

Dt : 19/10/2022

***imp**

Specialization process:

=>The process of taking one feature from the Parent and constructing the Child is known as Specialization process.

Specialization using Classes:

=>we use the following syntax to perform specialization process

using Classes:

CClass ob = (CClass)new PClass();

Coding rule:

=>In Specialization process the PClass must be Pre-defined class or Built-in class,else raises "java.lang.ClassCastException".

Ex:

Serialization and De-Serialization process

Cloning process.

Note:

=>Specialization process cannot be performed using Interfaces and AbstractClasses.

Ex:

```

package maccess;

import test.*;

public class DemoSpecialization {

    public static void main(String[] args) {

        System.out.println("****Specialization using Classes****");

        CClass ob1 = (CClass)new PClass();

        ob1.m1(11);

        ob1.m2(12);

        ob1.m3(13);

        //IClass ob2 = (IClass)new ITest();//Error

        //EClass ob3 = (EClass)new AClass();//Error

    }

}

```

=====

Note:

=>Generalization process is also known as UpCasting process or Widening process or Implicit TypeCasting process.

=>Specialization process is also known as DownCasting process or Narrowing process or Explicit TypeCasting process.

=====

faq:

wt is the diff b/w

(i)HAS-A relation

(ii)IS-A relation

(i)HAS-A relation:

=>References concept is known as "HAS-A" relation,because one object "HAS-A" reference of another Object.

(ii)IS-A relation:

=>Inheritance process is known as "IS-A" relation,because CClass is refered by PClass.

=====

***imp**

InnerClasses in Java:

=>The process of declaring the class inside the class is known as InnerClass or Nested Class.

=>These InnerClasses are categorized into two types:

(1)Member InnerClasses

(2)Anonymous InnerClasses

(1)Member InnerClasses:

=>The InnerClasses which are declared as members of class are knwon as Member InnerClasses.

=>These member InnerClasses are categorized into two types:

(a)Static member InnerClasses

(b)NonStatic member InnerClasses

(a)Static member InnerClasses:

=>The member InnerClasses which are declared with "static" keyword are known as Static member InnerClasses or Class member InnerClasses

Coding Rule:

(i)Static member InnerClasses can be declared with both "static" and NonStatic members.

(ii)The instance method which is declared in static member InnerClasses will have behaviour like "static method and can access only static members of OuterClass.

(iii)The static method which is declared in static member InnerClasses can access only static members of OuterClass.

syntax of object creation for static member InnerClasses:

OuterClass_name.InnerClass_name obj=new OuterClass_name.InnerClass_name();

Ex:

SubClass1.SubClass2 ob2 = new SubClass1.SubClass2();

Ex-Program:

SubClass1.java

```
package test;  
public class SubClass1 {  
    public int a=10;//Instance variable  
    public static int b=20;//Static variable
```

```

    public void m1() {
        System.out.println("****OuterClass m1()****");
        System.out.println("The value a:"+a);
        System.out.println("The value b:"+b);
    } //OuterClass method
    public static class SubClass2{
        public void m2() {
            System.out.println("****InnerClass Instance
m2()****");
            //System.out.println("The value a:"+a);
            System.out.println("The value b:"+b);
        }
        public static void m22() {
            System.out.println("****InnerClass static
m2()****");
            //System.out.println("The value a:"+a);
            System.out.println("The value b:"+b);
        }
    } //Static member InnerClass
} //OuterClass

```

DemoInnerClass1.java(MainClass)

```

package maccess;
import test.SubClass1;
public class DemoInnerClass1 {
    public static void main(String[] args) {
        SubClass1 ob1 = new SubClass1(); //OuterClass object
        ob1.m1(); //OuterClass method call
        SubClass1.SubClass2 ob2 = new SubClass1.SubClass2();
        //Static member InnerClass Object
        ob2.m2(); //InnerClass Instance method call
        SubClass1.SubClass2.m22(); //InnerClass Static method
call
    }
}

```

o/p:

******OuterClass m1()******

The value a:10

The value b:20

*****InnerClass Instance m2()*****

The value b:20

*****InnerClass static m2()*****

The value b:20

(b)NonStatic member InnerClasses:

**=>The member InnerClasses which are declared without "static" keyword
are known as NonStatic member InnerClasses**

=>These NonStatic member InnerClasses are categorized into two types:

(i)Instance member InnerClasses

(ii)Local member InnerClasses

(i)Instance member InnerClasses:

**=>The NonStatic member InnerClasses which are declared outside the
method of OuterClass are known as Instance member InnerClasses.**

Coding Rules:

**(i)Instance member InnerClasses can be declared with both "static"
and NonStatic members.**

**(ii)The Instance methods which is declared in Instance member
InnerClasses can access all the members of OuterClass directly.**

**(iii)The Static methods which are declared in Static member
InnerClasses can access only static members of OuterClass.**

syntax of creating object for Instance member InnerClasses:

OuterClass_name.InnerClass_name obj =

OuterClass_Object_name.new InnerClass_name();

Ex:

SubClass1.SubClass2 ob2 = ob1.new SubClass2();

Ex-Program:

SubClass1.java

```
package test;
public class SubClass1 {
    public int a=10;
    public static int b=20;
    public void m1() {
        System.out.println("****OuterClass m1()****");
        System.out.println("The value a:"+a);
        System.out.println("The value b:"+b);
    } //OuterClass method
    public class SubClass2{
        public void m2() {
            System.out.println("****InnerClass Instance
m2() ****");
            System.out.println("The value a:"+a);
            System.out.println("The value b:"+b);
        }
        public static void m22() {
            System.out.println("****InnerClass static m22()****");
            //System.out.println("The value a:"+a);
            System.out.println("The value b:"+b);
        }
    } //Instance member InnerClass
} //OuterClass
```

DemolInnerClass2.java(MainClass)

```
package maccess;
import test.SubClass1;
```

```

public class DemoInnerClass2 {
    public static void main(String[] args) {
        SubClass1 ob1 = new SubClass1(); //OuterClass object
        ob1.m1(); //OuterClass method_call
        SubClass1.SubClass2 ob2 = ob1.new SubClass2();
                                //Instance member InnerClass object
        ob2.m2(); //InnerClass Instance method_call
        SubClass1.SubClass2.m22(); //InnerClass static method_call
    }
}

```

o/p:

******OuterClass m1()******

The value a:10

The value b:20

******InnerClass Instance m2()******

The value a:10

The value b:20

******InnerClass static m22()******

The value b:20

=====