Experiment no.6

date: 14/8/24

Roll no: B014

Aim: To implement single, multilevel and hierarchical inheritance.

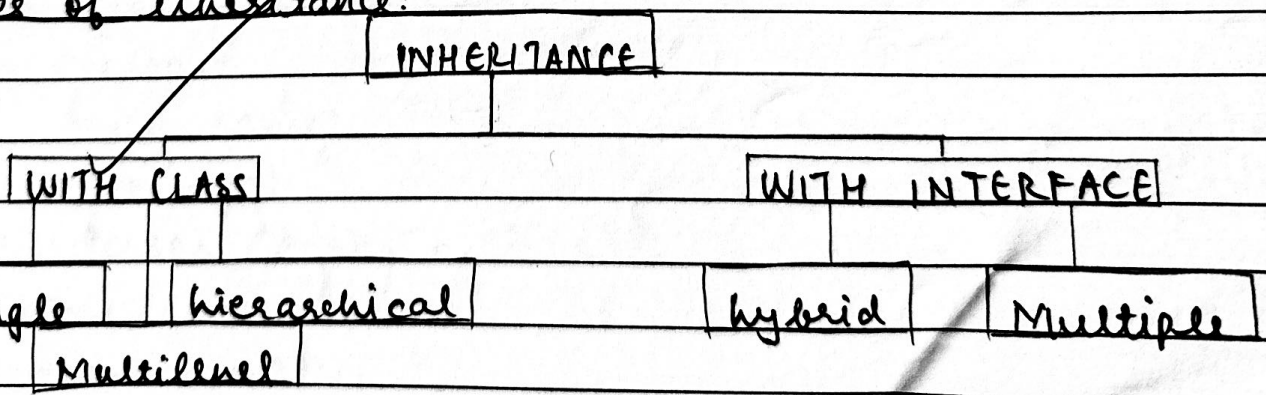
Theory:

Inheritance- Inheritance in Java is a mechanism in which one object acquires all the properties and behaviours of a parent object.

Need of inheritance in Java:

- Inheritance allows programmers to create classes that are built upon existing classes, this helps in code reuseability.
- The child class is able to inherit the properties of the parent class.
- With inheritance only, method overriding can be performed; thus inheritance is important in Java.
- The base class can be extended as per need.

Types of inheritance:



Method overriding: . If subclass (child class) has the same method declared in the parent class, it is known as method overriding. It is a feature that allows sub-class to provide a specific implementation of a method that is already provided by one of its super-classes.

. Method overriding is one of the ways through which Java achieves 'Run time Polymorphism'.

example: Consider a class vehicle; we have defined the 'run' method in the subclass as defined in the super class, but, it has some specific implementation. The name and parameter of the method are the same, and there is a relationship between the classes, so there is method overriding.

procedure:

Program 1- Implement a base class person and student as the derived class. Take essential data fields, constructor and method.

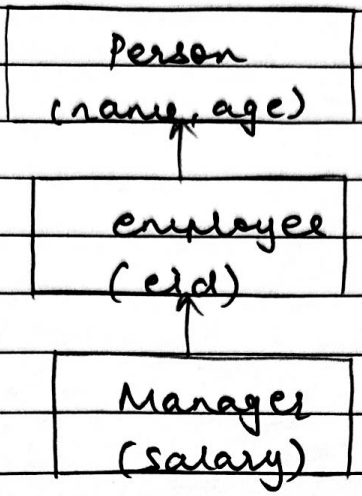
```
import java.util.Scanner;
class Person {
    String name;
    int age;
public Person (String n, int a){
    this.name = n;
    this.age = a;
}
public void display (){
    System.out.println("name: "+name+ "age: " +age);
```

```
}}
class Student extends Person {
    String studid;
    float percentage;

public Student (String n, float a, String id, float p){
 super (n, a);
  this.studid = id;
  this.percentage = p;
}
public void displayinfo(){
  System.out.println("student id : " + studid + " percentage :"
                                        + percentage);
}}
public class Inherit {
  public static void main (String[] args){
   Scanner sc = new Scanner (System.in);
  System.out.println ("enter student id : ");
  String id = sc.next();
  System.out.println ("enter percentage : ");
  float p = sc.nextFloat();
  System.out.println ("enter name : ");
   String n = sc.next();
  System.out.println ("enter age : ");
   int a = sc.nextInt();
  Student s = new Student (id, p, n, a);
  s.display();
  s.displayinfo();
 }
}
```

output:-

```
enter student id:
Z123

Enter percentage:
89

enter name:
Rudraa

Enter age:
16

name: rudraa age: 16
studentid: Z123 percentage: 89
```

program 2:



Implement the inheritance shown in above figure. Use concept of method overriding to override the display method. Also, use super keyword.

```
import java.util.Scanner;
class Person {
    String name;
    int age;
public Person(String n, int a){
    this.name=n;
    this.age=a;
}
public void display(){
   System.out.println("name: " + name + "age: "+age);
}}

class Employee extends Person{
    String eid;
public Employee(String n, int a, String id){super(n,a);
    this.eid = id;
}
```

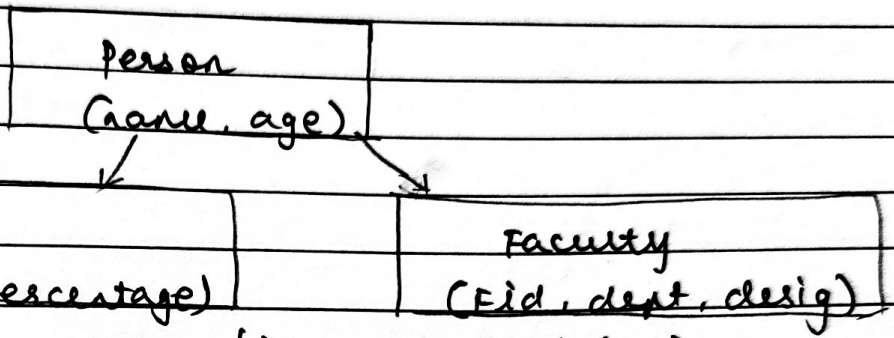
```
@Override
 public void display(){
  super.display();
  System.out.println("Eid : " +eid);
}
}

class Manager extends Employee{
    int salary;
public Manager (String n, int a, String id, int s){
    super(n, a, id);
     this.salary = s;
}
@Override
public void display(){
    super.display();
  System.out.println("salary : " +salary);
}
}
public class multilevel{
 public static void main(String[] args){
     Scanner sc = new Scanner(System.in);
 System.out.println("enter name: ");
  String n = sc.next();
 System.out.println("Enter age: ");
   int a = sc.nextInt();
 System.out.println("Enter id: ");
  String id = sc.next();
 System.out.println("enter salary: ");
   int s = sc.nextInt();
```

output:

```
enter name:
  rudraa
enter age:
   16
enter id:
   2543
enter salary:
  40,000

name: rudraa   age: 16
Eid: 2543      salary: 40,000
```

```
Manager m = new Manager (n, a, id, s);
m.display();
}
}
```

program 3:



Implement the shown hierarchical inheritance. Demonstrate method overriding of the same. Use super keyword.

```
import java.util.Scanner;
class Person {
    String name;
    int age;
public Person (String n, int a) {
    this.name = n;
    this.age = a;
}
public void display() {
    System.out.println("name: " + name + "age: " + age);
}
}

Class student extends Person {
    String id;
```

```
    int rollno;
    float percentage;

public Student(String n, int a, String id, int s, float p){
    super(n, a);
    this.id = id;
    this.rollno = s;
    this.percentage = p;
}
@Override
public void display(){
    super.display();
    System.out.println("id: " + id + "rollno: " + rollno +
                       "percentage: " + percentage);
}
}

class Faculty extends Person{
    String id;
    String dept;
    String designation;

public Faculty(String n, int a, String id, String d, String dl
  { super(n, a, id);
    this.dept = d;
    this.designation = dl;
}
@Override
public void display(){
    super.display();
```

```
        System.out.println("id: " + id + "Dept: " + dept +
                           "Designation: " + designation);
    }
}
public class inherit {
    public static void main (String[] args) {
        Scanner sc = new Scanner (System.in);
    System.out.println("enter name: ");
        String n = sc.next();
    System.out.println ("enter age: ");
        int a = sc.nextInt();
    System.out.println ("enter id: ");
        String id = sc.next();
    system.out.println("enter rollno: ");
        int r = sc.nextInt();
    System.out.println("enter percentage: ");
        float p = sc.nextFloat();
    system.out.println ("enter dept: ");
        String d = sc.next();
    System.out.println("enter designation: ");
        String dl = sc.next();

    System.out.println("Student info: ");
    Student s = new Student(n, a, id, r, p);
        s.display();

    System.out.println("Faculty info: ");
        Faculty f = new Faculty (n, a, id, d, dl);
        f.display();
    }
}
```

output:-

```
enter name:
  rudraa
enter age:
  16
enter rollno:
    I541
enter percentage:
    89
enter dept:
     CSE
enter designation:
    abc

Student info:
name: rudraa  age: 16
id: I541  rollno: 14
percentage: 89
Faculty info:
name: rudraa  age: 16
id: I541  dept: CSE
designation: abc
```

Conclusion: Hence, we have implemented programs on single, multilevel and hierarchical inheritance in Java.

21/8/21