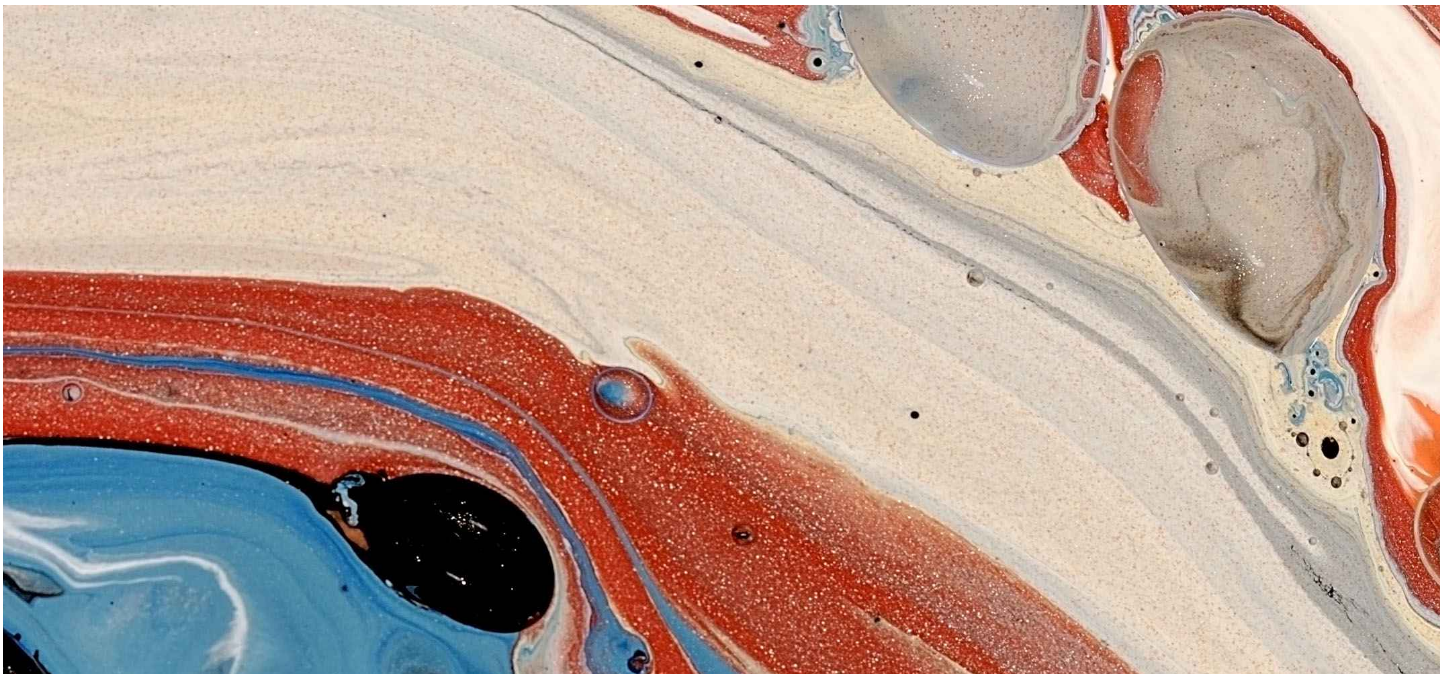

33-Java 8: Method Reference

Author: Vincent Lau

Learning Objectives

> Understand the basic syntax of using Method Reference
>
> Understand the relationship between Lambda and Method Reference in different scenario

Introduction

Method references provide a concise way to refer to methods or constructors without actually invoking them. They can be seen as shorthand notation for lambdas when the lambda's sole purpose is to call a method. Method references can improve code readability and reduce boilerplate when working with functional interfaces.

There are several types of method references:

Reference to Static Method

Method Reference

`ClassName::staticMethodName`

```java
List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5);
Consumer<Integer> sysout = System.out::println;
numbers.forEach(sysout);
```

Lambda Expression

```java
List<Integer> numbers = Arrays.asList(1, 2, 3, 4, 5);
numbers.forEach(number -> System.out.println(number));
```

Reference to an Instance Method of a Particular Object

Method Reference

`object::instanceMethodName`

```java
String name = "John";
Supplier<Integer> lengthSupplier = name::length;
int length = lengthSupplier.get();
```

Lambda Expression

```java
String name = "John";
Supplier<Integer> lengthSupplier = () -> name.length();
int length = lengthSupplier.get();
```

Reference to an Instance Method of an Arbitrary Object of a Particular Type

Method Reference

`ClassName::instanceMethodName`

```java
List<String> words = Arrays.asList("apple", "banana", "cherry");
Comparator<String> compareToIgnoreCase = String::compareToIgnoreCase;
words.sort(compareToIgnoreCase);
```

Lambda Expression

```java
List<String> words = Arrays.asList("apple", "banana", "cherry");
Comparator<String> compareToIgnoreCase = (s1, s2) ->
s1.compareToIgnoreCase(s2);
words.sort(compareToIgnoreCase);
```

Reference to a Constructor

Method Reference

`ClassName::new`

```java
Function<String, Integer> stringToInteger = Integer::valueOf;
Integer number = stringToInteger.apply("123");
```

Lambda Expression

```java
Function<String, Integer> stringToInteger = s -> Integer.valueOf(s);
Integer number = stringToInteger.apply("123");
```

Summary

Method references are a powerful tool for simplifying code that involves functional programming. They allow you to directly reference existing methods and constructors, making your code more concise and expressive.

In summary, method references are a key feature in Java 8 that work hand in hand with lambda expressions to provide a more functional programming style. They offer a convenient way to pass method references as arguments to functional interfaces and enhance the readability of your code.