```java
package src;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.Arrays;
import java.util.InputMismatchException;
import java.util.Scanner;

public class ProjectMain {
    static int floor;
    public static People[] queue = new People[6];

    public static void main(String[] args) throws IOException {
        // create an array and fill it with employee objects
        People[] employees = new People[50];
        // read in all the employees from file
        try {
            File fileObj = new File("C:\\Users\\cbcha\\Downloads\\employees.txt");
            Scanner scanObj = new Scanner(fileObj);
            for (int i = 0; i < 50; i++) {
                if (scanObj.hasNextLine()) {
                    String data = scanObj.nextLine();
                    int idTest;
                    int depTest;
                    String nameTest;

                    String idString = data.substring(0, 3);
                    String depString = data.substring(data.length() - 1, data.length());

                    idTest = Integer.parseInt(idString);
                    depTest = Integer.parseInt(depString);
                    nameTest = data.substring(3, data.length() - 1);
```

```
34
35                      People entry = new People(idTest
   , depTest, nameTest);
36                      employees[i] = entry;
37                  }
38              }
39          } catch (FileNotFoundException e) {
40              System.out.println("The file you entered
   in incorrect, please enter a new file:");
41              Scanner keyboard = new Scanner(System.in
   );
42              String newFile = keyboard.nextLine();
43              File fileObj = new File(newFile);
44              Scanner scanObj = new Scanner(fileObj);
45              for (int i = 0; i < 50; i++) {
46                  if (scanObj.hasNextLine()) {
47                      String data = scanObj.nextLine();
48                      int idTest;
49                      int depTest;
50                      String nameTest;
51
52                      String idString = data.substring(
   0, 3);
53                      String depString = data.substring
   (data.length() - 1, data.length());
54
55                      idTest = Integer.parseInt(
   idString);
56                      depTest = Integer.parseInt(
   depString);
57                      nameTest = data.substring(3, data
   .length() - 1);
58
59                      People entry = new People(idTest
   , depTest, nameTest);
60                      employees[i] = entry;
61                  }
62              }
63          }
64          // get which employee is waiting next in the
   queue and fill up the queue with 6 employees
```

```
65          Scanner keyboard = new Scanner(System.in);
66          int counter = 0;
67          while (counter < 6) {
68              try {
69                  System.out.println("Please scan your
     ID number: ");
70                  int scan = keyboard.nextInt();
71                  for (int i = 0; i < 50; i++) {
72                      if (scan == employees[i].
    getIDnum()) {
73                          System.out.println("This
    VERIFIED employee is going to floor " + employees[i
    ].getFloorNum());
74                          floor = employees[i].
    getFloorNum();
75                          // add this employee to the
    queue
76                          queue[counter] = new People(
    employees[i].getIDnum(), employees[i].getFloorNum
    (), employees[i].geteName());
77                          counter++;
78                      }
79                  }
80              } catch (InputMismatchException e) {
81                  System.out.println("Input error,
    please input an integer");
82                  System.out.println("Please scan your
     ID number: ");
83                  int scan = keyboard.nextInt();
84                  for (int i = 0; i < 50; i++) {
85                      if (scan == employees[i].
    getIDnum()) {
86                          System.out.println("This
    VERIFIED employee is going to floor " + employees[i
    ].getFloorNum());
87                          floor = employees[i].
    getFloorNum();
88                          // add this employee to the
    queue
89  //                      queue.add(new People(employees
    [i].getIDnum(), employees[i].getFloorNum(),
```

```
 89 employees[i].geteName()));
 90                          queue[counter] = new People(
    employees[i].getIDnum(), employees[i].getFloorNum
    (), employees[i].geteName());
 91                          counter++;
 92                      }
 93                  }
 94              }
 95              System.out.println("The elevator is now
    fill and will bring all 6 passengers to their floor
    .");
 96          }
 97          // now there are 6 employees ready to be
    brought to their floor
 98          int one = queue[0].getFloorNum();
 99          int two = queue[1].getFloorNum();
100          int three = queue[2].getFloorNum();
101          int four = queue[3].getFloorNum();
102          int five = queue[4].getFloorNum();
103          int six = queue[5].getFloorNum();
104          int[] numbers = {one, two, three, four, five
    , six};
105          // sort the floors in ascending order and
    drop off each employee
106          Arrays.sort(numbers);
107          for (int i = 0; i < 6; i++) {
108              Elevator ride = new Elevator(numbers[i
    ]);
109              System.out.println(ride);
110              counter--;
111              if (counter == 0) {
112                  System.out.println("All the
    employees have been dropped off, the ride was
    sucessful");
113              }
114          }
115      }
116
117
118 }
119
```

```
120
121
122
```