

Interface in Java 7 and above

Interface in Java 7

```
interface Interface {  
    void method();  
}  
  
class Cls implements Interface {  
    @Override  
    public void method() { System.out.println("method() in Cls"); }  
}  
  
public class InterfaceEx {  
    public static void main(String[] args) {  
        Interface obj = new Cls();  
        obj.method();  
    }  
}
```

method() in Cls

Java 8 Interface Changes

- Constant variables
- Abstract methods
- Default methods
- Static methods

Default Methods

```
interface Interface {  
    void method();  
    default void newDefaultMethod() { System.out.println("newDefaultMethod() in Interface"); }  
}  
  
class Cls implements Interface {  
    @Override  
    public void method() { System.out.println("method() in Cls"); }  
}  
  
public class InterfaceEx {  
    public static void main(String[] args) {  
        Interface obj = new Cls();  
        obj.method();  
        obj.newDefaultMethod();  
    }  
}
```

```
method() in Cls  
newDefaultMethod() in Interface
```

```
interface Interface {  
    void method();  
    default void newDefaultMethod() { System.out.println("newDefaultMethod() in Interface"); }  
}  
  
class Cls implements Interface {  
    @Override  
    public void method() { System.out.println("method() in Cls"); }  
}  
  
public class InterfaceEx {  
    public static void main(String[] args) {  
        Cls obj = new Cls();  
        obj.method();  
        obj.newDefaultMethod();  
    }  
}
```

```
method() in Cls  
newDefaultMethod() in Interface
```

```

interface Interface {
    void method();
    default void newDefaultMethod() { System.out.println("newDefaultMethod() in Interface"); }
}

class Cls implements Interface {
    @Override
    public void method() { System.out.println("method() in Cls"); }

    @Override
    public void newDefaultMethod() { System.out.println("newDefaultMethod() in Cls"); }
}

public class InterfaceEx {
    public static void main(String[] args) {
        Cls obj = new Cls();
        obj.method();
        obj.newDefaultMethod();

        Interface obj2 = new Cls();
        obj2.method();
        obj2.newDefaultMethod();
    }
}

```

```

method() in Cls
newDefaultMethod() in Cls
method() in Cls
newDefaultMethod() in Cls

```

```

interface Interface {
    void method();
    default void newDefaultMethod() { System.out.println("newDefaultMethod() in Interface"); }
}

class Cls implements Interface {
    @Override
    public void method() { System.out.println("method() in Cls"); }

    @Override
    public void newDefaultMethod() {
        Interface.super.newDefaultMethod();
        System.out.println("newDefaultMethod() in Cls");
    }
}

public class InterfaceEx {
    public static void main(String[] args) {
        Cls obj = new Cls();
        obj.method();
        obj.newDefaultMethod();

        Interface obj2 = new Cls();
        obj2.method();
        obj2.newDefaultMethod();
    }
}

```

```

method() in Cls
newDefaultMethod() in Interface
newDefaultMethod() in Cls
method() in Cls
newDefaultMethod() in Interface
newDefaultMethod() in Cls

```

```

interface Interface {
    void method();
    default void newDefaultMethod() { System.out.println("newDefaultMethod() in Interface"); }
}

interface SubInterface extends Interface {
    void method();
}

class Cls implements SubInterface {
    @Override
    public void method() { System.out.println("method() in Cls"); }
}

public class InterfaceEx {
    public static void main(String[] args) {
        Cls obj = new Cls();
        obj.method();
        obj.newDefaultMethod();

        SubInterface obj2 = new Cls();
        obj2.method();
        obj2.newDefaultMethod();

        Interface obj3 = new Cls();
        obj3.method();
        obj3.newDefaultMethod();
    }
}

```

```

method() in Cls
newDefaultMethod() in Interface
method() in Cls
newDefaultMethod() in Interface
method() in Cls
newDefaultMethod() in Interface

```



```
interface Interface {  
    void method();  
    default void newDefaultMethod() { System.out.println("newDefaultMethod() in Interface"); }  
}  
  
interface SubInterface extends Interface {  
    void method();  
    @Override  
    default void newDefaultMethod() { System.out.println("newDefaultMethod() in SubInterface"); }  
}  
  
class Cls implements SubInterface {  
    @Override  
    public void method() {  
        //Interface.super.newDefaultMethod();  
        System.out.println("method() in Cls");  
    }  
}
```

```

interface Interface {
    void method();
    default void newDefaultMethod() { System.out.println("newDefaultMethod() in Interface"); }
}

class SuperCls {
    public void newDefaultMethod() { System.out.println("newDefaultMethod() in SuperCls"); }
}

class Cls extends SuperCls implements Interface {
    @Override
    public void method() { System.out.println("method() in Cls"); }
}

public class InterfaceEx {
    public static void main(String[] args) {
        Cls obj = new Cls();
        obj.method();
        obj.newDefaultMethod();

        Interface obj2 = new Cls();
        obj2.method();
        obj2.newDefaultMethod();
    }
}

```

method() in Cls
 newDefaultMethod() in SuperCls
 method() in Cls
 newDefaultMethod() in SuperCls

```
interface Interface {  
    void method();  
    default void newDefaultMethod() { System.out.println("newDefaultMethod() in Interface"); }  
}
```

```
interface AnotherInterface {  
    void newDefaultMethod();  
}
```

```
class Cls implements Interface, AnotherInterface {  
    @Override  
    public void method() { System.out.println("method() in Cls"); }  
}
```

```
public class InterfaceEx {  
    public static void main(String[] args) {  
        Cls obj = new Cls();  
        obj.method();  
        obj.newDefaultMethod();  
  
        Interface obj2 = new Cls();  
        obj2.method();  
        obj2.newDefaultMethod();  
    }  
}
```

```
javac InterfaceEx.java  
InterfaceEx.java:10: error: Cls is not abstract and does not override abstract method newDefaultMethod() in AnotherInterface  
class Cls implements Interface, AnotherInterface {  
    ^  
1 error
```

```

interface Interface {
    void method();
    default void newDefaultMethod() { System.out.println("newDefaultMethod() in Interface"); }
}

interface AnotherInterface {
    void newDefaultMethod();
}

class Cls implements Interface, AnotherInterface {
    @Override
    public void method() { System.out.println("method() in Cls"); }
    @Override
    public void newDefaultMethod() { System.out.println("newDefaultMethod() in Cls"); }
}

public class InterfaceEx {
    public static void main(String[] args) {
        Cls obj = new Cls();
        obj.method();
        obj.newDefaultMethod();

        Interface obj2 = new Cls();
        obj2.method();
        obj2.newDefaultMethod();

        AnotherInterface obj3 = new Cls();
        //obj3.method();
        obj3.newDefaultMethod();
    }
}

```

```

method() in Cls
newDefaultMethod() in Cls
method() in Cls
newDefaultMethod() in Cls
newDefaultMethod() in Cls

```

```

interface Interface {
    void method();
    default void newDefaultMethod() { System.out.println("newDefaultMethod() in Interface"); }
}

interface AnotherInterface {
    default void newDefaultMethod() { System.out.println("newDefaultMethod() in AnotherInterface"); }
}

class Cls implements Interface, AnotherInterface {
    @Override
    public void method() { System.out.println("method() in Cls"); }
}

public class InterfaceEx {
    public static void main(String[] args) {
        Cls obj = new Cls();
        obj.method();
        obj.newDefaultMethod();

        Interface obj2 = new Cls();
        obj2.method();
        obj2.newDefaultMethod();

        AnotherInterface obj3 = new Cls();
        //obj3.method();
        obj3.newDefaultMethod();
    }
}

```

javac InterfaceEx.java

Exception in thread "main" java.lang.Error: Unresolved compilation problem:

Duplicate default methods named newDefaultMethod with the parameters () and () are inherited from the types AnotherInterface and Interface

```

interface Interface {
    void method();
    default void newDefaultMethod() { System.out.println("newDefaultMethod() in Interface"); }
}

interface AnotherInterface {
    default void newDefaultMethod() { System.out.println("newDefaultMethod() in AnotherInterface"); }
}

class Cls implements Interface, AnotherInterface {
    @Override
    public void method() { System.out.println("method() in Cls"); }
    @Override
    public void newDefaultMethod() { System.out.println("newDefaultMethod() in Cls"); }
}

public class InterfaceEx {
    public static void main(String[] args) {
        Cls obj = new Cls();
        obj.method();
        obj.newDefaultMethod();

        Interface obj2 = new Cls();
        obj2.method();
        obj2.newDefaultMethod();

        AnotherInterface obj3 = new Cls();
        //obj3.method();
        obj3.newDefaultMethod();
    }
}

```

```

method() in Cls
newDefaultMethod() in Cls
method() in Cls
newDefaultMethod() in Cls
newDefaultMethod() in Cls

```

```

interface Interface {
    void method();
    default void newDefaultMethod() { System.out.println("newDefaultMethod() in Interface"); }
}

interface AnotherInterface {
    default void newDefaultMethod() { System.out.println("newDefaultMethod() in AnotherInterface"); }
}

class Cls implements Interface, AnotherInterface {
    @Override
    public void method() { System.out.println("method() in Cls"); }
    @Override
    public void newDefaultMethod() {
        Interface.super.newDefaultMethod();
        AnotherInterface.super.newDefaultMethod();
        System.out.println("newDefaultMethod() in Cls");
    }
}

public class InterfaceEx {
    public static void main(String[] args) {
        Cls obj = new Cls();
        obj.method();
        obj.newDefaultMethod();

        Interface obj2 = new Cls();
        obj2.method();
        obj2.newDefaultMethod();

        AnotherInterface obj3 = new Cls();
        //obj3.method();
        obj3.newDefaultMethod();
    }
}

```

```

method() in Cls
newDefaultMethod() in Interface
newDefaultMethod() in AnotherInterface
newDefaultMethod() in Cls
method() in Cls
newDefaultMethod() in Interface
newDefaultMethod() in AnotherInterface
newDefaultMethod() in Cls
newDefaultMethod() in Interface
newDefaultMethod() in AnotherInterface
newDefaultMethod() in Cls

```

```

interface Interface {
    void method();
    default void newDefaultMethod() { System.out.println("newDefaultMethod() in Interface"); }
}

interface SubInterface extends Interface {
    default void newDefaultMethod() {
        Interface.super.newDefaultMethod();
        System.out.println("newDefaultMethod() in SubInterface");
    }
}

class Cls implements SubInterface {
    @Override
    public void method() { System.out.println("method() in Cls"); }
}

public class InterfaceEx {
    public static void main(String[] args) {
        Cls obj = new Cls();
        obj.method();
        obj.newDefaultMethod();

        Interface obj2 = new Cls();
        obj2.method();
        obj2.newDefaultMethod();

        SubInterface obj3 = new Cls();
        obj3.method();
        obj3.newDefaultMethod();
    }
}

```

method() in Cls
 newDefaultMethod() in Interface
 newDefaultMethod() in SubInterface
 method() in Cls
 newDefaultMethod() in Interface
 newDefaultMethod() in SubInterface
 method() in Cls
 newDefaultMethod() in Interface
 newDefaultMethod() in SubInterface


```

interface Interface {
    void method();
    default void newDefaultMethod() { System.out.println("newDefaultMethod() in Interface"); }
}

abstract class AbstractClass {
    abstract void newDefaultMethod();
}

class Cls extends AbstractClass implements Interface {
    @Override
    public void method() { System.out.println("method() in Cls"); }

    @Override
    public void newDefaultMethod() { System.out.println("newDefaultMethod() in Cls"); }
}

public class InterfaceEx {
    public static void main(String[] args) {
        Cls obj = new Cls();
        obj.method();
        obj.newDefaultMethod();

        Interface obj2 = new Cls();
        obj2.method();
        obj2.newDefaultMethod();
    }
}

```

```

method() in Cls
newDefaultMethod() in Cls
method() in Cls
newDefaultMethod() in Cls

```

Static Methods

```
interface Interface {  
    void method();  
    default void newDefaultMethod() { System.out.println("newDefaultMethod() in Interface"); }  
    static void staticMethod() { System.out.println("staticMethod() in Interface"); }  
}  
  
class Cls implements Interface {  
    @Override  
    public void method() { System.out.println("method() in Cls"); }  
}  
  
public class InterfaceEx {  
    public static void main(String[] args) {  
        Interface obj = new Cls();  
        //obj.staticMethod(); // compile-time error  
        Interface.staticMethod();  
  
        Cls obj2 = new Cls();  
        //obj2.staticMethod(); // compile-time error  
        Interface.staticMethod();  
    }  
}
```

staticMethod() in Interface
staticMethod() in Interface

Java 9 Interface Changes

- Constant variables
- Abstract methods
- Default methods
- Static methods
- Private methods
- Private Static methods

Private Methods

```
interface Interface {  
    void method();  
    //private void privateAbstractMethod();  
    //private default void privateDefaultMethod() {}  
  
    default void defaultMethod1() {  
        privateMethod1();  
    }  
  
    default void defaultMethod2() {  
        System.out.println("Default Method in Interface");  
    }  
  
    private void privateMethod1() {  
        System.out.println("Private Method1 in Interface");  
  
        privateMethod2();  
        defaultMethod2();  
        method();  
    }  
  
    private void privateMethod2() {  
        System.out.println("Private Method2 in Interface");  
    }  
}
```

Private Static Method

```
interface Interface {  
    //static void staticMethod();  
    //private static void privateAbstractStaticMethod();  
    //private static default void privateStaticDefaultMethod() {}  
  
    static void staticMethod() {  
        privateStaticMethod1();  
    }  
  
    private static void privateStaticMethod1() {  
        //method();  
        //defaultMethod();  
        //privateMethod();  
    }  
  
    private static void privateStaticMethod2() {  
        privateStaticMethod1();  
    }  
  
    void method();  
    default void defaultMethod() {}  
    private void privateMethod() {}  
}
```

Private Methods

```
interface Interface {  
    void method();  
    default void defaultMethod() {}  
    static void staticMethod() {}  
    private static void privateStaticMethod() {}  
  
    private void privateMethod() {  
        method();  
        defaultMethod();  
        staticMethod();  
        privateStaticMethod();  
    }  
}
```