1 Solve this.

Fresh business scenario to apply inheritance , polymorphism to emp based organization scenario.

Create Emp based organization structure --- Emp , Mgr , Worker

All of above classes must be in --com.app.org

1.1 Emp state--- id(int), name, deptId , basicSalary(double)

Accept all of above in constructor arguments.

Methods ---1.To get emp details -- override toString((inherited from Object class)

1.2. compute net salary ---ret 0

(eg : public double computeNetSalary(){return 0;})

1.2 Mgr state ---id,name,basic,deptId , perfBonus

Add suitable constructor

Methods ----1. get mgr details : override toString (inherited from Emp class)

1. compute net salary (formula: basic+perfBonus) -- override computeNetSalary

2. get performance bonus. --add a new method to return bonus.(getter)

1.3 Worker state --id,name,basic,deptId,hoursWorked,hourlyRate

Methods :

1. get worker details -- : override toString. (inherited from Emp class)

2. compute net salary (formula: = basic+(hoursWorked\*hourlyRate) --override computeNetSalary

3. get hrlyRate of the worker -- add a new method to return hourly rate of a worker.(getter)

Organize classes in inheritance hierarchy.

Write TestOrganization in "tester" package.

Create suitable array to store organization details.

Provide following options

1. Hire Manager

I/P : all manager details

2. Hire Worker

I/P : all worker details

3. Display information of all employees(toString) & display net salary (by invoking computeNetSal),

4. Exit

EMPLOYEE

**package** com.app.org;

**public** **class** Employee {

**private** **int** empid,deptId;

**private** String name;

**protected** **double** Salary;

**public** Employee() {

**super**();

System.***out***.println("-----Default Constructor of Employee----");

}

**public** Employee(**int** empid,String name,**int** deptId,**double** Salary)

{

**this**.Salary=Salary;

**this**.deptId=deptId;

**this**.empid=empid;

**this**.name=name;

}

**public** **int** getEmpid() {

**return** empid;

}

**public** String getName() {

**return** name;

}

**public** String toString() {

**return** "Employee Id :- "+empid+"\nName :- "+name+"\nDepartment Id :- "+deptId+"\nSalary :- "+Salary;

}

**public** **double** computeNetSal() {

**return** Salary;

}

}

MANAGER

**package** com.app.org;

**public** **class** Manager **extends** Employee{

**private** **double** perfBonus;

**public** Manager() {

**super**();

}

**public** Manager(**int** empid,String name,**int** deptId, **double** Salary,**double** perfBonus) {

**super**(empid,name,deptId,Salary);

**this**.perfBonus=perfBonus;

}

**public** String toString() {

**return** **super**.toString()+"\nPerformance Bonus :- "+perfBonus;

}

**public** **double** computeNetSal()

{

**return** Salary+perfBonus;

}

**public** **double** perfBonus() {

**return** perfBonus;

}

}

WORKER

**package** com.app.org;

**public** **class** Worker **extends** Employee {

**private** **double** hoursWorked;

**private** **double** hourlyRate;

**public** Worker() {

**super**();

}

**public** Worker(**int** empid, String name, **int** deptId, **double** Salary, **double** hoursWorked, **double** hourlyRate) {

**super**(empid, name, deptId, Salary);

**this**.hourlyRate = hourlyRate;

**this**.hoursWorked = hoursWorked;

}

**public** **double** getHourlyRate() {

**return** hourlyRate;

}

**public** **void** setHoulryRate(**double** hourlyRate) {

**this**.hourlyRate=hourlyRate;

}

**public** **double** computeNetSalary() {

**return** **super**.Salary+(hoursWorked\*hourlyRate);

}

**public** String toString() {

**return** **super**.toString()+"\nHours Worked :- "+hoursWorked+"\nHourly Rate :- "+hourlyRate;

}

}

LOGIC FOR MENU DRIVEN

**package** com.code.tester;

**import** java.util.Scanner;

**import** com.app.org.Employee;

**import** com.app.org.Manager;

**import** com.app.org.Worker;

**public** **class** logic {

**static** Scanner *sc* = **new** Scanner(System.***in***);

**static** **int** *index* = 0;

**static** Employee[] *e* = **new** Employee[5];

// Hire Manager

**public** **static** **void** hiremanager() {

**if** (*index* < *e*.length) {

System.***out***.println("Enter Manager Details :\n");

System.***out***.println("'Employee Id' 'Name' 'DeptId' 'Salary' 'Performance Bounus' ");

*e*[*index*++] = **new** Manager(*sc*.nextInt(), *sc*.next(), *sc*.nextInt(), *sc*.nextDouble(), *sc*.nextDouble());

System.***out***.println("\nManager Added Successfully!!!");

} **else**

System.***out***.println("Array is full!!!");

}

//Hire Worker

**public** **static** **void** hireworker() {

**if** (*index* < *e*.length) {

System.***out***.println("Enter Worker Details:\n");

System.***out***.println("'Employee Id' 'Name' 'DeptId' 'Salary' 'Hourly Rate' 'Hours Worked' ");

*e*[*index*] = **new** Worker(*sc*.nextInt(), *sc*.next(), *sc*.nextInt(), *sc*.nextDouble(), *sc*.nextDouble(),

*sc*.nextDouble());

*index*++;

System.***out***.println("Worker Added Successfully!!!");

} **else**

System.***out***.println("Array is full!!!");

}

//Display

**public** **static** **void** display() {

**boolean** status=**false**;

**for** (**int** i = 0; i < *e*.length; i++) {

**if** (*e*[i] != **null**) {

System.***out***.println("\nEmployee" + (i + 1) + ":-\n\n" + *e*[i]);

**if** (*e*[i] **instanceof** Manager)

System.***out***.println("Net Salary of Manager :- " + ((Manager) *e*[i]).computeNetSal());

**else**

System.***out***.println("Net Salary of Worker :- " + ((Worker) *e*[i]).computeNetSalary());

status=**true**;

}

}

**if** (status == **false**)

System.***out***.println("Array is empty");

}

//Fire

**public** **static** **void** fire() {

System.***out***.println("To fire an employee, enter employee id :- ");

**int** temp\_id=*sc*.nextInt();

**boolean** status = **false**;

**for**(**int** i=0;i<*e*.length;i++) {

**if**(*e*[i]!=**null**) {

**if**(*e*[i].getEmpid()==temp\_id)

{

**if** (*e*[i] **instanceof** Manager)

{System.***out***.println(*e*[i].getName()+" Fired!!!"+" Designation:- Manager");

*e*[i]=**null**;}

**else**

{System.***out***.println(*e*[i].getName()+" Fired !!! "+"Designation :- Worker");

*e*[i]=**null**;}

status=**true**;

}

**else**

status=**false**;

}

**continue**;

}

**if**(status=**false**)

System.***out***.println("Employee id not found!!!\n");

*sc*.close();

}

}

MENU DRIVEN

**package** com.code.tester;

**import** java.util.Scanner;

**public** **class** Test {

**public** **static** **void** main(String[] args) {

**int** ch=0;

Scanner sc = **new** Scanner(System.***in***);

**do** {

System.***out***.println(" \n1.Hire Manager 2.Hire Worker" + " 3.Display Details 4.Fire Employee 5.Exit ");

System.***out***.println("\nEnter Choice :- ");

**if** (sc.hasNextInt())

ch = sc.nextInt();

**switch** (ch) {

**case** 1:

logic.*hiremanager*();

**break**;

**case** 2:

logic.*hireworker*();

**break**;

**case** 3:

logic.*display*();

**break**;

**case** 4:

logic.*fire*();

**break**;

**case** 5:

System.***out***.println("Exiting...");

System.*exit*(1);

**default** :

System.***out***.println("Enter valid choice!!!");

}

} **while** (ch != 5);

sc.close();

}

}









