# Java 9 改进的 Optional 类

Optional 类在 Java 8 中引入，Optional 类的引入很好的解决空指针异常。。在 java 9 中, 添加了三个方法来改进它的功能：

* stream()
* ifPresentOrElse()
* or()

### **stream() 方法**

**语法**

public Stream<T> stream()

stream 方法的作用就是将 Optional 转为一个 Stream，如果该 Optional 中包含值，那么就返回包含这个值的 Stream，否则返回一个空的 Stream（Stream.empty()）。

## 实例

import java.util.Arrays; import java.util.List; import java.util.Optional; import java.util.stream.Collectors; import java.util.stream.Stream; public class Tester { public static void main(String[] args) { List<Optional<String>> list = Arrays.asList ( Optional.empty(), Optional.of("A"), Optional.empty(), Optional.of("B")); //filter the list based to print non-empty values //if optional is non-empty, get the value in stream, otherwise return empty List<String> filteredList = list.stream() .flatMap(o -> o.isPresent() ? Stream.of(o.get()) : Stream.empty()) .collect(Collectors.toList()); //Optional::stream method will return a stream of either one //or zero element if data is present or not. List<String> filteredListJava9 = list.stream() .flatMap(Optional::stream) .collect(Collectors.toList()); System.out.println(filteredList); System.out.println(filteredListJava9); } }

执行输出结果为：

[A, B][A, B]

### **ifPresentOrElse() 方法**

**语法**

public void ifPresentOrElse(Consumer<? super T> action, Runnable emptyAction)

ifPresentOrElse 方法的改进就是有了 else，接受两个参数 Consumer 和 Runnable。

ifPresentOrElse 方法的用途是，如果一个 Optional 包含值，则对其包含的值调用函数 action，即 action.accept(value)，这与 ifPresent 一致；与 ifPresent 方法的区别在于，ifPresentOrElse 还有第二个参数 emptyAction —— 如果 Optional 不包含值，那么 ifPresentOrElse 便会调用 emptyAction，即 emptyAction.run()。

## 实例

import java.util.Optional; public class Tester { public static void main(String[] args) { Optional<Integer> optional = Optional.of(1); optional.ifPresentOrElse( x -> System.out.println("Value: " + x),() -> System.out.println("Not Present.")); optional = Optional.empty(); optional.ifPresentOrElse( x -> System.out.println("Value: " + x),() -> System.out.println("Not Present.")); } }

执行输出结果为：

Value: 1Not Present.

### **or() 方法**

**语法**

public Optional<T> or(Supplier<? extends Optional<? extends T>> supplier)

如果值存在，返回 Optional 指定的值，否则返回一个预设的值。

## 实例

import java.util.Optional; import java.util.function.Supplier; public class Tester { public static void main(String[] args) { Optional<String> optional1 = Optional.of("Mahesh"); Supplier<Optional<String>> supplierString = () -> Optional.of("Not Present"); optional1 = optional1.or( supplierString); optional1.ifPresent( x -> System.out.println("Value: " + x)); optional1 = Optional.empty(); optional1 = optional1.or( supplierString); optional1.ifPresent( x -> System.out.println("Value: " + x)); } }

执行输出结果为：

Value: MaheshValue: Not Present