

Java 9 Private Interface Methods

[< prev](#)[next >](#)

In Java 9, we can create private methods inside an interface. Interface allows us to declare private methods that help to **share** common code between **non-abstract** methods.

Before Java 9, creating private methods inside an interface cause a compile time error. The following example is compiled using Java 8 compiler and throws a compile time error.

Java 9 Private Interface Methods Example

```
interface Sayable{
    default void say() {
        saySomething();
    }
    // Private method inside interface
    private void saySomething() {
        System.out.println("Hello... I'm private method");
    }
}

public class PrivateInterface implements Sayable {
    public static void main(String[] args) {
        Sayable s = new PrivateInterface();
        s.say();
    }
}
```

Output:

```
PrivateInterface.java:6: error: modifier private not allowed here
```

 **Note:** To implement private interface methods, compile source code using Java 9 or higher versions only.

Now, lets execute the following code by using Java 9. See the output, it executes fine.

Java 9 Private Interface Methods Example

```
interface Sayable{
    default void say() {
        saySomething();
    }
    // Private method inside interface
    private void saySomething() {
        System.out.println("Hello... I'm private method");
    }
}

public class PrivateInterface implements Sayable {
    public static void main(String[] args) {
        Sayable s = new PrivateInterface();
        s.say();
    }
}
```

Output:

```
Hello... I'm private method
```

Such like, we can also create **private static methods** inside an interface. See, the following example.

Java 9 Private Static Methods Example

```
interface Sayable{
    default void say() {
        saySomething(); // Calling private method
        sayPolitely(); // Calling private static method
    }
    // Private method inside interface
    private void saySomething() {
        System.out.println("Hello... I'm private method");
    }
    // Private static method inside interface
    private static void sayPolitely() {
        System.out.println("I'm private static method");
    }
}

public class PrivateInterface implements Sayable {
    public static void main(String[] args) {
        Sayable s = new PrivateInterface();
        s.say();
    }
}
```

Output:

```
Hello... I'm private method
I'm private static method
```