The Java Function & Consumer Functional Interfaces: Case Study ex14

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

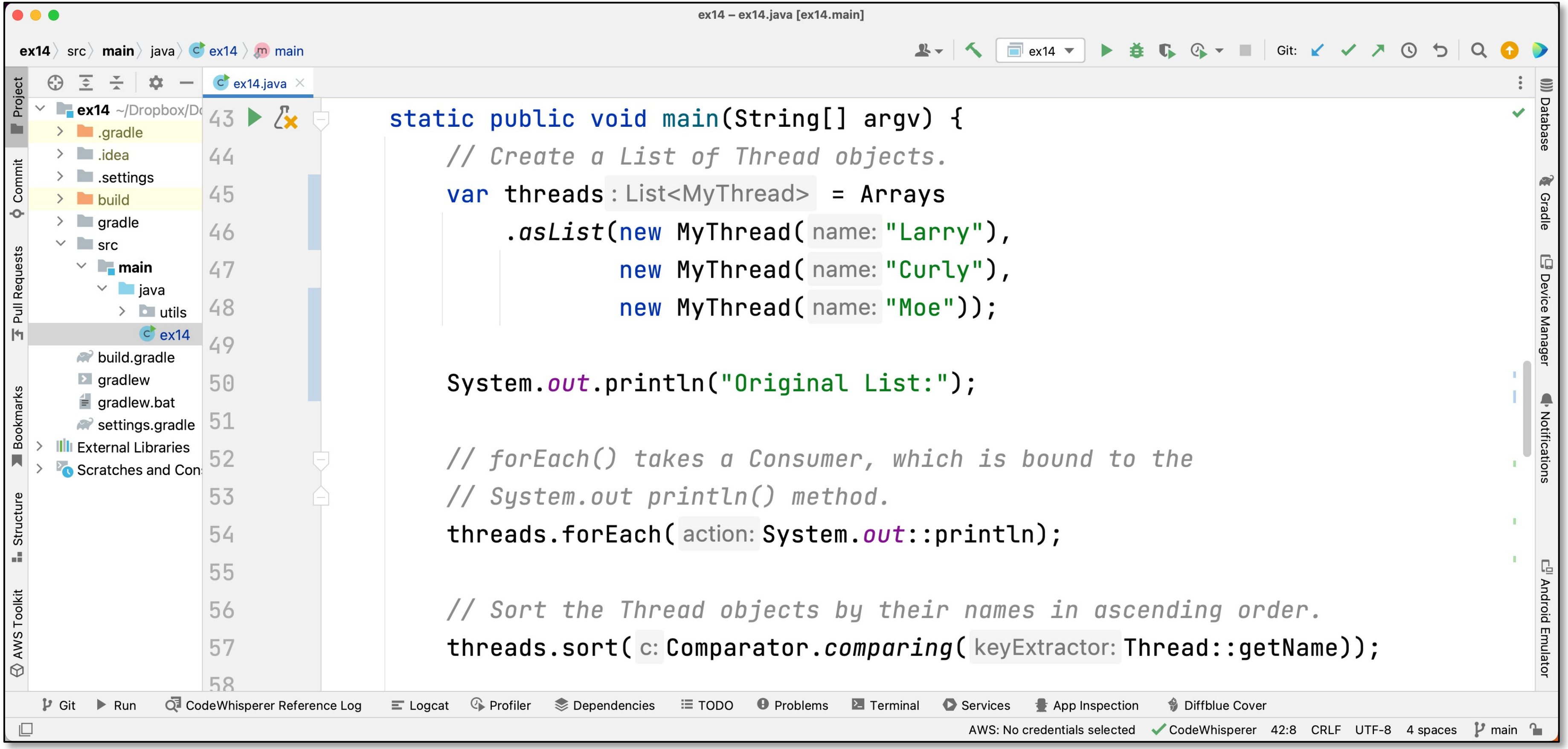
Learning Objectives in this Part of the Lesson

- Know how the Java Consumer & Function functional interfaces can be used to print & sort Thread objects by their names

```
var threads = Arrays
   .asList(new MyThread("Larry"),
           new MyThread("Curly"),
           new MyThread("Moe"));

threads.forEach(System.out::println);

threads.sort(Comparator.comparing
                  (Thread::getName));

threads.forEach(System.out::println);
...
```

Applying the Java Consumer & Function Functional Interfaces

Applying the Java Consumer & Function Interfaces



```
static public void main(String[] argv) {
    // Create a List of Thread objects.
    var threads : List<MyThread> = Arrays
        .asList(new MyThread( name: "Larry"),
                new MyThread( name: "Curly"),
                new MyThread( name: "Moe"));

    System.out.println("Original List:");

    // forEach() takes a Consumer, which is bound to the
    // System.out println() method.
    threads.forEach( action: System.out::println);

    // Sort the Thread objects by their names in ascending order.
    threads.sort( c: Comparator.comparing( keyExtractor: Thread::getName));
```

See github.com/douglascraigschmidt/ModernJava/tree/main/FP/ex14

End of the Java Consumer & Function Functional Interfaces: Case Study ex14