PROGRAMME

Durée : 4 jours soit 21h

Introduction

- Welcome to the course
- How we deal with questions
- Exercises with partial solutions
- Certificate of Training

History of Concurrency

- Benefits of threads
- Risks of threads
- Threads are everywhere
- Short Java 7 Primer

Thread Safety

- What is thread safety?
- Atomicity
- Locking
- Guarding state with locks
- Liveness and performance

Sharing Objects

- Visibility
- Publication and escape
- Thread confinement
- Immutability
- Safe publication

Composing Objects

- Designing a thread-safe class
- Instance confinement
- Delegating thread safety
- Adding functionality to existing thread-safe classes
- Documenting synchronization policies

Building Blocks

- Synchronized containers
- Concurrent containers
- Blocking queues and the producer-consumer pattern
- Blocking and interruptible methods
- Synchronizers
- Building an efficient, scalable result cache
- Summary

Task Execution

Durée : 4 jours soit 21h

- Executing tasks in threads
- The Executor framework
- Finding exploitable parallelism

Cancellation and Shutdown

- Task cancellation
- Stopping a thread-based service
- Handling abnormal thread termination
- JVM shutdown

Applying Thread Pools

- Tasks and Execution Policies
- Sizing thread pools
- Configuring ThreadPoolExecutor
- Extending ThreadPoolExecutor
- Parallelizing recursive algorithms

SwingWorker and Fork/Join

- SwingWorker (Java 6)
- Fork/Join (Java 7)
- Liveness, Performance, and Testing

Avoiding Liveness Hazards

- Deadlock
- Avoiding and diagnosing deadlocks
- Avoiding and diagnosing deadlocks

Performance and Scalability

- Thinking about performance
- Amdahl's and Little's laws
- Costs introduced by threads
- Reducing lock contention
- Example: Comparing Map performance
- Reducing context switch overhead

Testing Concurrent Programs

- Testing for correctness
- Testing for performance
- Advanced Topics

Explicit Locks

- Lock and ReentrantLock
- Performance considerations
- Fairness
- Synchronized vs ReentrantLock

Durée : 4 jours soit 21h

- Read-write locks

Building Customs Synchronizers

- Managing state dependence
- Using condition queues
- Explicit condition objects
- AbstractQueuedSynchronizer (AQS)
- Summary

Atomic Variables and Nonblocking Synchronization

- Disadvantages of locking
- Hardware support for concurrency
- Atomic variable classes
- Nonblocking algorithms
- Summary

Conclusion

- Tips on where to learn more
- Thank you!