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

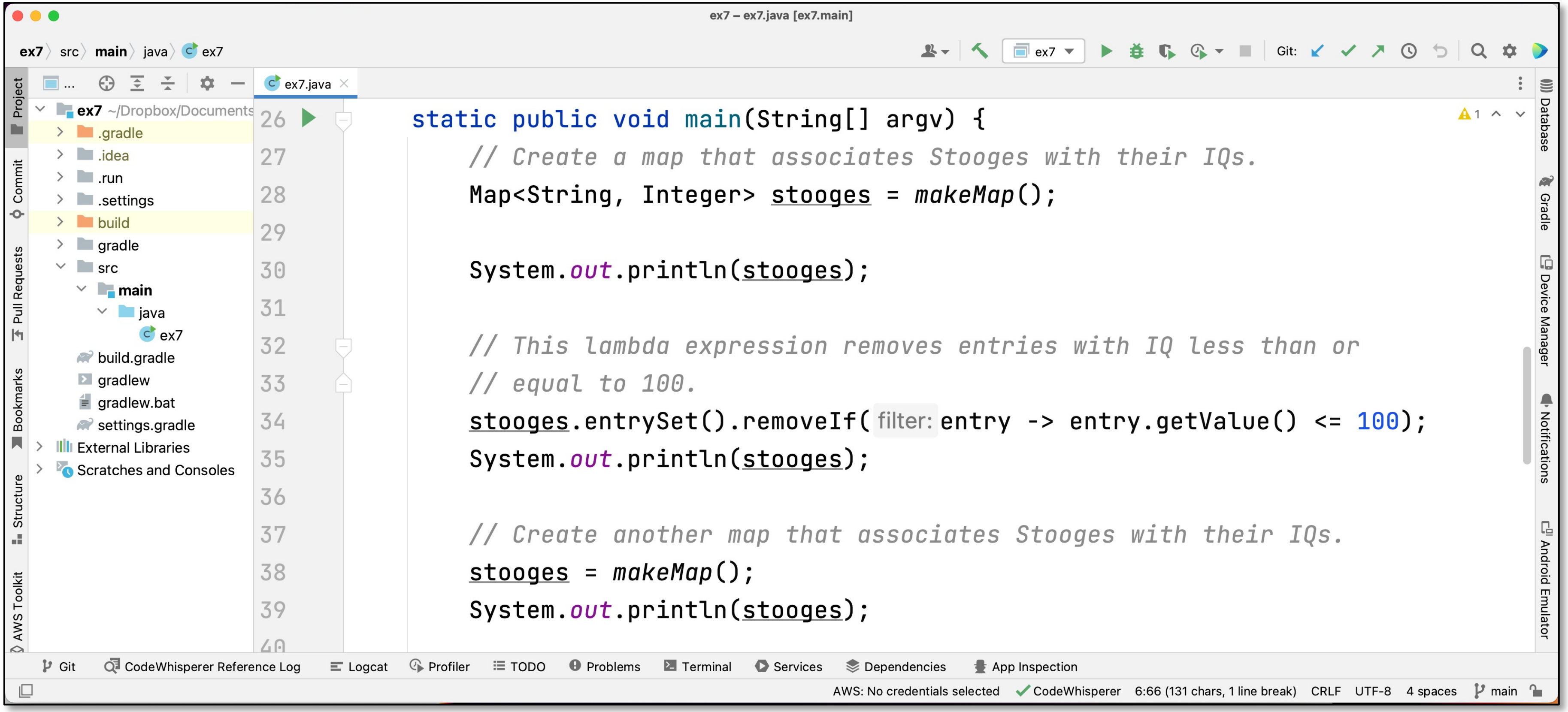
- Know how the Java Predicate functional interface can be used in conjunction with the Java HashMap removeIf() method

```
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));
```

See docs.oracle.com/javase/8/docs/api/java/util/function/Predicate.html

Applying the Java Predicate Functional Interface

Applying the Java Predicate Functional Interface



```java
static public void main(String[] argv) {
    // Create a map that associates Stooges with their IQs.
    Map<String, Integer> stooges = makeMap();

    System.out.println(stooges);

    // This lambda expression removes entries with IQ less than or
    // equal to 100.
    stooges.entrySet().removeIf(filter: entry -> entry.getValue() <= 100);
    System.out.println(stooges);

    // Create another map that associates Stooges with their IQs.
    stooges = makeMap();
    System.out.println(stooges);
```

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

End of the Java Predicate Functional Interface: Case Study ex8