

Starting with Java 8, interfaces in Java can contain static methods. This feature enhances the capability of interfaces by allowing them to contain utility methods directly related to the interface without requiring an implementing class.

#### Key Points about Static Methods in Interfaces

**Utility Methods:** Static methods in interfaces are generally used to define utility methods. These methods are associated with the interface itself, rather than with any instance that implements the interface.

**Access:** Static methods in interfaces can only be accessed in a static way—using the interface name, not through an instance of an implementing class.

**No Overriding:** Static methods in interfaces are not inherited by classes that implement the interface.

**Implementation:** Unlike abstract methods in interfaces, static methods can have a full implementation.

#### Example of a Static Method in an Interface

Java

Collapse

```
1interface MathOperations {
2    // Static method in an interface
3    static int add(int a, int b) {
4        return a + b;
5    }
6
7    // Abstract method
8    double calculate(double a, double b);
9}
10
11class Calculator implements MathOperations {
12    @Override
13    public double calculate(double a, double b) {
14        return a * b; // Example implementation
15    }
16}
17
18public class StaticMethodInInterfaceExample {
19    public static void main(String[] args) {
20        // Accessing the static method from the interface
21        int sum = MathOperations.add(5, 10);
22        System.out.println("Sum: " + sum); // Output: Sum: 15
23
24        // Using the implementing class
25        Calculator calculator = new Calculator();
26        double product = calculator.calculate(5, 10);
27        System.out.println("Product: " + product); // Output: Product: 50.0
28    }
29}
```

#### Explanation

**Static Method:** `add(int a, int b)` is a static method within the `MathOperations` interface. It performs a utility operation that is directly related to mathematical operations.

**Accessing Static Method:** The static method is accessed using the interface name `MathOperations.add(5, 10)`. This demonstrates that static methods are associated with the interface itself.

**Implementation in Class:** The `Calculator` class implements the `MathOperations` interface, providing an implementation for the abstract method `calculate(double a, double b)`. However, it does not, and cannot, override the static method.

#### Interview Questions about Static Methods in Interfaces

What is a static method in an interface, and how do you use it in Java 8?

Can static methods in interfaces be overridden by implementing classes?

Why would you use a static method in an interface instead of a class?

Are static methods in interfaces inherited by implementing classes?

Benefits of Static Methods in Interfaces

**Encapsulation of Utility Methods:** Static methods in interfaces encapsulate utility functions that are closely related to the interface, making them easier to find and use.

**Code Organization:** They help in organizing helper or utility methods without cluttering implementing classes.

With static methods in interfaces, Java 8 introduced a flexible way to define reusable utility methods directly in the interfaces, enhancing the modularity and organization of code.