

```
1 package Lab08;
2
3 import java.util.Arrays;
4
5
6 public class Employee {
7
8     private int employeeID;
9     private double pay, allowance, netSalary, incomeTax;
10    private String name;
11
12    public Employee(Employee e[], int i) {
13        readEmployeeInfo(e, i);
14        calAllowance();
15        calTax();
16        calSalary();
17        System.out.println(this);
18    }
19
20    static int getInput(String msg) {
21        Scanner s = new Scanner(System.in);
22        System.out.println(msg);
23        return s.nextInt();
24    }
25
26    static void search(Employee e[], int id) {
27        for (Employee emp : e) {
28            if (emp.employeeID == id) {
29                displayAll(new Employee[] {emp});
30                return;
31            }
32        }
33        System.out.println("No Employee found.");
34    }
35
36    void readEmployeeInfo(Employee[] e, int i) {
37        Scanner s = new Scanner(System.in);
38        do {
39            System.out.println("Enter ID: ");
40            employeeID = s.nextInt();
41        } while (validateID(e, i, employeeID));
42        s.nextLine();
43        System.out.println("Enter Name: ");
44        name = s.nextLine();
45        System.out.println("Enter Basic Pay: ");
46        pay = s.nextDouble();
47    }
48
49    boolean validateID(Employee[] e, int i, int id) {
50        for (Employee emp : e) {
51            if (emp == null)
52                continue;
53            if (emp.employeeID == id)
54                return true;
55        }
56        //
57        for (int j = 0; j < i; j++) {
58            if (this.employeeID == e[j].employeeID) {
```

```
59 //             System.out.println("Invalid id.");
60 //             return true;
61 //         }
62 //     }
63     return false;
64 }
65
66 void calAllowance() {
67     this.allowance = .5 * this.pay;
68 }
69
70 void calTax() {
71     double gross = pay + allowance;
72     if (gross <= 5000)
73         incomeTax = 0;
74     else if (gross > 5000 && gross <= 6000)
75         incomeTax = .1 * pay;
76     else if (gross > 6000 && gross <= 10000)
77         incomeTax = .15 * pay;
78     else
79         incomeTax = .2 * pay;
80 }
81
82 void calSalary() {
83     netSalary = pay + allowance - incomeTax;
84 }
85
86 public int getEmployeeID() {
87     return employeeID;
88 }
89
90 public void setEmployeeID(int employeeID) {
91     this.employeeID = employeeID;
92 }
93
94 public double getPay() {
95     return pay;
96 }
97
98 public void setPay(double pay) {
99     this.pay = pay;
100 }
101
102 public double getAllowance() {
103     return allowance;
104 }
105
106 public void setAllowance(double allowance) {
107     this.allowance = allowance;
108 }
109
110 public double getNetSalary() {
111     return netSalary;
112 }
113
114 public void setNetSalary(double netSalary) {
115     this.netSalary = netSalary;
```

```
116     }
117
118     public double getIncomeTax() {
119         return incomeTax;
120     }
121
122     public void setIncomeTax(double incomeTax) {
123         this.incomeTax = incomeTax;
124     }
125
126     public String getName() {
127         return name;
128     }
129
130     public void setName(String name) {
131         this.name = name;
132     }
133
134     @Override
135     public String toString() {
136         return "Employee [employeeID=" + employeeID + ", pay=" + pay + ", allowance=" +
allowance + ", netSalary="
137             + netSalary + ", incomeTax=" + incomeTax + ", name=" + name + "]\n";
138     }
139
140     public static void displayAll(Employee[] e) {
141
142         System.out.println
143         ("|-----|");
144         System.out.printf("%-10s%-20s%-15s%-15s%-15s%-15s|\n", "EMP ID", "Name", "Basic Pay",
"Allowance", "Income Tax",
145             "Net Salary");
146         System.out.println
147         ("|-----|");
148
149         for(Employee emp : e) {
150             System.out.printf("%-10d%-20s%-15.2f%-15.2f%-15.2f%-15.2f|\n", emp.employeeID,
emp.name, emp.pay, emp.allowance, emp.incomeTax,
151                 emp.netSalary);
152         }
153         System.out.println
154         ("|-----|");
155     }
156
157     public static void sortBasedOnNames(Employee[] e) {
158         Arrays.sort(e, (e1, e2) -> e1.name.compareTo(e2.name));
159     }
160 }
161 }
162 }
```