Page No.
Date

Devanshu Surana
PC-23, 1032210755
Batch C1

OOCJ Lab Assignment - 2B

Problem statement:
Write a statement in Java showing hierarchical inheritance ls with base class as employee and derived classes as FullTime-Employee and InternEmployee with methods DisplaySalary in Base Class and Calculatesalary in derived classes. Calculate salary method will calculate as per increment given to full time and intern Employees. Full time employee - 50% hike, InternEmployees - 25% hike. Display salary before and after hike.

Objective:
1. To study inheritance in Java
2. To study why to use inheritance
3. To study types of Inheritance

Theory:
Inheritance in Java: It is a mechanism in which one object acquires all the properties and behaviours of a parent object. The idea behind inheritance in Java is that we can create new classes that are built upon existing classes. When we inherit from an existing class, we can reuse methods and fields of the parent class, we Moreover we can add new methods and fields in your current class also.
We use inheritance in Java for method overriding

code reusability and for Abstraction also

Types:

1. Single Inheritance:
Subclasses inherit the features of one supe

2. Multilevel Inheritance: A derived class is inheriting a base class, and as well as class also acts as the base class for

3. Hierarchichal Inheritance: One class serves as superclass for more than one subclass.

4. Multiple Inheritance: Java does not support Inheritance with classes. In Java, we can ach multiple Inheritance only through Interfaces.

5. Hybrid Inheritance: Combination of more th types of Inheritances, single and multiple can be achieved through Interfaces only as Inheritance is not supported by Java.

Conclusion:

Thus, we have successfully implemented Inheritance in Java.

FAQ's

1) Java does not support multiple Inheritance with classes. In Java we can achieve multiple inheritance only through Interfaces.

An Interface is a blueprint of a class that provides a set of abstract methods that a class must implement.

2) Whenever one class inherits another class, it is called as IS-A relationship.

IS-A relationship is completely related to Inheritance.

IS-A relationship is additionally used for code reusability in Java and to avoid code redundancy it can simply be achieved by extending an Interface or class by using extend keyword.

3) Constructors and Initializers both static intilalization block and instanc Initialization block are not inherited to the subclass but they get executed during the instantiation of the subclass.

Algorithm

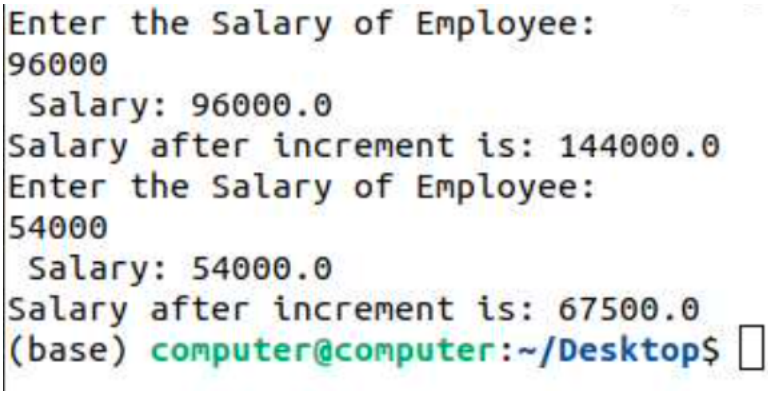
```
step1: start
step 2: create base class employee
step 3: create derived classes using extend keyword
step 4: Calculate salary by giving 50 % hike to fulltime
            and 25 % to intern
step 5: Display salary after hike
step 6:    End
```

Name :- Devanshu Surana
Roll No: PC -23
PRN :- 1032210755

OOCCJ LAB ASSIGNMENT 2B

```
import java.io.*;
import java.lang.*;
import java.util.*;

class employee{
Scanner myObj= new Scanner(System.in);

double salary;

void getsalary(){
System.out.println("Enter the Salary of Employee: ");
salary=myObj.nextDouble();
System.out.println(" Salary: " + salary);
}
void displaysalary(){
System.out.println("Salary after increment is: " + salary);
}



}
class InternEmployee extends employee{

void calculatesalary(){
salary = 1.25 * salary;

}
}
```

```
class PermEmployee extends employee{

void calculatesalary(){
salary = 1.50 * salary;

}
}
class 2b{
   public static void main(String args[]){

   PermEmployee e1=new PermEmployee();
   InternEmployee e2=new InternEmployee();
   e1.getsalary();
   e1.calculatesalary();
   e1.displaysalary();
   e2.getsalary();
   e2.calculatesalary();
   e2.displaysalary();
   }
}
```



```
Enter the Salary of Employee:
96000
 Salary: 96000.0
Salary after increment is: 144000.0
Enter the Salary of Employee:
54000
 Salary: 54000.0
Salary after increment is: 67500.0
(base) computer@computer:~/Desktop$ 
```