USN: 2GI19CS175 Student Name: Venkatesh G Dhongadi

**Title of the Experiment:** Implementation of class and its member methods

**Experiment No.**4 **Date:** 25/10/2020

**Problem Statement:**

A company has two types of employees – FullTime and Partime. The company records for each employee his/her name, age, address, salary and gender. Given the basic salary of the FullTime employee the components of his/her gross salary are: Dearness allowance – 75% of basic salary, HRA – 7.5% of basic salary, IT – 10% of basic. The salary of a Partime employee is dependent on the qualification, experience, number of working hours and the rate per hour, as below:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Qualification | | |
| Experience | BE | MTech | Ph.D. |
| 1-5 years | 300 Rs. | 500 Rs. | 800 Rs. |
| 6-10 years | 400 Rs. | 700 Rs. | 1200 Rs. |
| >10 years | 500 Rs. | 1000 Rs. | 1500 Rs. |

Model this as a problem of hierarchical inheritance by:

1) Identifying the super class with its data members and member functions.

2) Identify the sub-class/sub-classes and their associated data members and member functions.

Test the program by creating objects of the classes that are so identified.

**Objectives of the Experiment:**

1. Learn how inheritance works in Java
2. Learn how to create base and derived classes
3. Understand the use of extends keyword
4. Understand the use of inheritance in a real-life application
5. Learn how to base class constructors in derived class
6. Learn about abstract classes and method overriding
7. Learn to Display the result in a readable/proper format

**Program Source Code:**

abstract class Employee

{

    String name;

    int age;

    String address;

    char gender;

    double salary;

    Employee(String name,int age,String address, char gender)

    {

        this.name = name;

        this.age = age;

        this.address = address;

        this.gender = gender;

    }

    void showDetails()

    {

        System.out.println("Name : "+name);

        System.out.println("Age :"+age);

        System.out.println("Address :"+address);

        System.out.println("Gender : "+gender);

        System.out.println("Salary :"+salary);

    }

    abstract void computeSalary();

}

class FTEmployee extends Employee

{

    double basic;

    FTEmployee(String name,int age,String address, char gender,double basic)

    {

        super(name,age,address,gender);

        this.basic = basic;

    }

    @Override

    void computeSalary()

    {

        double da,hra,tax;

        da = basic \* 0.75;

        hra = basic \* 0.075;

        tax = basic \* 0.1;

        salary = basic + da + hra - tax;

    }

}

class PTEmployee extends Employee

{

    String qual;

    int exp,hrsWorked;

    PTEmployee(String name,int age,String address, char gender,String qual,int exp,int hrsWorked)

    {

        super(name,age,address,gender);

        this.qual = qual;

        this.exp = exp;

        this.hrsWorked = hrsWorked;

    }

    @Override

    void computeSalary()

    {

        switch(qual)

        {

            case "BE":

                if(exp>=1 && exp<=5)

                    salary = hrsWorked \* 300;

                else if(exp<=10)

                    salary = hrsWorked \* 400;

                else

                    salary = hrsWorked \* 500;

                break;

            case "Mtech":

                if(exp>=1 && exp<=5)

                    salary = hrsWorked \* 500;

                else if(exp<=10)

                    salary = hrsWorked \* 700;

                else

                    salary = hrsWorked \* 1000;

                break;

            case "phD":

                if(exp>=1 && exp<=5)

                    salary = hrsWorked \* 800;

                else if(exp<=10)

                    salary = hrsWorked \* 1200;

                else

                    salary = hrsWorked \* 1500;

                break;

        }

    }

}

public class app {

public static void main(String[] args) {

        FTEmployee f1 = new FTEmployee("Rohit M Borse",007,"Bhagya Nagar, Belagavi",'M',59003);

        PTEmployee p1 = new PTEmployee("Sanket M Mungarwadi",23,"Mahantesh Nagar",'M',"BE",12,1000);

        f1.computeSalary();

        f1.showDetails();

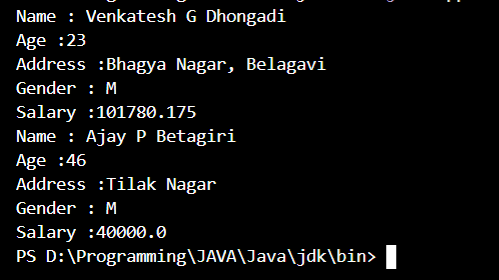
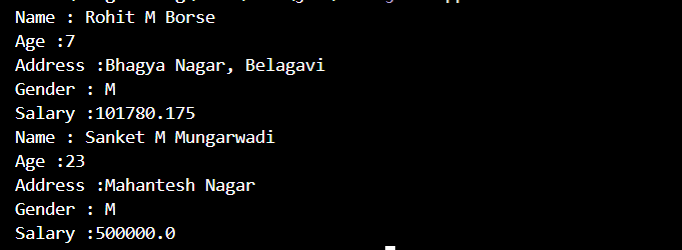
        p1.computeSalary();

        p1.showDetails();

    }

}

**OUTPUT:**



**Outcomes of the Experiment:** At the end of the laboratory sessions the students should be able to

1. Demonstrate the use inheritance in solving real-life problems.
2. Identify appropriate type of inheritance to use for a particular scenario
3. Creating derived classes using extends keyword
4. Learn how to call super class constructors
5. Identify how to implement inheritance in real life applications

**Conclusions:**  From the given problem statement, we could identify the necessary methods and use the appropriate type of inheritance and the necessary program logic. We understood how to calculate the salary of different types of employees based on their qualification. The program was written in Visual Studio Code by creating a project. We understood the usage of the IDE in typing the code, debugging, running the program and observing the output. We also understood the use of the built-in class System and its method println to display the result. The program was executed for two-three sets of input and results obtained were verified to be correct and recorded.

**PRACTICE PROBLEM**

The class Cylinder inherits all the instance variables (radius and color) and methods (getRadius(), getArea(), among others) from its superclass Circle. It further defines a variable called height, three methods getHeight(), setHeight() and getVolume() and its own constructors. Implement the hierarchy as shown below:

Circle

radius:double = 1.0

color:String = “Red”

Circle()

Circle(radius:double)

Circle(radius:double, color:String)

getRadius():double

setRadius(radius:double):void

getColor():String

getColor(color:String):void

getArea():double

Cylinder

height:double = 1.0

Cylinder()

Cylinder(height:double)

Cylinder(height:double, radius:double)

Cylinder(height:double, radius:double,

color:String)

getHeight():double

setHeight(height:double):void

getVolume():double

**Program Source Code:**

import java.lang.\*;

import java.io.\*;

import java.util.\*;

class Circle{

    double radius;

    String color;

    Circle(){

        radius = 1.0;

        color="Aqua";

    }

    Circle(double radius){

        this.radius=radius;

        color="Grey";

    }

    Circle(double radius, String color){

        this.radius=radius;

        this.color=color;

    }

    double getRadius() {

        return radius;

    }

    void setRadius(double radius) {

        this.radius=radius;

    }

    String getColor() {

        return color;

    }

    void setColor(String color) {

        this.color=color;

    }

    double getArea() {

        return (Math.PI\*radius\*radius);

    }

}

//subclass

class Cylinder extends Circle{

    double height;

    Cylinder(){

        super();

        height=1.0;

    }

    Cylinder(double height){

        super();

        this.height=height;

    }

    Cylinder(double height, double radius){

        super(radius);

        this.height=height;

    }

    Cylinder(double height, double radius, String color){

        super(radius,color);

        this.height=height;

    }

    double getHeight() {

        return height;

    }

    void setHeight(double height) {

        this.height=height;

    }

    double getVolume() {

        return (Math.PI\*radius\*radius\*height);

    }

}

public class app {

    public static void main(String[] args) {

        Circle c=new Circle(3.0,"Aqua");

        System.out.println("Radius of circle = "+c.getRadius()+"\nColor of Circle = "+c.getColor());

        c.setColor("Blue");

        System.out.println("Changed Color of the cirlce : "+c.getColor()+(String.format("\nThe Area of the circle : %.2f",c.getArea())));

        Cylinder c1=new Cylinder(3.0,4.0,"Black");

        System.out.println("Radius of the Cylinder : "+c1.getRadius()+"\nHeight of the Cylinder : "+c1.getHeight()+"\nColor of the Cylinder : "+c1.getColor()+String.format("\nVolume of the Cylinder : %.2f",c1.getVolume()));

    }

}

**OUTPUT:**



