

Walkthrough of the Java StructuredTaskScope Code

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 Java's structured concurrency model
- Recognize the classes used to program Java's structure concurrency model
- Evaluate the design & performance of various Java concurrency models
- Learn how StructuredTaskScope is implemented
 - Focus on StructuredTaskScope

```
@PreviewFeature(feature = PreviewFeature.Feature.STRUCTURED_CONCURRENCY)
public class StructuredTaskScope<T>
    implements AutoCloseable {
    private final ThreadFactory factory;
    private final ThreadFlock flock;
    private final ReentrantLock shutdownLock = new ReentrantLock();

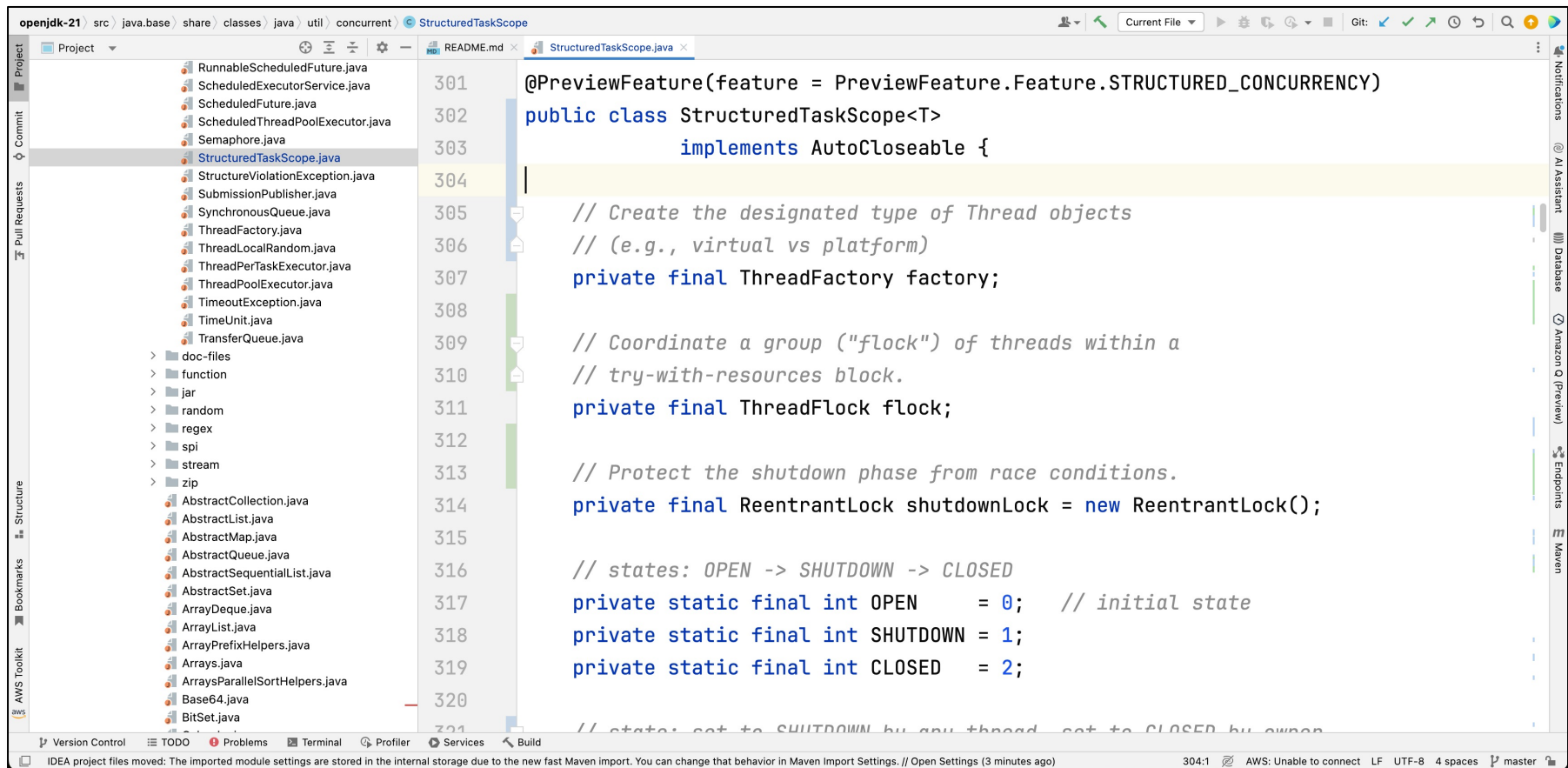
    // states: OPEN -> SHUTDOWN -> CLOSED
    private static final int OPEN      = 0;    // initial state
    private static final int SHUTDOWN  = 1;
    private static final int CLOSED    = 2;

    // state: set to SHUTDOWN by any thread, set to CLOSED by owner,
    // read by any thread
    private volatile int state;

    // Counters to support checking that the task scope owner joins
    // before processing results and attempts join before closing the
    // task scope. These counters are accessed only by the owner
    // thread.
    private int forkRound;           // incremented when the first
    //                                // subtask is forked after join
    private int lastJoinAttempted;  // set to the current fork round
    //                                // when join is attempted
    private int lastJoinCompleted;  // set to the current fork round
    //                                // when join completes
```

Walkthrough of the Java StructuredTaskScope Code

Walkthrough of the Java StructuredTaskScope Code



The screenshot shows an IDE window for the file `StructuredTaskScope.java`. The code is as follows:

```
301 @PreviewFeature(feature = PreviewFeature.Feature.STRUCTURED_CONCURRENCY)
302 public class StructuredTaskScope<T>
303     implements AutoCloseable {
304     |
305     // Create the designated type of Thread objects
306     // (e.g., virtual vs platform)
307     private final ThreadFactory factory;
308
309     // Coordinate a group ("flock") of threads within a
310     // try-with-resources block.
311     private final ThreadFlock flock;
312
313     // Protect the shutdown phase from race conditions.
314     private final ReentrantLock shutdownLock = new ReentrantLock();
315
316     // states: OPEN -> SHUTDOWN -> CLOSED
317     private static final int OPEN      = 0; // initial state
318     private static final int SHUTDOWN = 1;
319     private static final int CLOSED   = 2;
320
321     // states: set to SHUTDOWN by any thread, set to CLOSED by owner
```

The IDE interface includes a project tree on the left, a toolbar at the top, and a status bar at the bottom. The status bar shows the current file is at line 304:1, with a warning about AWS connection, and indicates 4 spaces and master branch.

See github.com/openjdk/jdk

End of Walkthrough of Java
StructuredTaskScope Code