

**Q1:**  
**ALL Answers are in attached audio file.**

**Q2, build class diagram and write down java code in order to solve question 2 (50)**

**Create a class named Employee that can be used to calculate the salaries of different employees. The Employee class should keep a track of the employee ID, name, department, salary, and designation with appropriate accessor and mutator methods. Also create an equals() method that overrides Object's equals() method, where employees can check if their designation is identical. Next, create two additional classes named Manager and Clerk that are derived from Employee. Create an overridden method named addBonus that returns the salary of the employee after adding up the bonus. There is a default bonus of Rs.2000/month. Managers have a bonus of Rs.3000/month and clerks have a bonus of Rs.1000/month. Finally create a display method to print the details of the employee. You may assume the initial salary of an employee and other necessary values. Furthermore calculate the salary deductions based on the number of days an employee is on leave. Consider 20 working days per month. Add a method that calculates the deductions of each employee based on their leave record. In your main method, create an array of type deduction filled with sample data of all types of Employees. Finally calculate the total deduction that iterates through the array and returns the total amount of deductions of all the employees in a month.**

**Code**

```
package finalterm;

public class Employee {

    private double emply_ID;
    private String name;
    private String department;
    private double salary;
    private String designation;
    private double bonus=2000;
    public double netsalary;

    public double array_deduction(double days) {
        return (days/20)*netsalary;
    }

    public void deduction(double no_of_days) {
        this.netsalary=netsalary*((20-no_of_days)/20);
    }

    public Employee() { }

    public Employee(String name,double salary) {
        this.name=name;
        this.salary=salary;
    }
    public void setEmp_Id(double id) {
        this.emply_ID = id;
    }
    public void setName(String name) {
        this.name = name;
    }
}
```

```

}
public void setSalary(double salary) {
this.salary = salary;
}
public void setDepartment(String department) {
this.department = department;
}
public void setDesignation(String designation) {

this.designation = designation;

if(designation.equals("Manager")){
Manager obj=new Manager();
bonus=obj.addBonus();
}
else if(designation.equals("Clerk")) {
Clerk obj=new Clerk(); bonus=obj.addBonus();
} }
public double getEmp_Id() {
return emply_ID;
}

public String getName() {
return name;
}

public double getSalary() {

netsalary=this.salary+addBonus();
return netsalary;
}
public String getDepartment() {
return department;
}

public String getDesignation() {
return designation; }

public void display() {
System.out.println("Employee Id:"+getEmp_Id());
System.out.println("Employee Name:"+getName());
System.out.println("Employee Designation:"+getDesignation());
System.out.println("Employee Salary:"+getSalary());
}
@Override
public boolean equals(Object obj) {
Employee emp=(Employee)obj;

if(this.designation.equals(emp.getDesignation())) {
return true; }
else { return false; } }

public double addBonus() {
return bonus;
}

```

```

public static void main(String[] args) {

    double Total_deduction[];
    Total_deduction=new double[10];

    Employee e1=new Employee();

    e1.setEmp_Id(1);
    e1.setName("Abuzar");
    e1.setDepartment("Computer Checking");
    e1.setDesignation("Manager");
    e1.setSalary(5000);
    e1.display();
    e1.deduction(1);
    Total_deduction[0]= e1.array_deduction(1);

    System.out.println("Deducted Salary is = "+e1.netsalary);

    Employee e2=new Employee();
    e2.setEmp_Id(2);
    e2.setName("Umair");
    e2.setDepartment("Software Analyzing");
    e2.setDesignation("Clerk");
    e2.setSalary(4000);
    e2.display();
    e2.deduction(3);
    Total_deduction[1]= e1.array_deduction(3);

    System.out.println("Deducted Salary is = "+e2.netsalary);

    Employee e3=new Employee();
    e3.setEmp_Id(3);
    e3.setName("Rashid");
    e3.setDepartment("EE");
    e3.setDesignation("Employee");
    e3.setSalary(3000);
    e3.display();
    e3.deduction(5);
    Total_deduction[2]= e1.array_deduction(5);

    System.out.println("Deducted Salary is = "+e3.netsalary);
    double sum=0;
    for (double values : Total_deduction) {
        sum=sum+values;
    }

    System.out.println("The sum of array of deduction is = " + sum);

    if(e3.equals(e2)) {
        System.out.println("Same Designation"); }
    else {
        System.out.println("Different Designation"); } } }

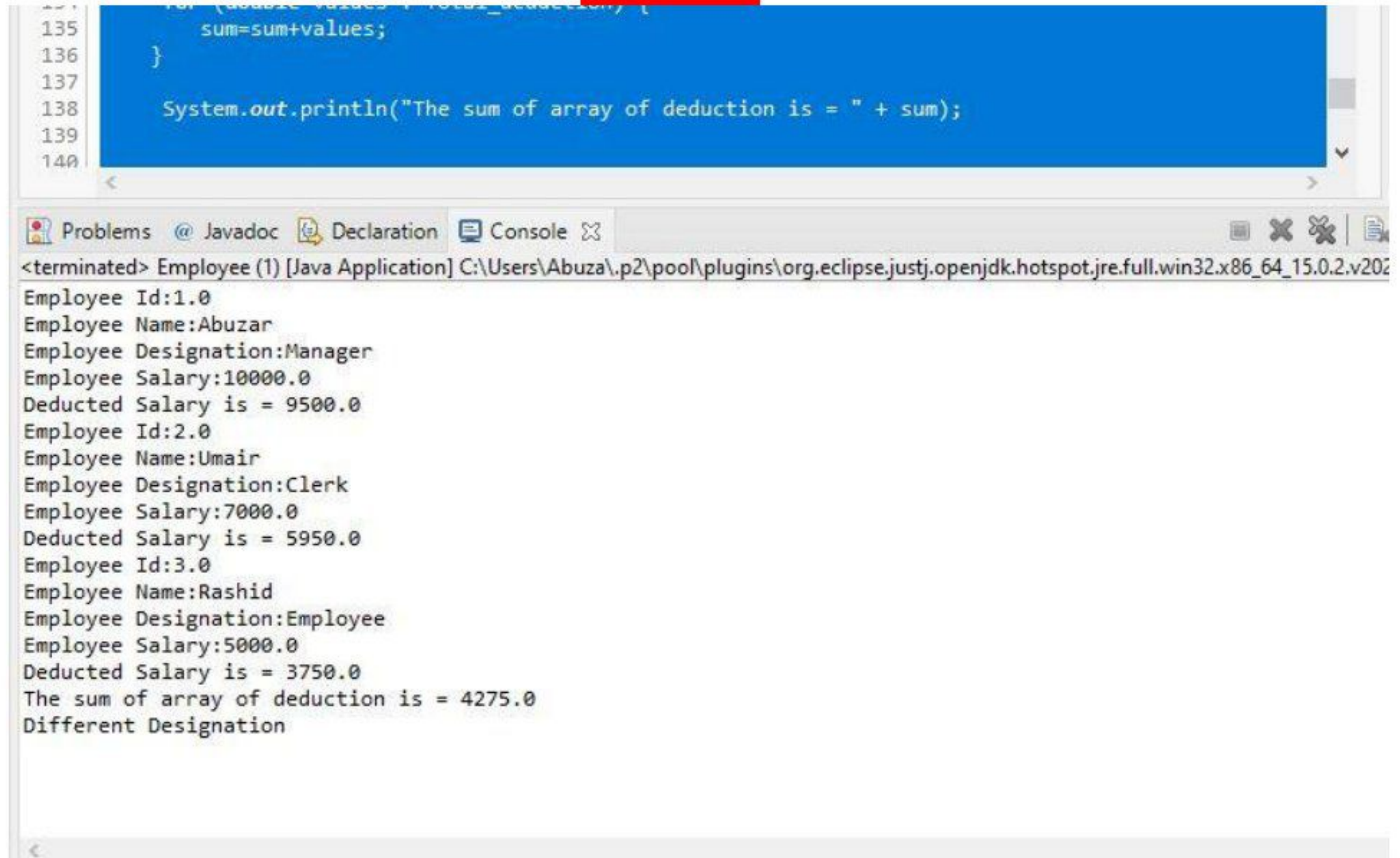
class Manager extends Employee {

```

```
@Override
public double addBonus() {
return (super.addBonus()+3000);
} }
```

```
class Clerk extends Employee { public double addBonus() {
return (super.addBonus()+1000);
}}}
```

## OUTPUT



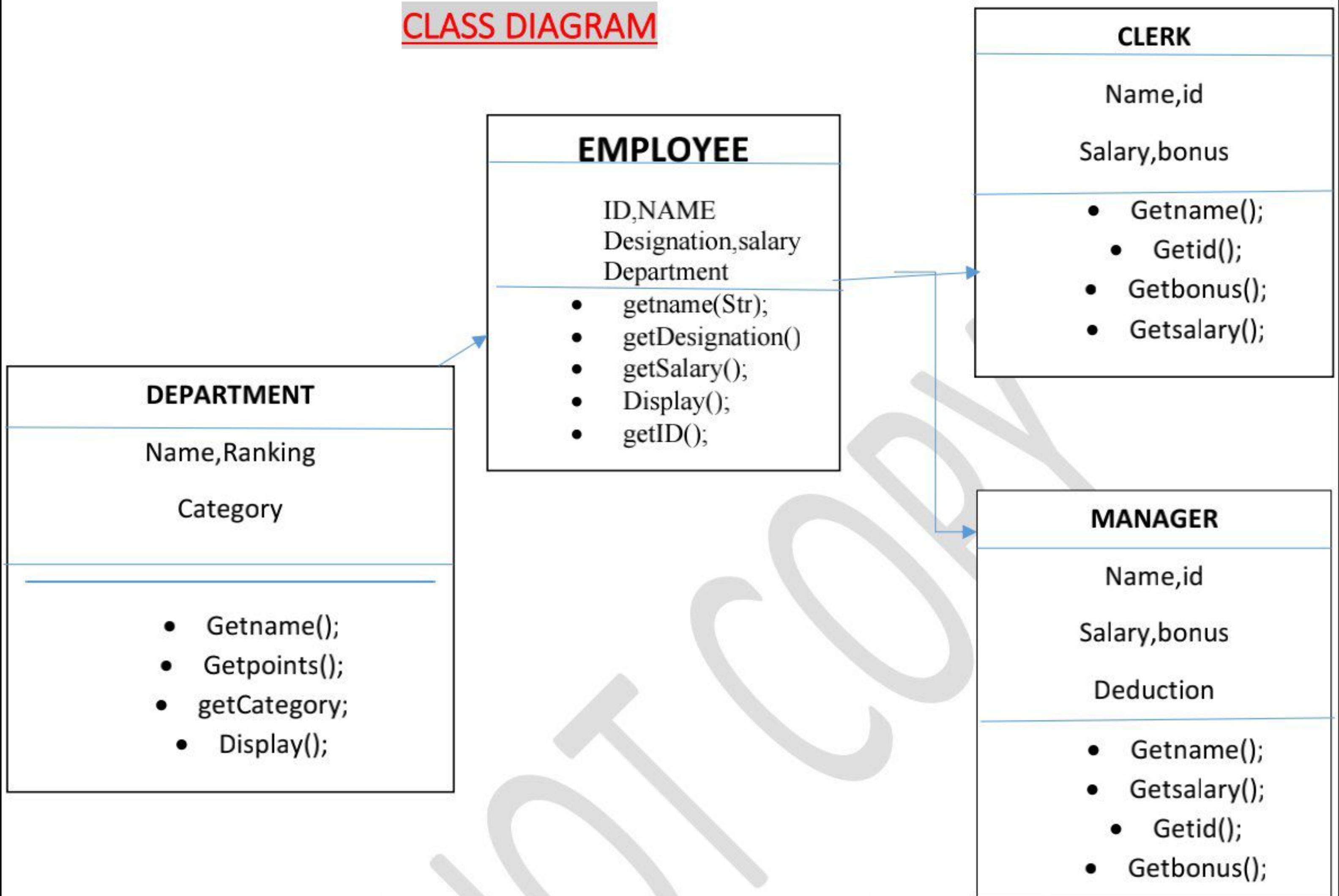
The screenshot shows the Eclipse IDE's console window. The top part of the console displays the following Java code snippet:

```
135     sum=sum+values;
136 }
137
138 System.out.println("The sum of array of deduction is = " + sum);
139
140
```

Below the code, the console output is displayed, showing the execution of a Java application. The output includes the following text:

```
<terminated> Employee (1) [Java Application] C:\Users\Abuza\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.2.v20z
Employee Id:1.0
Employee Name:Abuzar
Employee Designation:Manager
Employee Salary:10000.0
Deducted Salary is = 9500.0
Employee Id:2.0
Employee Name:Umair
Employee Designation:Clerk
Employee Salary:7000.0
Deducted Salary is = 5950.0
Employee Id:3.0
Employee Name:Rashid
Employee Designation:Employee
Employee Salary:5000.0
Deducted Salary is = 3750.0
The sum of array of deduction is = 4275.0
Different Designation
```

## CLASS DIAGRAM



**THE END**