     output.append(String.format(" --- "));
100    for(int i = 0; i < 12; i++) output.append(String.format("-"));
101    output.append(String.format(" "));
102    for(int i = 0; i < 12; i++) output.append(String.format("-"));
103    output.append(String.format("\n"));
104
105    for(int i = 0; i < firstNames.length; i++) {
106        output.append(String.format("%-30s %-3d  %-12s $%12.2f\n", firstNames[i] + " "
107            + lastNames[i], employeeAge[i], employeeType[i], employeePay[i]));
108    }
109
110    int sumOfAges = 0;
111    float averageAge = 0f;
112    int numComissionEmployee = 0, numHourlyEmployee = 0, numSalaryEmployee = 0;
113    double sumOfCommissionPay = 0, sumOfHourlyPay = 0, sumOfSalaryPay = 0;
114    double averageCommission = 0, averageHourly = 0, averageSalary = 0;
115    for(int i = 0; i < firstNames.length; i++) {
116        sumOfAges += employeeAge[i];
117        if(employeeType[i].equals("Commission")) {
118            sumOfCommissionPay += employeePay[i];
119            numComissionEmployee++;
120        } else if(employeeType[i].equals("Hourly")) {
121            sumOfHourlyPay += employeePay[i];
122            numHourlyEmployee++;
123        } else if(employeeType[i].equals("Salary")) {
124            sumOfSalaryPay += employeePay[i];
125            numSalaryEmployee++;
126        }
127    }
128    averageAge = (float) sumOfAges / firstNames.length;
129    output.append(String.format("\nAverage age:          %12.1f\n", averageAge));
130    averageCommission = sumOfCommissionPay / numComissionEmployee;
131    output.append(String.format("Average commission:  $%12.2f\n", averageCommission));
132    averageHourly = sumOfHourlyPay / numHourlyEmployee;
133    output.append(String.format("Average hourly wage:  $%12.2f\n", averageHourly));
134    averageSalary = sumOfSalaryPay / numSalaryEmployee;
135    output.append(String.format("Average salary:      $%12.2f\n", averageSalary));
136
137    HashMap<String, Integer> firstNameMap = new HashMap<String, Integer>();
138    int numOfDuplicateFirstNames = 0;
139    for(int i = 0; i < firstNames.length; i++) {
140        if(firstNameMap.containsKey(firstNames[i])) {
141            firstNameMap.put(firstNames[i], firstNameMap.get(firstNames[i]) + 1);
142            numOfDuplicateFirstNames++;
143        } else {
144            firstNameMap.put(firstNames[i], 1);
145        }
146    }
147
148    output.append(String.format("\nFirst names with more than one person sharing it:\n"));
149    if(numOfDuplicateFirstNames > 0) {
150        Set<String> set = firstNameMap.keySet();
151        for(String name : set) {
152            if(firstNameMap.get(name) > 1) {
153                output.append(String.format("%s, # people with this name: %d\n", name
154                    , firstNameMap.get(name)));
155            }
156        }
157    } else {
158        output.append(String.format("All first names are unique"));
159    }
160
161    HashMap<String, Integer> lastNameMap = new HashMap<String, Integer>();
162    int numOfDuplicateLastNames = 0;
163    for(int i = 0; i < lastNames.length; i++) {
164        if(lastNameMap.containsKey(lastNames[i])) {
165            lastNameMap.put(lastNames[i], lastNameMap.get(lastNames[i]) + 1);
166            numOfDuplicateLastNames++;
167        } else {
168            lastNameMap.put(lastNames[i], 1);
169        }
170    }
171
172    output.append(String.format("\nLast names with more than one person sharing it:\n"));
173    if(numOfDuplicateLastNames > 0) {
174        Set<String> set = lastNameMap.keySet();
175        for(String name : set) {
176            if(lastNameMap.get(name) > 1) {
177                output.append(String.format("%s, # people with this name: %d\n", name
178                    , lastNameMap.get(name)));
179            }
180        }

```

```
181         } else {
182             output.append(String.format("All last names are unique"));
183         }
184
185         //close the file
186         fileIn.close();
187
188         return output.toString();
189     }
190
191 }
```