***Dt : 26/9/2022***

***\*imp***

***Concrete methods in Interfaces:(Java8 - new feature)***

***=>From Java8 version onwrads the interface can be declared with***

***the following concrete methods:***

***1.static concrete methods***

***2.default concrete methods***

***3.private concrete methods***

***1.static concrete methods:***

***=>The concrete methods in interfaces which are declared with***

***'static' keyword are known as 'static concrete methods' and which***

***are introduced by Java8 version.***

***Coding Rule:***

***=>Static concrete methods of interfaces are not available to***

***implementation classes,which means Static concrete methods cannot***

***be accessed by Impl\_Classes.***

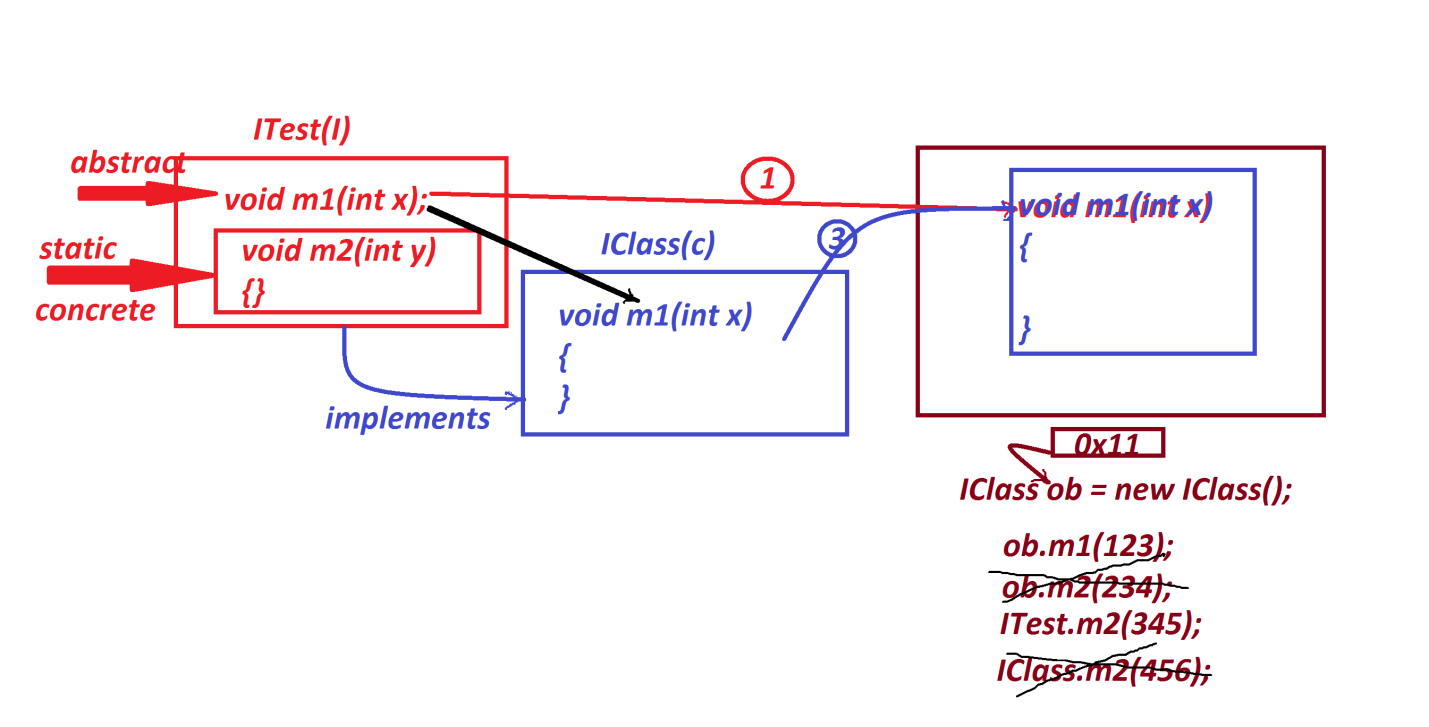
***Note:***

***=>These static concrete methods will get the memory within the***

***interface while interface loading and canbe accessed with***

***interface\_name.***

***Diagram:***

******

***Ex:***

***ITest.java***

***package test;***

***public interface ITest {***

***public abstract void m1(int x);***

***public static void m2(int y) {***

***System.out.println("====static m2(y)===");***

***System.out.println("The value y:"+y);***

***}***

***}***

***IClass.java***

***package test;***

***public class IClass implements ITest{***

***public void m1(int x) {***

***System.out.println("====method m1(x)====");***

***System.out.println("The value x:"+x);***

***}***

***}***

***DemoInterface3.java(MainClass)***

***package maccess;***

***import test.\*;***

***public class DemoInterface3 {***

***public static void main(String[] args) {***

***IClass ob = new IClass();***

***ob.m1(121);***

***// ob.m2(234);//Error***

***ITest.m2(234);***

***// IClass.m2(456); //Error***

***}***

***}***

***o/p:***

***====method m1(x)====***

***The value x:121***

***====static m2(y)===***

***The value y:234***

***==========================================================***

***2.default concrete methods:***

***=>The concrete methods in interfaces which are declared with***

***"default" keyword are known as default concrete methods and which***

***are introduced by Java8 version.***

***Coding Rule:***

***=>default concrete methods are available to implementation classes***

***which means these default concrete methods will get the memory within***

***the object of implementation class and canbe accessed with***

***implementation class object\_name.***

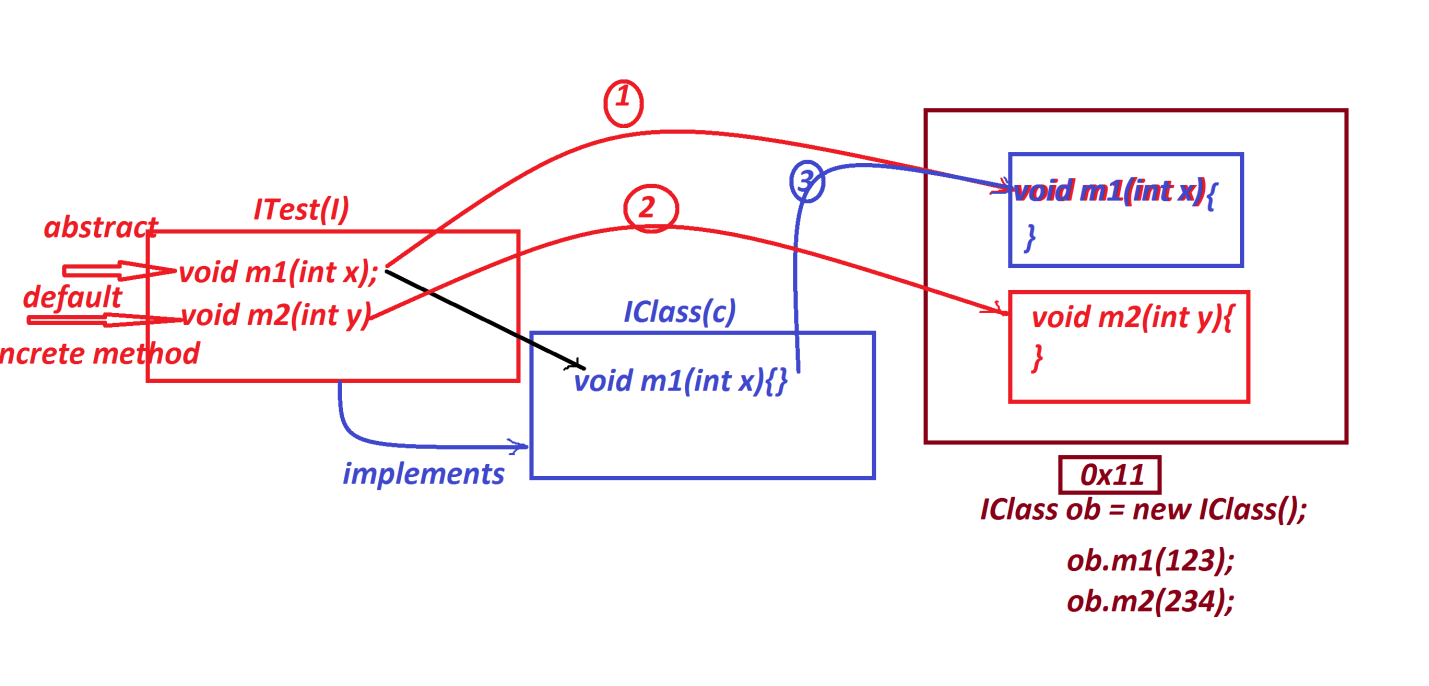
***Note:***

***=>default concrete methods are only NonStatic methods.***

***=>default keyword not available in class,which means we must not***

***use 'default' keyword in classes to declare programming components.***

***Diagram:***

******

***Ex:***

***ITest.java***

***package test;***

***public interface ITest {***

***public abstract void m1(int x);***

***public default void m2(int y) {***

***System.out.println("====default m2(y)===");***

***System.out.println("The value y:"+y);***

***}***

***}***

***IClass.java***

***package test;***

***public class IClass implements ITest{***

***public void m1(int x) {***

***System.out.println("====method m1(x)====");***

***System.out.println("The value x:"+x);***

***}***

***}***

***DemoInterface4.java(MainClass)***

***package maccess;***

***import test.\*;***

***public class DemoInterface4 {***

***public static void main(String[] args) {***

***IClass ob = new IClass();***

***ob.m1(121);***

***ob.m2(234);***

***}***

***}***

***o/p:***

***====method m1(x)====***

***The value x:121***

***====default m2(y)===***

***The value y:234***

***======================================================***

***3.private concrete methods:***

***=>The concrete methods in interfaces which are declared with***

***'private' keyword are known as 'Private Concrete methods' and which***

***are introduced by Java9 version.***

***=>Private Concrete methods are categorized into two types:***

***(i)static private concrete methods***

***(ii)Non-Static private concrete methods.***

***Coding rule:***

***=>Private concrete methods are accessed only inside the interface,***

***which means private concrete methods canbe accessed by NonPrivate***

***methods of same interface.***

***Ex:***

***ITest.java***

***package test;***

***public interface ITest {***

***public abstract void m1(int x);***

***private static void m2(int y) {***

***System.out.println("===private static m2(y)====");***

***System.out.println("The value y:"+y);***

***}***

***private void m3(int z) {***

***System.out.println("===private Non-static m3(z)====");***

***System.out.println("The value z:"+z);***

***}***

***public default void show(int y,int z) {***

***ITest.m2(y);***

***this.m3(z);***

***}***

***}***

***IClass.java***

***package test;***

***public class IClass implements ITest{***

***public void m1(int x) {***

***System.out.println("===method m1(x)====");***

***System.out.println("The value x:"+x);***

***}***

***}***

***DemoInterface5.java(MainClass)***

***package maccess;***

***import test.\*;***

***public class DemoInterface5 {***

***public static void main(String[] args) {***

***IClass ob = new IClass();***

***ob.m1(12);***

***ob.show(13, 14);***

***}***

***}***

***o/p:***

***===method m1(x)====***

***The value x:12***

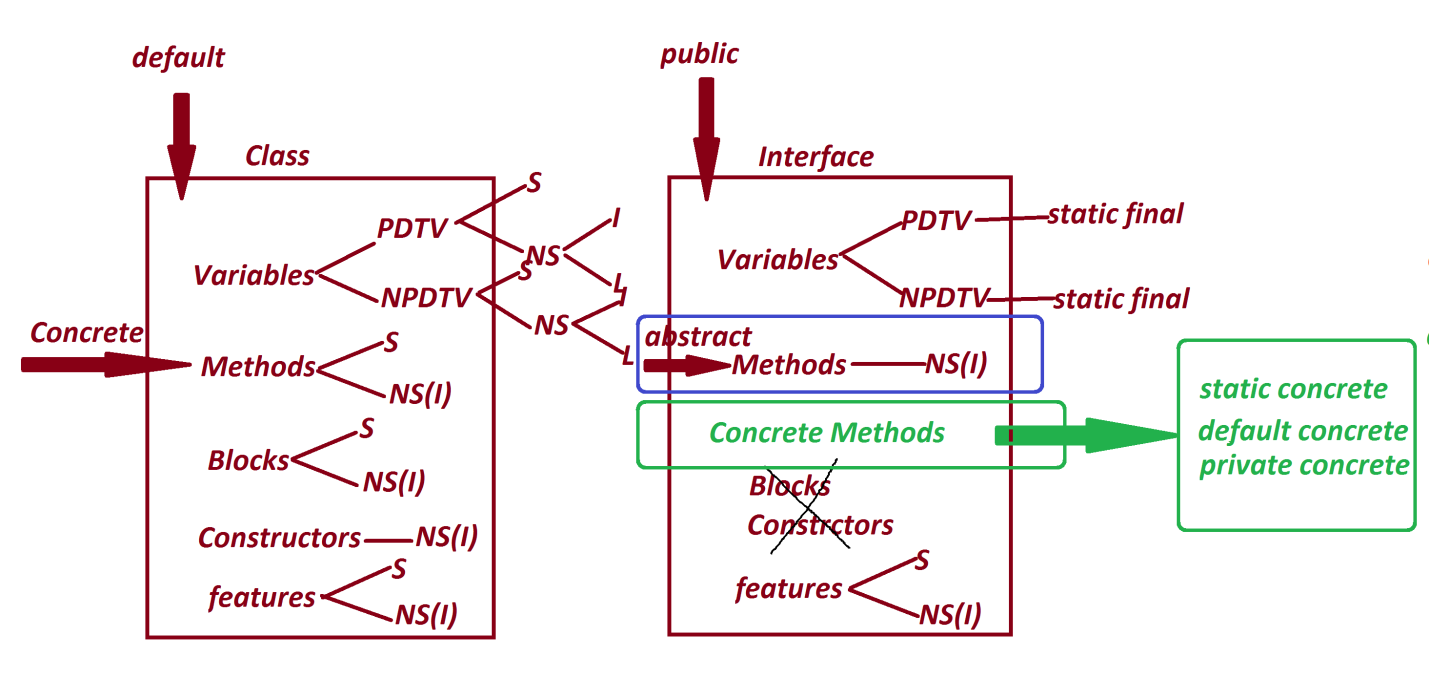
***===private static m2(y)====***

***The value y:13***

***===private Non-static m3(z)====***

***The value z:14***

***=========================================================***

******