

Untitled-2

```
1 import java.io.*;
2 import java.util.*;
3
4 public class PayrollSystem {
5     private static final String FILE_NAME = "employees.txt";
6     private static Map<String, Employee> employees = new HashMap<>();
7
8     public static void main(String[] args) {
9         loadEmployees();
10        Scanner scanner = new Scanner(System.in);
11        String command;
12
13        while (true) {
14            System.out.println("\nEmployee Payroll System");
15            System.out.println("1. Add Employee");
16            System.out.println("2. Calculate Salary");
17            System.out.println("3. View Employee");
18            System.out.println("4. Exit");
19            System.out.print("Enter command: ");
20            command = scanner.nextLine();
21
22            switch (command) {
23                case "1":
24                    addEmployee(scanner);
25                    break;
26                case "2":
27                    calculateSalary(scanner);
28                    break;
29                case "3":
30                    viewEmployee(scanner);
31                    break;
32                case "4":
33                    saveEmployees();
34                    System.out.println("Exiting...");
35                    scanner.close();
36                    return;
37                default:
38                    System.out.println("Invalid command. Please try again.");
39            }
40        }
41    }
42
43    private static void addEmployee(Scanner scanner) {
44        System.out.print("Enter employee ID: ");
45        String id = scanner.nextLine();
46        if (employees.containsKey(id)) {
47            System.out.println("Employee ID already exists.");
48            return;
49        }
50    }
51}
```

```
49     }
50
51     System.out.print("Enter employee name: ");
52     String name = scanner.nextLine();
53     System.out.print("Enter employee hourly wage: ");
54     double hourlyWage = Double.parseDouble(scanner.nextLine());
55     System.out.print("Enter hours worked: ");
56     double hoursWorked = Double.parseDouble(scanner.nextLine());
57
58     Employee employee = new Employee(id, name, hourlyWage, hoursWorked);
59     employees.put(id, employee);
60     System.out.println("Employee added successfully.");
61 }
62
63 private static void calculateSalary(Scanner scanner) {
64     System.out.print("Enter employee ID to calculate salary: ");
65     String id = scanner.nextLine();
66     Employee employee = employees.get(id);
67
68     if (employee != null) {
69         double salary = employee.calculateSalary();
70         System.out.println("Salary for " + employee.getName() + ":
VSCODE_PRINT_CONTENTquot; + salary);
71     } else {
72         System.out.println("Employee not found.");
73     }
74 }
75
76 private static void viewEmployee(Scanner scanner) {
77     System.out.print("Enter employee ID to view details: ");
78     String id = scanner.nextLine();
79     Employee employee = employees.get(id);
80
81     if (employee != null) {
82         System.out.println("Employee ID: " + employee.getId());
83         System.out.println("Name: " + employee.getName());
84         System.out.println("Hourly Wage: VSCODE_PRINT_CONTENTquot; +
employee.getHourlyWage());
85         System.out.println("Hours Worked: " + employee.getHoursWorked());
86     } else {
87         System.out.println("Employee not found.");
88     }
89 }
90
91 private static void loadEmployees() {
92     try (BufferedReader reader = new BufferedReader(new FileReader(FILE_NAME))) {
93         String line;
94         while ((line = reader.readLine()) != null) {
95             String[] parts = line.split(",");
96             if (parts.length == 4) {
```

```
97         String id = parts[0];
98         String name = parts[1];
99         double hourlyWage = Double.parseDouble(parts[2]);
100        double hoursWorked = Double.parseDouble(parts[3]);
101
102        Employee employee = new Employee(id, name, hourlyWage, hoursWorked);
103        employees.put(id, employee);
104    }
105    }
106    } catch (IOException e) {
107        System.out.println("No existing employee file found. Starting fresh.");
108    }
109    }
110
111    private static void saveEmployees() {
112        try (BufferedWriter writer = new BufferedWriter(new FileWriter(FILE_NAME))) {
113            for (Employee employee : employees.values()) {
114                writer.write(employee.getId() + "," + employee.getName() + ","
115                    + employee.getHourlyWage() + "," + employee.getHoursWorked());
116                writer.newLine();
117            }
118        } catch (IOException e) {
119            System.out.println("Error saving employee data.");
120        }
121    }
122
123    private static class Employee {
124        private String id;
125        private String name;
126        private double hourlyWage;
127        private double hoursWorked;
128
129        public Employee(String id, String name, double hourlyWage, double hoursWorked) {
130            this.id = id;
131            this.name = name;
132            this.hourlyWage = hourlyWage;
133            this.hoursWorked = hoursWorked;
134        }
135
136        public String getId() {
137            return id;
138        }
139
140        public String getName() {
141            return name;
142        }
143
144        public double getHourlyWage() {
145            return hourlyWage;
146        }
```

```
147
148     public double getHoursWorked() {
149         return hoursWorked;
150     }
151
152     public double calculateSalary() {
153         return hourlyWage * hoursWorked;
154     }
155 }
156 }
157
```