Learning to Love the Lambda in the Stream

Introduction to Java 8 Lambda and Functional Interfaces

Speaker Introduction

- Richard Roda
- Sr. Technical Lead at DXC Technology
- Over 15 years of Java development experience
- OCA Java and Security+ certifications
- Linked In: https://www.linkedin.com/in/richardroda
- Twitter: @Richard_Roda
- These slides (pdf): https://tinyurl.com/love-lambda

What is a Lambda Expression?

- In Java, it is an unnamed function that may be bound to an interface as an object.
- Example 1

-
  ```
  Predicate<Integer> isFive = n -> n == 5;
  System.out.println(isFive.test(4)); // false
  ```
- Lambdas may only exist when assigned to a Functional Interface

-
  ```
  n -> n == 6; // Does not compile
  ```

Functional Interface (FI) in Java 8

- "A functional interface is any interface that contains only one abstract method." -- Oracle Java Tutorial
- Example 2- Valid Functional Interface

```
@FunctionalInterface // Optional
public interface Example2 {
    boolean equals(Object other); // In Object
    int hashCode(); // In Object
    int myMethod(); // Abstract.
    default int myMethod2() {return myMethod();}
    static int myMethod3() {return 0;}
}
```

Binding Lambda to Example2 FI vs Anonymous Inner class

- Both of these implement myMethod defined in Example2.
- Since there is only 1 abstract method, the lambda may omit specifiers required for method declarations.
- Method types and return values are inferred from the FI.

```
public class Example3 {
    static public void main(String[] args) {
        Example2 lambda = () -> 3; // 8 chars
        Example2 innerClass = new Example2() {
            @Override public int myMethod() {
                return 3;
            }
        }; // 5 lines of code
        System.out.println(lambda.myMethod()); // 3
        System.out.println(innerClass.myMethod()); // 3
    }
}
```

Functional Interface Conventions

- The following conventions apply for type variables used by Java 8 FIs:
- T – First argument
- U – Second argument
- R – Return Value
- Any of the above are omitted if not used.
- If an FI lacks an argument, T is used for the return value instead of R.