# Exploring the Consumer Functional Interface in Java 8

## Introduction

In this tutorial, we will \*\*code and explore\*\* the \*\*Consumer Functional Interface\*\* introduced in \*\*Java 8\*\*. All new functional interfaces introduced in Java 8 are available in the \*\***java.util.function**\*\* package.

## Exploring the Consumer Functional Interface

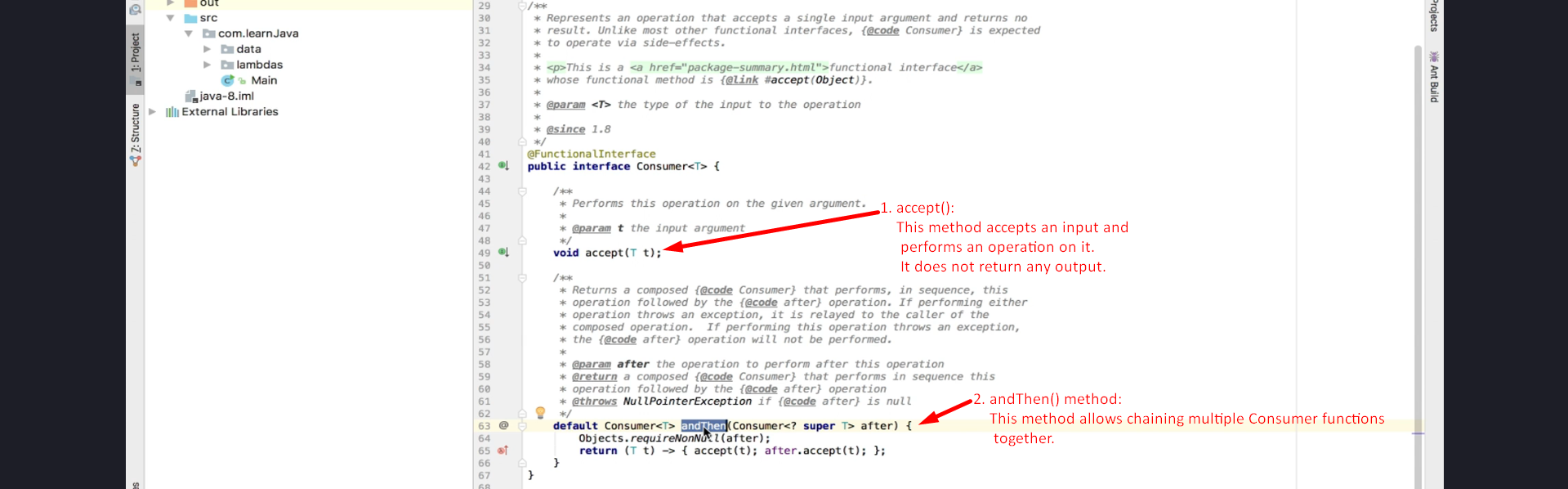
1. Open \*\*IntelliJ IDEA\*\* and press \*\*Shift\*\* twice to open the \*\*Search Everywhere\*\* window.

2. Search for \*\*Consumer Interface\*\*.

3. The \*\*Consumer interface\*\* has an \*\*'I' icon\*\*, indicating that it is an \*\*interface\*\*.

4. The \*\*Consumer interface\*\* is annotated with \*\*@FunctionalInterface\*\*, which means it contains exactly one abstract method.

## Methods in Consumer Interface

  
- We will explore more about this method in upcoming tutorials.

## Exploring the java.util.function Package

1. Click on the \*\*Consumer interface icon\*\* in IntelliJ.

2. This takes us to the \*\*source code\*\*, which is part of the \*\*rt.jar\*\* file.

3. The \*\*java.util.function\*\* package contains several functional interfaces:

- \*\*BiConsumer\*\*

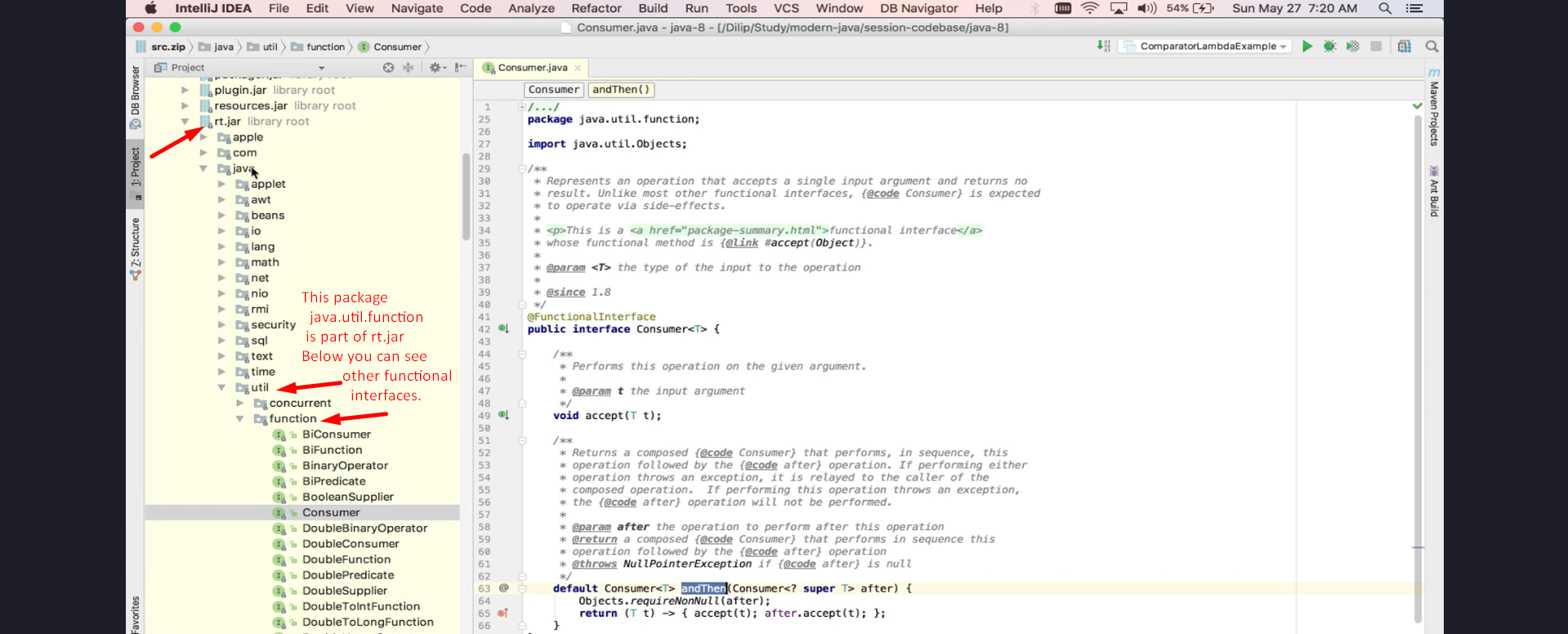
- \*\*BiPredicate\*\*

- \*\*BiFunction\*\*

- \*\*UnaryOperator\*\*

- \*\*BinaryOperator\*\*

4. All these interfaces were introduced in \*\*Java 8\*\* and are categorized as \*\*Functional Interfaces\*\*.



## Implementing the Consumer Functional Interface

1. Create a new package named \*\*functionalinterfaces\*\*.

2. Create a class named \*\*ConsumerExample\*\*.

3. Make the class executable by adding the \*\*public static void main()\*\* method.

### \*\*Code Implementation:\*\*

import java.util.function.Consumer;  
  
public class ConsumerExample {  
 public static void main(String[] args) {  
 // Creating Consumer using Lambda Expression  
 Consumer<String> c1 = (s) -> System.out.println(s.toUpperCase());  
  
 // Passing input using accept() method  
 c1.accept("Java8");  
 }  
}

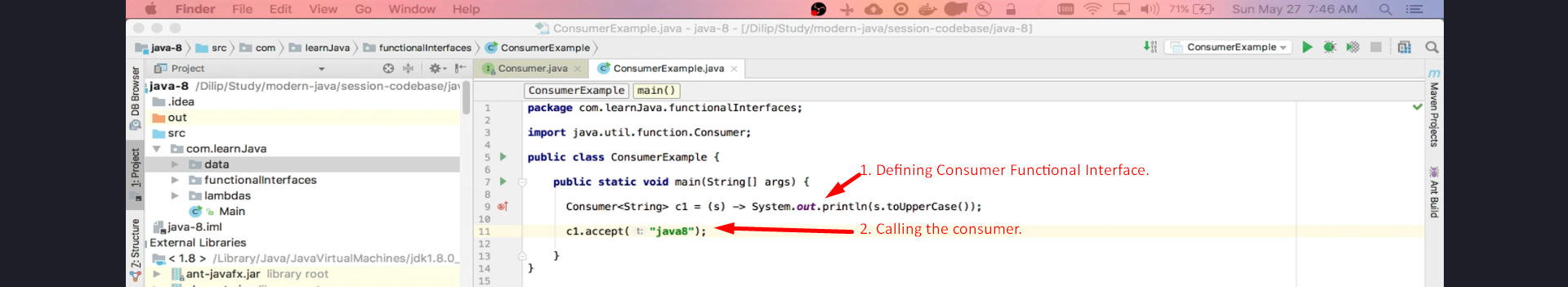
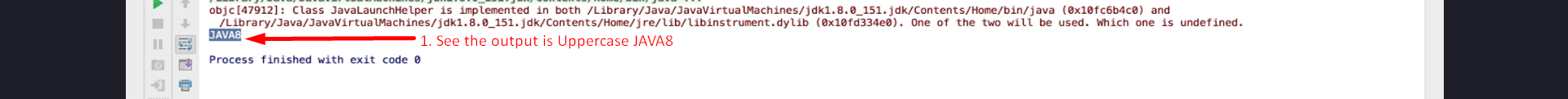
## Understanding the Code

1. \*\*Consumer<String> c1\*\* → Defines a \*\*Consumer functional interface\*\* that accepts a \*\*String input\*\*.

2. \*\*(s) -> System.out.println(s.toUpperCase())\*\* → The lambda function converts input to \*\*uppercase\*\* and prints it.

3. \*\*c1.accept("Java8")\*\* → Calls the `accept()` method and passes the string \*\*Java8\*\*.

4. \*\*Expected Output:\*\* `JAVA8`

  
**Output**:  


## Real-World Use Cases

In upcoming tutorials, we will explore \*\*real-world examples\*\* of the \*\*Consumer Functional Interface\*\*, demonstrating its practical applications.

## Conclusion

- The \*\*Consumer Functional Interface\*\* accepts an input and \*\*performs an operation without returning a result\*\*.

- It is part of the \*\*java.util.function\*\* package introduced in \*\*Java 8\*\*.

- We explored its key methods (`accept()` and `andThen()`).

- Upcoming tutorials will \*\*demonstrate real-world examples\*\* and additional functional interfaces.

### \*\*End of Tutorial\*\*

Thank you for watching!