

The Java Predicate Functional Interface: Case Study ex8

Douglas C. Schmidt

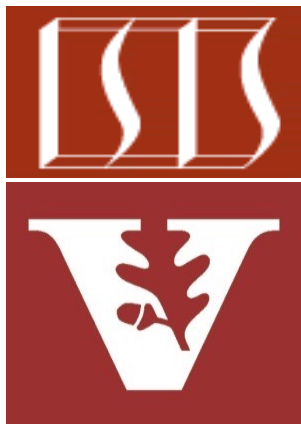
d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science

**Institute for Software
Integrated Systems**

**Vanderbilt University
Nashville, Tennessee, USA**



Learning Objectives in this Part of the Lesson

- Know how the Java Predicate functional interface can be used in conjunction with the Java HashMap removeIf() method

```
Map<String, Integer> stooges =  
    makeMap();
```

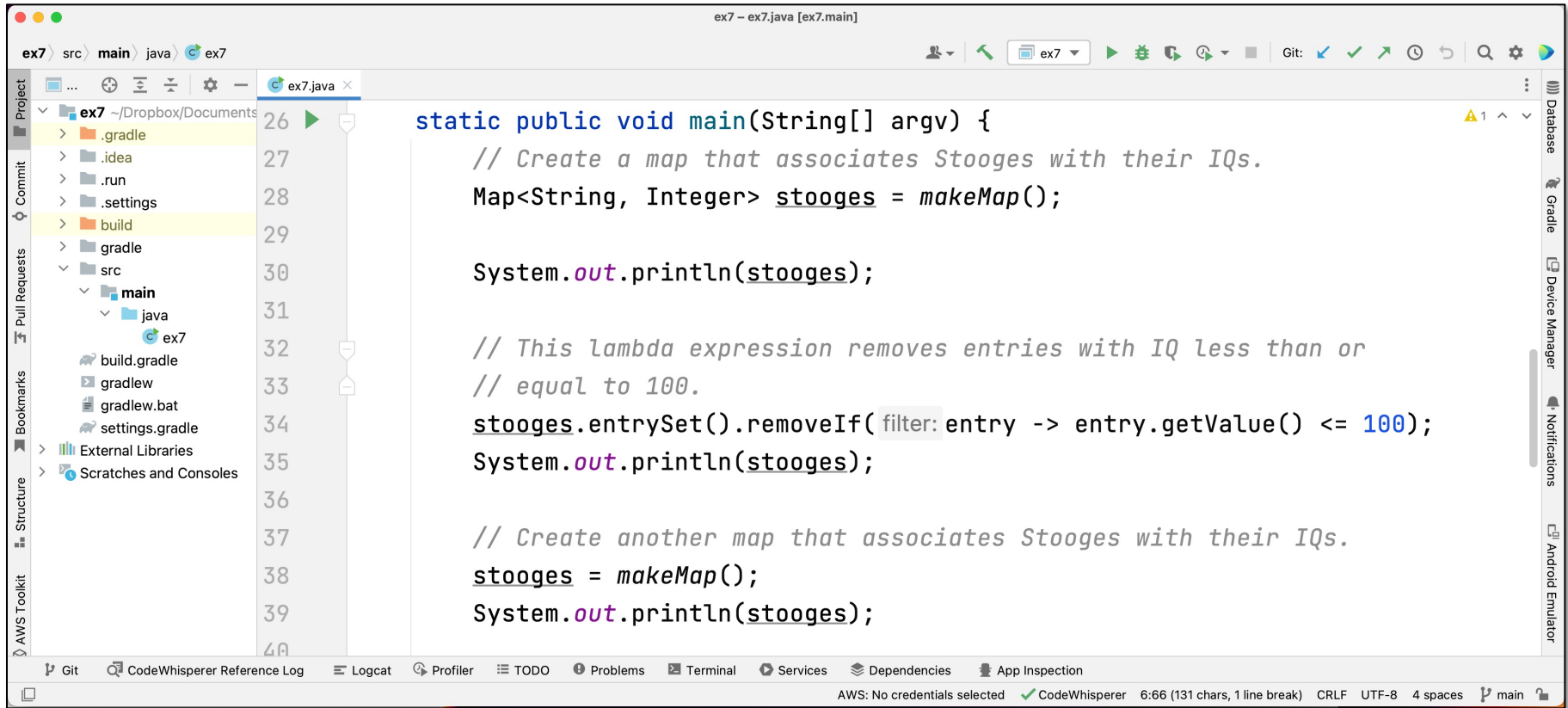
```
Predicate<Map.Entry<String,  
    Integer>> iq = entry ->  
    entry.getValue() <= 100;
```

```
Predicate<Map.Entry<String,  
    Integer>> curly = entry ->  
    entry.getKey().equals("Curly");
```

```
stooges.entrySet()  
    .removeIf(iq.and(curly));
```

Applying the Java Predicate Functional Interface

Applying the Java Predicate Functional Interface



The screenshot shows an IDE window titled "ex7 - ex7.java [ex7.main]". The left sidebar displays a project structure for "ex7" with folders like ".gradle", ".idea", ".run", ".settings", "build", "gradle", "src", "main", and "java". The main editor area shows the following Java code:

```
26 static public void main(String[] argv) {
27     // Create a map that associates Stooges with their IQs.
28     Map<String, Integer> stooges = makeMap();
29
30     System.out.println(stooges);
31
32     // This lambda expression removes entries with IQ less than or
33     // equal to 100.
34     stooges.entrySet().removeIf(filter: entry -> entry.getValue() <= 100);
35     System.out.println(stooges);
36
37     // Create another map that associates Stooges with their IQs.
38     stooges = makeMap();
39     System.out.println(stooges);
40 }
```

The bottom status bar indicates "AWS: No credentials selected", "CodeWhisperer", "6:66 (131 chars, 1 line break)", "CRLF", "UTF-8", "4 spaces", and "main".

See github.com/douglasraigschmidt/ModernJava/tree/main/FP/ex8

End of the Java Predicate Functional Interface: Case Study ex8