11th Nov Live Class Static and Inheritance in Java

INSTRUCTOR: Hyder

Topic: static and inheritance in java

horizontal line

1. Instance variables memory will be allocated inside every object separately, how many objects you create that many times memory will be allocated.
2. For static variables memory will be allocate only once. The same value will be used among all the objects.
3. Static methods are also called as generic methods, and the non static methods are also called as specific methods since they are object specific.
4. Below is writing a program for loan app

| import java.util.Scanner;  class Farmer{  private float pa;  private float td;  private float si;  private static float ri; // since rate of in is comman to all hence we can maek it as static varible.  // static variable write inside the static block  static{  ri=2.5f; // this is the professonal way of writing for the static varible  }  void input(){  Scanner scan = new Scanner(System.in);  System.out.println("kindly enter the principal amount");  pa = scan.nextFloat();  System.out.println("kindly mention time during requried");  td= scan.nextFloat();  }  void compute(){  si=(pa\*td\*ri)/100;  }  void disp(){  System.out.println("SI is" + si);  }  }  public class LaunchLoan {  public static void main(String[] args) {  Farmer f1 = new Farmer();  Farmer f2 = new Farmer();  f1.input();  f1.compute();  f1.disp();  f2.input();  f2.compute();  f2.disp();  }    } |
| --- |

Inheritance(code reusabulity)concept starting here:-

1. In java between two class we can establish the relationship with extends keyword. The process of occurring the properties of another class using the extends keyword is called as “inheritance”. Before extends keyword what ever class name(which takes the property) you have that is called as subclass/ derived class/ child class. And what ever class you mention after the extends keyword is called as parent class/ base class/ superclass. With the help of extends keyword we will get he is-a relation ship.

Example:

class Demo1 {

String name = "hyder";

int age = 28;

void disp() {

System.out.println("demo1 " + age + name);

}

}

class Demo2 extends Demo1 {

}

public class Inheritance {

public static void main(String[] args) {

Demo2 d = new Demo2();

d.disp();

}

}

1. Java supports two types of relation ships those are is-a parent-child relation ship and has-a relation ships. Refer: fig1
2. Some impartenet points related to inheritance are
   1. Single inheritance is allowed.
   2. Object class is parent of all classes
   3. Multilevel inheritance is allowed example below program.

class Demo11 {

String name = "hyder";

int age = 28;

void disp() {

System.out.println("demo1 " + age + name);

}

}

class Demo12 extends Demo11{

}

class Demo13 extends Demo12{

}

class Demo14 extends Demo13{

}

public class Inheritance {

public static void main(String[] args) {

Demo14 d = new Demo14();

d.disp();

}

}

* 1. One base/parent class can have any no of child class. That is called as hirarcular inheritance allowed.

Example:

​​class Demo111 {

String name = "hyder";

int age = 28;

void disp() {

System.out.println("demo1 " + age + name);

}

}

class Demo112 extends Demo111{

}

class Demo113 extends Demo111{

}

class Demo114 extends Demo112{

}

public class Inheritance {

public static void main(String[] args) {

Demo113 d = new Demo113();

d.disp();

}

}

* 1. Multiple inheritance is not allowed. (two fathers). This is called as diamond shaped problem. fig: 2,3
  2. Cyclic inheritance is not allowed.(father is telling you are my father and child is telling no no you are my father). Fig 4
  3. Private members of the class does does not participate in inheritance why because to preserve encapsulation. Fig:5
  4. Constructors will not participate in inheritance. Why because constructors get called during the object creation. How ever parent class constructor will called because of super method called which is present in the chiid class. fig:6

Next topics are “memory map + inheritance + constructor (super(), this()):

class Parentt{

int a,b,c;

Parentt()

{

a=10;

b=20;

System.out.println("Parent constructor");

}

Parentt(int a, int b)

{

a=10;

b=20;

System.out.println("Parametarised constructor");

}

}

class Childd extends Parentt{

int x,y;

Childd(){

x=100;

y=200;

}

Childd(int x, int y){

this.x=x;

this.y=y;

}

void disp(){

System.out.println(a);

System.out.println(b);

System.out.println(x);

System.out.println(y);

}

}

public class LaunchConst {

public static void main(String[] args) {

Childd ch = new Childd();

ch.disp();

}

}