Concurrency without Pain
Using Threads and Virtual Threads of Java

Duration: 4 days

Programming Concurrency is one of the most difficult task for programmers. The traditional concurrency libraries are mired with accidental complexities. Creating applications that are correct and high performant can soon turn into a daunting task if we're not careful. In this course, you'll will learn how to safely create high performant concurrent code using the traditional APIs and also learn the power and benefits of the more recent development in Java, specifically structured concurrency with Project Loom.

The course has a good balance of interactive lectures and hands-on exercises. The attendees are expected to pair-up and work on the lab exercises. The instructor will assist the attendees as they work on the labs. The objective of the course is for the attendees to gain an in depth practical knowledge of the concepts so they can put them to immediate use on their real projects.

The course content will be customized to meet your teams' specific needs. Please review this detailed outline and suggest changes (additions, deletions, modifications) as you feel fit.

Topics

Concurrent Programming
* Creating threads
* Main reasons to rely on multithreading
* Issues with working with threads
* Exercises

Thread Safety
* Threads and race conditions
* Java Memory Model
* Happens-Before
* Synchronization primitives
* Java 5 libraries
* Java 7 and Java 8 facilities
* Exercises

Making use of Threads
* Divide and conquer solutions
* Deciding number of threads
* Computation Intensive Operations
* I/O intensive Operations

* Measuring speedup
* Exercises

Concurrency using Parallel Streams

* When to use Parallel Streams vs. Executor services
* Functional Operations and threading
* Deciding if parallelization is the right choice
* Default pool size
* Configuring pool size
* Programmatically controlling pool size
* Measuring speedup
* Exercises

Rethinking Multithreading

* Mutability and threading
* Dealing with Shared Mutability
* Isolated Mutability
* Using Persistent data structures
* Exercises

Working Safely with Shared Mutability

* Dealing with Shared Mutability
* Preserving invariants
* Ensuring visibility
* Atomicity and concurrency
* Facilitating greater throughput
* Exercises

Virtual Threads and Project Loom

* Problems with Threads
* Asynchronous programming and Non blocking operations
* Separating tasks from threads
* Creating and using virtual threads
* Benefits of virtual threads
* Using virtual threads to perform IO and Microservices calls
* When to use Virtual Threads
* When not to use Virtual Threads
* Exercises

Structured Concurrency

* Coroutines and continuations
* Managing the lifetime of operations
* Parent vs. child tasks
* Coordinating between tasks
* Dealing with exceptions gracefully
* Designing with virtual streams and structure concurrency
* Exercises

Automated Testing Concurrent Code
* How to verify correctness of concurrent code
* Test first approach
* Automated testing for thread safety
* Designing for testability
* Exercises

About the Instructor

Dr. Venkat Subramaniam is an award-winning author, founder of Agile Developer, Inc., creator of agilelearner.com, and an instructional professor at the University of Houston.

He has trained and mentored thousands of software developers in the US, Canada, Europe, and Asia, and is a regularly-invited speaker at several international conferences. Venkat helps his clients effectively apply and succeed with sustainable agile practices on their software projects.

Venkat is a (co)author of multiple technical books, including the 2007 Jolt Productivity award winning book Practices of an Agile Developer. You can find a list of his books at agiledeveloper.com. You may read more about Venkat and Agile Developer, Inc. at http://agiledeveloper.com.