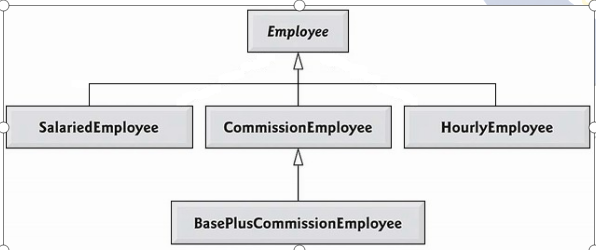
LAB # 08

Task No 01: Create a payroll system using classes, inheritance, and polymorphism. Four types of employees are paid weekly.

1. **Salaried employees:** Fixed salary irrespective of hours.
2. **Hourly employees:** 40 hours salary and overtime (> 40 hours).
3. **Commission employees:** Paid by a percentage of sales.
4. **Base-plus-commission employees:** Base salary and a percentage of sales.

The information about each employee is his/her first name, last name, and national identity card number. The reset depends on the type of employee.



Code:

Main:

package lab08task01;

public class Lab08task01 {

    public static void main(String[] args) {

        Employee e1 = new SalariedEmployee("Abdullah", "Sadiq", "42401-0000000-0", 20000);

        Employee e2 = new CommissionEmployee("Abdul", "Wahab", "4200-0000000-0", 20000, 2.58);

        Employee e3 = new BasePlusCommissionEmployee("Ali", "Nasir", "42030-0000000-0", 15000, 2.3, 80000);

        Employee e4 = new HourlyEmployee("Fahad", "Mustafa", "321-23-2985", 3000, 30);

        System.out.println("First Employee:");

        System.out.println(e1.tostring());

        System.out.println("Salary: " + e1.earnings());

        System.out.println("\nSecond Employee:");

        System.out.println(e2.tostring());

        System.out.println("Salary: " + e2.earnings());

        System.out.println("\nThird Employee:");

        System.out.println(e3.tostring());

        BasePlusCommissionEmployee currentEmployee = (BasePlusCommissionEmployee) e3;

        double oldbasesalary = currentEmployee.getBaseSalary();

        System.out.println("Old Base Salary: " + oldbasesalary);

        currentEmployee.setBaseSalary(1.10 \* oldbasesalary);

        System.out.println("New salary with 10% increase is: " + currentEmployee.getBaseSalary());

        System.out.println("\nFourth Employee: ");

        System.out.println(e4.tostring());

        System.out.println("Salary: " + e4.earnings());

    }

}

Employee (Parent):

package lab08task01;

public class Employee {

    private String fName;

    private String lName;

    private String CNIC;

    public Employee() {

        fName = "Not Available";

        lName = "Not Available";

        CNIC = "0";

    }

    public Employee(String fName, String lName, String CNIC) {

        this.fName = fName;

        this.lName = lName;

        this.CNIC = CNIC;

    }

    public String getfName() {

        return fName;

    }

    public void setfName(String fName) {

        this.fName = fName;

    }

    public String getlName() {

        return lName;

    }

    public void setlName(String lName) {

        this.lName = lName;

    }

    public String getCNIC() {

        return CNIC;

    }

    public void setCNIC(String CNIC) {

        this.CNIC = CNIC;

    }

    public String tostring() {

        return fName + " " + lName + " CNIC# " + CNIC;

    }

    public double earnings() {

        return 0.00;

    }

}

SalariedEmployee (Child):

package lab08task01;

public class SalariedEmployee extends Employee {

    private double weeklysalary;

    public SalariedEmployee() {

        super();

        this.weeklysalary = 0;

    }

    public SalariedEmployee(String fName, String lName, String CNIC, double weeklysalary) {

        super(fName, lName, CNIC);

        this.weeklysalary = weeklysalary;

    }

    public double getWeeklysalary() {

        return weeklysalary;

    }

    public void setWeeklysalary(double weeklysalary) {

        if (weeklysalary < 0) {

            System.out.println("Weekly Salary cannot be negative");

        } else {

            this.weeklysalary = weeklysalary;

        }

    }

    @Override

    public String tostring() {

        return "Salaried Employee: " + super.tostring();

    }

    @Override

    public double earnings() {

        return weeklysalary;

    }

}

CommissionEmployee (Child):

package lab08task01;

public class CommissionEmployee extends Employee {

    private double grossSales;

    private double commissionRate;

    public CommissionEmployee() {

        super();

        this.commissionRate = 0;

        this.grossSales = 0;

    }

    public CommissionEmployee(String fName, String lName, String CNIC, double grossSales,

            double commissionRate) {

        super(fName, lName, CNIC);

        this.commissionRate = commissionRate;

        this.grossSales = grossSales;

    }

    public double getGrossSales() {

        return grossSales;

    }

    public void setGrossSales(double grossSales) {

        if (grossSales < 0) {

            System.out.println("Gross Sales cannot be negative");

        } else {

            this.grossSales = grossSales;

        }

    }

    public double getCommissionRate() {

        return commissionRate;

    }

    public void setCommissionRate(double commissionRate) {

        if (commissionRate < 0) {

            System.out.println("Commision Rate cannot be negative");

        } else {

            this.commissionRate = commissionRate;

        }

    }

    @Override

    public String tostring() {

        return "Commission Employee: " + super.tostring();

    }

    @Override

    public double earnings() {

        return grossSales \* commissionRate;

    }

}

BasePlusCommissionEmployee (Child):

package lab08task01;

public class BasePlusCommissionEmployee extends CommissionEmployee {

    private double baseSalary;

    public BasePlusCommissionEmployee() {

        super();

        this.baseSalary = 0;

    }

    public BasePlusCommissionEmployee(String fName, String lName, String CNIC, double grossSales, double commissionRate, double baseSalary) {

        super(fName, lName, CNIC, grossSales, commissionRate);

        this.baseSalary = baseSalary;

    }

    public double getBaseSalary() {

        return baseSalary;

    }

    public void setBaseSalary(double baseSalary) {

        if (baseSalary < 0) {

            System.out.println("Base Salary cannot be negative");

        } else {

            this.baseSalary = baseSalary;

        }

    }

    @Override

    public String tostring() {

        return "Base plus " + super.tostring();

    }

    @Override

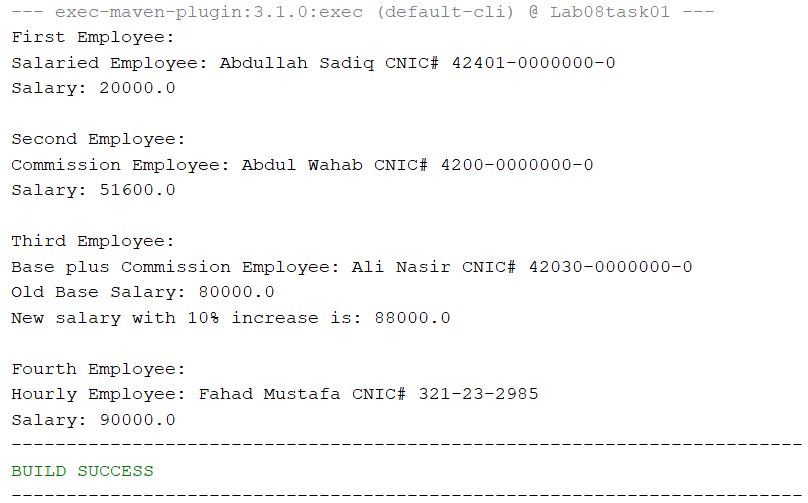
    public double earnings() {

        return baseSalary \* super.earnings();

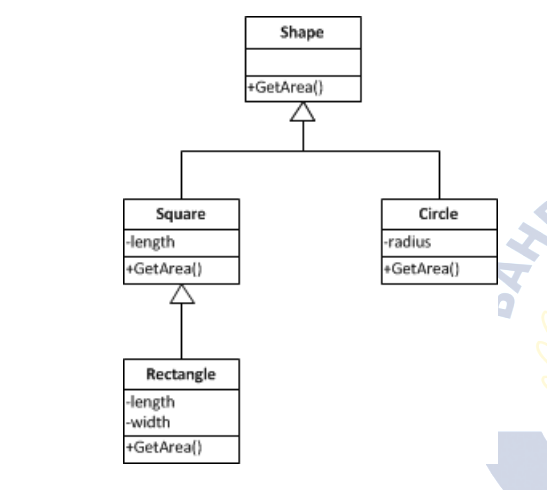
    }

}

Output:



Task No 02: You must implement the following diagram including some attributes and other functions:



Code:

Main:

package lab08task02;

public class Lab08task02 {

    public static void main(String[] args) {

        Shape s1 = new Square(15);

        s1.getArea();

        Shape s2 = new Rectangle(35, 15);

        s2.getArea();

        Shape s3 = new Circle(23.5);

        s3.getArea();

    }

}

Shape (Parent):

package lab08task02;

public class Shape {

    public void getArea() {

        System.out.println("Hello Shape!");

    }

}

Circle (Child):

package lab08task02;

public class Circle extends Shape {

    private double radius, pi = 3.142;

    Circle() {

        radius = 0;

    }

    Circle(double radius) {

        this.radius = radius;

    }

    public void getArea() {

        System.out.println("\nArea of the circle is:" + pi \* radius \* radius + " meters square");

    }

}

Square (Child):

package lab08task02;

public class Square extends Shape {

    private double length;

    public Square() {

        length = 0;

    }

    public Square(double length) {

        this.length = length;

    }

    public double getLength() {

        return length;

    }

    public void setLength(double length) {

        this.length = length;

    }

    public void getArea() {

        System.out.println("Area of Square is:" + length \* length + " meter square");

    }

}

Rectangle (Child):

package lab08task02;

public class Rectangle extends Square{

    private double width;

    Rectangle() {

        super();

        width = 0;

    }

    Rectangle(double length, double width) {

        super(length);

        this.width = width;

    }

    public double getWidth() {

        return width;

    }

    public void setWidth(double width) {

        this.width = width;

    }

    public void getArea() {

        System.out.println("\nArea of Rectangle is:" + getLength() \* getWidth() + " meters square");

    }

}

Output:

