**Object Oriented Programming**

**COMSATS UNIVERSITY ISLAMABAD LAHORE CAMPUS**

****

**Lab Assignment #02**

**Submitted to:**

**Sir- Abubakar Ashraf**

**Submitted by:**

**Nasratullah (FA18-BCS-401)**

**Submission Date:**

**23 Mar, 2022**

**BSE Software Engineering**

**Section-B**

**Inheritance Task:**

**Employee Class :**

|  |
| --- |
| package LabAssignment\_2;  public class Employee {  private String FirstName;  private String LastName;  private String SecurityNumber;   //------------------------------------------------- CONSTRUCTORS ---------------------------------------------------  public Employee(String firstName, String lastName, String ssn) {  FirstName = firstName;  LastName = lastName;  SecurityNumber = ssn;  }   //------------------------------------------------- GETTERS AND SETTERS ---------------------------------------------------  public String getFirstName() {  return FirstName;  }   public void setFirstName(String firstName) {  FirstName = firstName;  }   public String getLastName() {  return LastName;  }   public void setLastName(String lastName) {  LastName = lastName;  }   public String getSecurityNumber() {  return SecurityNumber;  }   public void setSecurityNumber(String ssn) {  SecurityNumber = ssn;  }   @Override  public String toString() {  return String.*format*("%s %s\n social security number: %s",  getFirstName(), getLastName(), getSecurityNumber());  } } |

**Commisioned Employee Class:**

|  |
| --- |
| package LabAssignment\_2;  public class CommisionEmployeee extends Employee{  private double grossSales; // gross weekly sales  private double commissionRate; // commission percentage.  private double bonus;  //------------------------------------------------- CONSTRUCTORS ---------------------------------------------------  public CommisionEmployeee(String firstName, String lastName, String ssn,double sales,double rate) {  super(firstName, lastName, ssn);  setGrossSales( sales );  setCommissionRate( rate );  if (this.commissionRate >= 10000 && this.commissionRate <= 20000){  this.bonus = 5000;  } else if (this.commissionRate >= 20000 && this.commissionRate <= 50000){  this.bonus = 10000;  }else if (this.commissionRate >= 50000){  this.bonus = 20000;  }  }  //------------------------------------------------- GETTERS AND SETTERS ---------------------------------------------------  public double setCommissionRate(double commissionRate )  {  this.commissionRate = commissionRate;  if (this.commissionRate >= 10000 && this.commissionRate <= 20000){  this.bonus = 5000;  } else if (this.commissionRate >= 20000 && this.commissionRate <= 50000){  this.bonus = 10000;  }else if (this.commissionRate >= 50000){  this.bonus = 20000;  }  return this.bonus;   }  public double getcommissionRate(){  return commissionRate;  }  public void setGrossSales(double Sales) {  grossSales = (Sales < 0.0) ? 0.0: Sales;  }   public double getGrossSales() {  return grossSales;  }   public double earning () {  return getcommissionRate() \* getGrossSales();  }   @Override  public String toString()  {  return String.*format*(  "%s: %s\n%s: $%,.2f; %s: %.2f",  "commission employee", super.toString(),  "gross sales", getGrossSales(),  "commission rate", getcommissionRate());  }  } |

**BaseCommision Employee Class:**

|  |
| --- |
| package LabAssignment\_2;  public class BasePlusCommisionEmployee extends CommisionEmployeee{  private double baseSalary;  //------------------------------------------------- CONSTRUCTORS ---------------------------------------------------  public BasePlusCommisionEmployee(String firstName, String lastName, String ssn,double sales, double rate,double salary) {  super(firstName, lastName, ssn,sales,rate);  setBaseSalary (salary);  }  //------------------------------------------------- GETTERS AND SETTERS ---------------------------------------------------   public double getBaseSalary() {  return baseSalary;  }   public void setBaseSalary(double salary) {  baseSalary = ( salary < 0.0 ) ? 0.0 : salary;  }    public double earning () {  return getcommissionRate() \* getGrossSales();  }   public String toString()  {  return String.*format*("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n" +  "\n%s %s; %s: $%,.2f",  "base-salaried", super.toString(),  "base salary", getBaseSalary());  } } |

**Hourly Employee Class:**

|  |
| --- |
| package LabAssignment\_2;  public class HourlyEmployee extends Employee{  private double wage;  private double hours;  //------------------------------------------------- CONSTRUCTORS ---------------------------------------------------   public HourlyEmployee(String firstName, String lastName, String ssn,double hourlyWage, double hoursWorked) {  super(firstName, lastName, ssn);  }   //------------------------------------------------- GETTERS AND SETTERS ---------------------------------------------------  public void setWage(double hourlyWage)  {  wage = (hourlyWage < 0.0) ? 0.0: hourlyWage;  }  public double getWage(){  return wage;  }   public void setHours (double hoursWorked)  {  hours = ((hoursWorked >= 0.0) && (hoursWorked <= 168.0))? hoursWorked: 0.0;  }  public double getHours()  {  return hours;  }   public double earnings ()  {  if (getHours() <= 40)  return getWage() \* getHours();  else  return 40 \* getWage() + (getHours() - 40) \* getWage() \* 1.5;   }   public String toString()  {  return String.*format*("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n hourly employee: %s: \n%s: $%,.2f; %s: %,.2f",  super.toString(),  "hourly wage", getWage(),  "hours worked", getHours());  }  } |

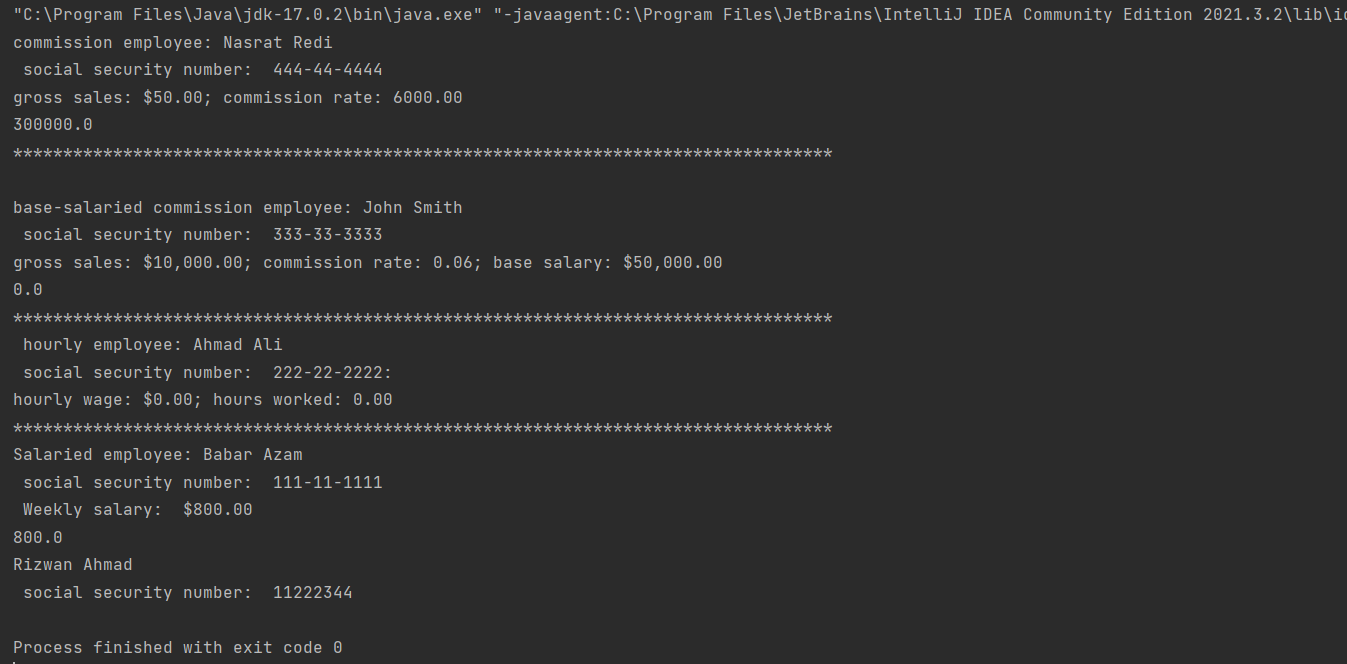
**Salaried Employee Class:**

|  |
| --- |
| package LabAssignment\_2;  public class SalariedEmployee extends Employee{  private double weeklySalary;  //------------------------------------------------- CONSTRUCTORS ---------------------------------------------------  public SalariedEmployee(String firstName, String lastName, String ssn, double salary) {  super(firstName, lastName, ssn);  setWeeklySalary(salary);   }  //------------------------------------------------- GETTERS AND SETTERS ---------------------------------------------------   public void setWeeklySalary(double salary) {  weeklySalary = salary < 0.0? 0.0: salary;  }  public double getWeeklySalary() {  return weeklySalary;  }   public double earnings()  {  return getWeeklySalary();  }   public String toString()  {  return String.*format*("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n" +  "Salaried employee: %s \n %s: $%,.2f " ,  super.toString(),"Weekly salary", getWeeklySalary());  }   } |

**Test Class:**

|  |
| --- |
| package LabAssignment\_2;  public class TestClass {  public static void main(String[] args) {  Employee emp0 =new Employee("Rizwan","Ahmad","11222344");  CommisionEmployeee commsionemp1 = new CommisionEmployeee("Nasrat","Redi","444-44-4444",50,6000);  BasePlusCommisionEmployee basepluscommemp2 = new BasePlusCommisionEmployee("John","Smith","333-33-3333",10000, .06 ,50000);  HourlyEmployee hourlemp3 = new HourlyEmployee("Ahmad", "Ali", "222-22-2222", 16.75, 40 );  SalariedEmployee salaryemp4 = new SalariedEmployee("Babar","Azam","111-11-1111", 800.00 );    System.*out*.println(commsionemp1.toString());  System.*out*.println(commsionemp1.earning());  System.*out*.println(basepluscommemp2.toString());  System.*out*.println(hourlemp3.earnings());  System.*out*.println(hourlemp3.toString());  System.*out*.println(salaryemp4.toString());  System.*out*.println(salaryemp4.earnings());  System.*out*.println(emp0.toString());         //System.out.println(commsionemp.getcommissionRate());        } } |

**Output:**

****

**Assignment Part\_2**

**Shape Class:**

|  |
| --- |
| package Assignment\_2\_Part\_2;  public class Shape {  public enum color {*Red*, *Orange*, *yellow*, *Blue*};  private color shapeColor;  private boolean isFilled;  //------------------------------------------------- CONSTRUCTORS ---------------------------------------------------    public Shape(color shapeColor, boolean isFilled) {  this.shapeColor = shapeColor;  this.isFilled = isFilled;  }   //------------------------------------------------- GETTERS AND SETTERS ---------------------------------------------------   public color getShapeColor() {  return shapeColor;  }   public void setShapeColor(color shapeColor) {  this.shapeColor = shapeColor;  }   public boolean isFilled() {  return isFilled;  }   public double setFilled(boolean filled) {  isFilled = filled;  return 0;  }   public double getArea() {  return 0;  }  public double getParameter(){  return 0;  }  } |

**Circle Class:**

|  |
| --- |
| package Assignment\_2\_Part\_2;  public class Circle extends Shape{  //--------------------instance variable--------------  private double radius;  //------------------------------------------------- CONSTRUCTORS ---------------------------------------------------  public Circle(color shapeColor, boolean isFilled,double radius) {  super(shapeColor, isFilled);  this.radius = radius;   }   //------------------------------------------------- GETTERS AND SETTERS ---------------------------------------------------\   public double getRadius() {  return radius;  }   public void setRadius(double radius) {  this.radius = radius;  }   @Override  public double getArea() {  return Math.*PI* \* (this.radius \* this.radius);  }   @Override  public double getParameter() {  return 2 \* Math.*PI*\*radius;  } } |

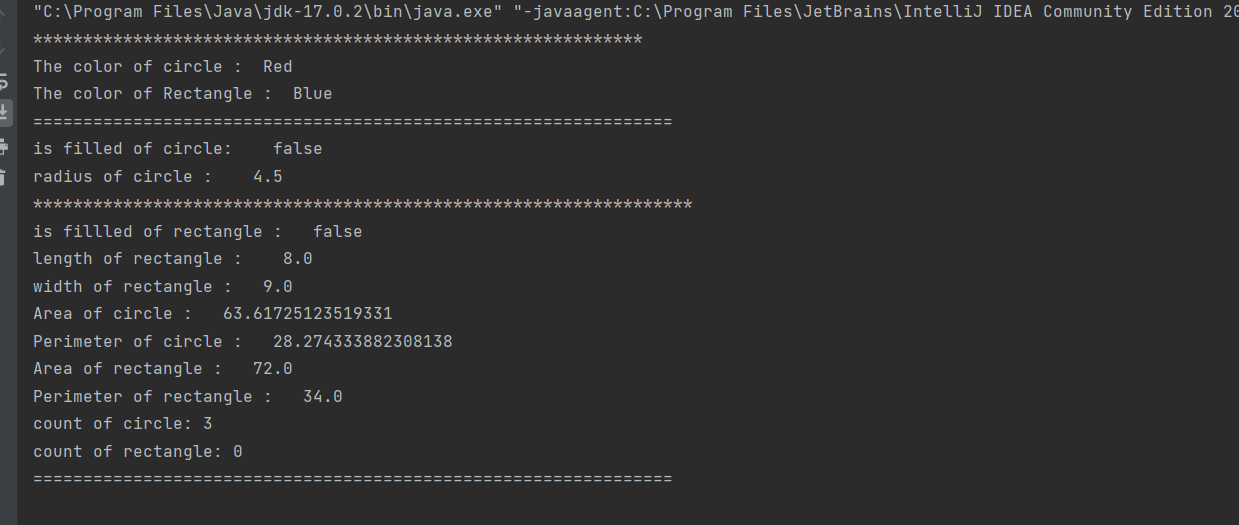
**Rectangle Class:**

|  |
| --- |
| package Assignment\_2\_Part\_2;  public class Rectangle extends Shape{  //---------------instance varible------  private double length;  private double width;   //------------------------------------------------- CONSTRUCTORS ---------------------------------------------------  public Rectangle(color shapeColor, boolean isFilled, int length, int width) {  super(shapeColor, isFilled);  this.width = width;  this.length = length;  }   //------------------------------------------------- GETTERS AND SETTERS ---------------------------------------------------   public double getLength() {  return length;  }   public void setLength(double length) {  this.length = length;  }   public double getWidth() {  return width;  }   public void setWidth(double width) {  this.width = width;  }   @Override  public double getParameter() {  return 2 \* (length + width);  }   @Override  public double getArea() {  return length \* width;  } } |

**Shape Test Class:**

|  |
| --- |
| package Assignment\_2\_Part\_2;  public class ShapeTest {  public static void main(String[] args) {  Circle circle1 = new Circle(Shape.color.*Red*,false,4.5);  Rectangle rect1 = new Rectangle(Shape.color.*Blue*,true, 8,9);  System.*out*.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.*out*.println("The color of circle : "+" "+circle1.getShapeColor());  System.*out*.println("The color of Rectangle : "+" "+rect1.getShapeColor());   System.*out*.println("================================================================");    System.*out*.println( "is filled of circle:" + " " + circle1.isFilled() + " " + "\nradius of circle :" + " " + circle1.getRadius());  System.*out*.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.*out*.println( "is fillled of rectangle :" + " " + circle1.isFilled()+ " " + "\nlength of rectangle :" + " " + rect1.getLength() + " " + "\nwidth of rectangle :" + " " + rect1.getWidth());       System.*out*.println("Area of circle :" +" " + circle1.getArea());  System.*out*.println("Perimeter of circle :"+" " + circle1.getParameter());   System.*out*.println("Area of rectangle :"+" "+ rect1.getArea());  System.*out*.println("Perimeter of rectangle :"+" " + rect1.getParameter());   *printEachShapeCount*(circle1, circle1, circle1);   System.*out*.println("================================================================");   }   public static void printEachShapeCount(Shape... shapes) {  int countCircle = 0;  int countRectangle = 0;  for (Shape i : shapes) {  if (i instanceof Circle) {  countCircle++;  }  if (i instanceof Rectangle) {  countRectangle++;  }  }  System.*out*.println("count of circle: " + countCircle);  System.*out*.println("count of rectangle: " + countRectangle);  }   } |

**Ouput:**

****