Dt : 12/10/2022

**imp*

Interfaces in Java:

=>Interfaces are collection of variables,abstract methods and concrete methods from Java8 version onwards.

(Upto Java7 version Interfaces are collection of Variables and abstract methods,which means Interface cannot hold concrete methods upto Java7)

faq:

define abstract methods?

=>The methods which are declared without method_body are known as abstract methods.

Structure of abstract methods:

```
return_type method_name(para_list);
```

faq:

define concrete methods?

=>The methods which are declared with method_body are known as Concrete methods.

Structure of Concrete methods:

return_type method_name(para_list)

{

//method_body

}

Coding Rules of Interface:

Rule-1 : we use "interface" keyword to declare interfaces.

syntax:

interface Interface_name

{

//members

}

Rule-2 : The programming components which are declared in interfaces are

automatically "public"

Note:

=>The programming components which are declared in classes without any

access modifier are considered as "default"

Rule-3 : Interfaces can be declared with both Primitive DataType variables

Non-Primitive DataType variables

Rule-4 : The variables which are declared in interfaces are automatically

Static and final Variables.

Note:

(i)Static Variables in interfaces will get the memory within the interface while interface loading and accessed with Interface_name.

(ii)final variables must be initialized with values and once initialized cannot be modified.

faq:

Can we declare NonStatic variables in interfaces?

=>No,we cannot declare NonStatic variables in interfaces because the variables automatically "static".

Rule-5 : The methods which are declared in interfaces are automatically NonStatic abstract methods.

(There is no concept of static abstract methods)

Rule-6 : Interfaces cannot be Instantiated,which means we cannot create object for Interfaces.

Rule-7 : Interfaces are implemented to classes using "implements" keyword and the classes are known as "Implementation classes".

syntax:

class ImplClass implements Interface

{

//members

}

Rule-8 : These implementation classes must construct body for abstract methods of Interface.

Note:

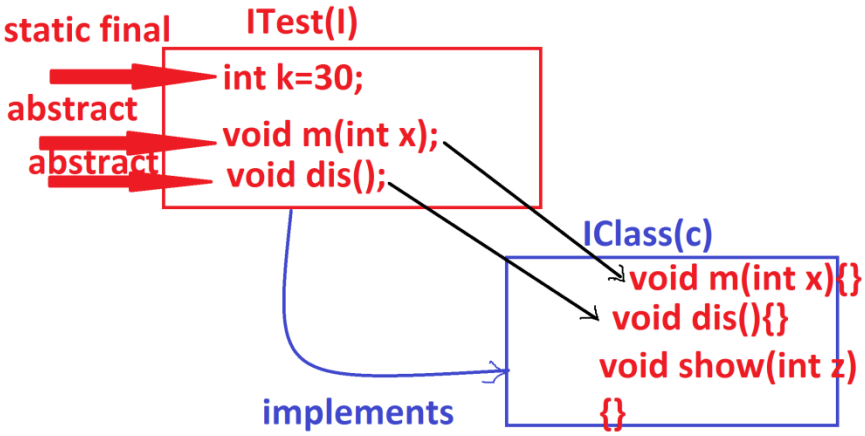
=>We create object for implementation class and the object is known as "Implementation Object".

Rule-9 : Interface canbe declared with any number of abstract methods without restriction.

Rule-10 : Implementation class must construct body for all abstract methods of Interface.

Rule-11 : Implementation classes can also be declared with Non-Implemented methods.

Diagram:



IClass ob = new IClass();

Single Inheritance

Ex:

ITest.java

```
package test;
public interface ITest {
     int k=30;
     void m(int x);
     void dis();
}
```

IClass.java

```
package test;
public class IClass implements ITest{
   public void m(int x)//Implemented and Overriding method
   {
        System.out.println("===m(x)====");
        System.out.println("The value x:"+x);
   }
   public void dis()//Implemented and Overriding method
   {
        System.out.println("====dis()====");
        System.out.println("The value k:"+k);
   }
   public void show(int z)//Non-Implemented method
   {
```

```
        System.out.println("====show(z)====");
        System.out.println("The value z:"+z);
    }
}
```

DemoInterface1.java(MainClass)

```
package maccess;
import test.*;
public class DemoInterface1 {
     public static void main(String[] args) {
        System.out.println("The value k:"+ITest.k);
        //ITest.k=300;//Error
        //ITest ob = new ITest();//Error
        IClass ob = new IClass();//Imple
        ob.m(123);
        ob.dis();
        ob.show(124);
     }
}
```

o/p:

The value k:30

===m(x)====

The value x:123

====dis()====

The value k:30

====show(z)====

The value z:124
