Java / Q320

www.myjavainterview.in

HOW DO YOU WRITE AN INTERFACE WITH DEFAULT AND STATIC METHOD ?

Ravi Bisht
@backend.interview.preparation

Instagram

About

- In Java, you can write an interface with both default and static methods, which was introduced in Java 8

- Default methods allow you to define a body for a method inside an interface, which implementing classes inherit

- Static methods in an interface can be called without creating an instance of the interface

1. Default Method

- defaultMethod() has a default keyword and provides an implementation

- Any class implementing MyInterface will inherit this method

- Classes can also override it if needed

- A default method is used when you want to provide a method implementation that can be inherited by implementing classes without requiring them to override it

- This was introduced in Java 8 to help with backward compatibility

```
public interface MyInterface {
    default void defaultMethod() {
        println("This is a default method in the interface.")
    }
```

2. Static Method

- staticMethod() has a static keyword and provides an implementation

- You can call a static method directly on the interface without needing an instance.
    - MyInterface.staticMethod())

- it's not inherited by classes that implement the interface

```
public interface MyInterface {
    static void staticMethod() {
        println("This is a static method in the interface.")
    }
}
```

Full Example - Interface

Here's a complete example combining both a default and a static method within the same interface:

```
public interface MyInterface {

    // Default method
    default void defaultMethod() {
        println("This is a default method in the interface.");
    }

    // Static method
    static void staticMethod() {
        println("This is a static method in the interface.");
    }
}
```

Full Example - Default Method

Implementing classes can call the default method directly, and they can also override it if needed

```
public class MyClass implements MyInterface {
    public void show() {
        defaultMethod();
        // Calling the default method from the interface
    }
}

public class Test {
    public static void main(String[] args) {
        MyClass obj = new MyClass();
        obj.show();
        // This will print:
        // "This is a default method in the interface."
    }
}
```

Full Example - Static Method

You call the static method directly on the interface itself, not through an instance

```
public class Test {
    public static void main(String[] args) {
        MyInterface.staticMethod();
        // This will print:
        //"This is a static method in the interface."
    }
}
```

Key Points

- Default Methods can be overridden by implementing classes

- Static Methods in interfaces cannot be overridden, as they belong to the interface itself, not to instances of implementing classes

Using default and static methods allows to add functionality without breaking existing implementations