

The FileCount Case Study:

FileCounterSequentialStreamTask

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

- Understand the design of the FileCounter case study
- Walkthrough the program implementation
 - Main driver & associated helper classes
 - FileCounterForkJoinTask
 - FileCounterSequentialStreamTask
 - Applies Java sequential streams & the Java fork-join framework

```
var forks = Stream
    .of(mFile.listFiles())

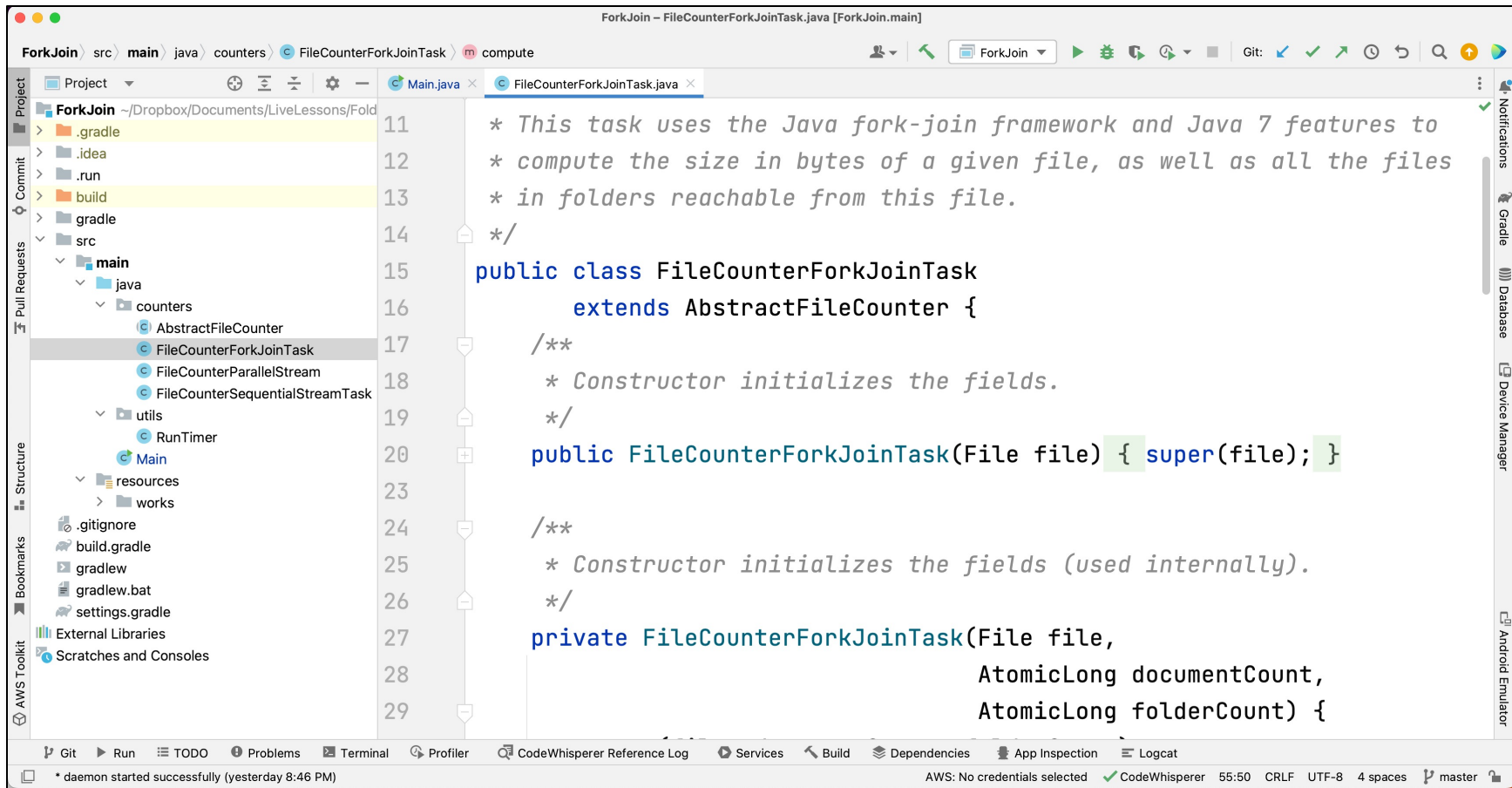
    .map(new
        FileCounterSequentialStreamTask
            (file,
             mDocumentCount,
             mFolderCount) .fork());

    .toList();

return forks
    .stream()
    .mapToLong(ForkJoinTask::join)
    .sum();
```

Walkthrough of the FileCounterSequential StreamTask

Walkthrough of the FileCounterSequentialStreamTask



The screenshot shows an IDE window titled "ForkJoin - FileCounterForkJoinTask.java [ForkJoin.main]". The left sidebar displays a project structure for "ForkJoin" with a tree view showing folders like ".gradle", ".idea", ".run", "build", "gradle", "src", "main", "utils", "resources", and "works", along with files like ".gitignore", "build.gradle", "gradlew", "gradlew.bat", "settings.gradle", "External Libraries", and "Scratches and Consoles". The main editor area shows the code for "FileCounterForkJoinTask.java" with line numbers 11 through 29. The code includes a class comment, a class declaration, a constructor, and a private constructor.

```
11  * This task uses the Java fork-join framework and Java 7 features to
12  * compute the size in bytes of a given file, as well as all the files
13  * in folders reachable from this file.
14  */
15  public class FileCounterForkJoinTask
16      extends AbstractFileCounter {
17      /**
18       * Constructor initializes the fields.
19       */
20      public FileCounterForkJoinTask(File file) { super(file); }
21
22
23
24      /**
25       * Constructor initializes the fields (used internally).
26       */
27      private FileCounterForkJoinTask(File file,
28                                     AtomicLong documentCount,
29                                     AtomicLong folderCount) {
```

See [Folders/ForkJoin/src/main/java/counters/FileCounterSequentialStreamTask.java](#)

End of the FileCount Case Study: FileCounter SequentialStreamTask