FORMATION JAVA CONCURRENCY SPECIALISTS

FORMATEURS : BEN EVANS ET MARTIJN VERBURG / DUREE : 4 JOURS (28 HEURES)

PROGRAMME

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

FORMATION JAVA CONCURRENCY SPECIALISTS

FORMATEURS : BEN EVANS ET MARTIJN VERBURG / DUREE : 4 JOURS (28 HEURES)

- Synchronizers
- Building an efficient, scalable result cache
- Summary

TASK EXECUTION

- 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

FORMATION JAVA CONCURRENCY SPECIALISTS

FORMATEURS : BEN EVANS ET MARTIJN VERBURG / DUREE : 4 JOURS (28 HEURES)

TESTING CONCURRENT PROGRAMS

- Testing for correctness
- Testing for performance
- Advanced Topics

EXPLICIT LOCKS

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