

Multithreading mit Java («JAVA11»)

Erweitern Sie Ihre Kenntnisse der Applikationsentwicklung mit der Programmiersprache Java, um vertieftes Wissen zu Themen wie Multithreading, Monitoren, Locks, Synchronisation, Threadsicherheit und Fork/Join-Framework zu erlangen.

Dauer: 2 Tage

Preis: 1'350.- zzgl. 8.1% MWST

Kursdokumente: Digicomp Kursunterlagen

Inhalt

1. Nebenläufigkeit
 - o Multitasking versus Multithreading
 - o Prozesse versus Threads
 - o Scheduling
 - o Modelle
2. Threads
 - o Zustände
 - o Enumeration `java.lang.Thread.State`
 - o Klasse `java.lang.Thread`
 - o User Threads versus Daemon Threads
 - o Prioritäten
 - o Klasse `java.lang.ThreadGroup`
3. Runnables
 - o Schnittstelle `java.lang.Runnable`
 - o Schnittstelle `java.util.concurrent.Executor`
 - o Schnittstelle `java.util.concurrent.ThreadFactory`
 - o Schnittstelle `java.util.concurrent.Callable`
4. Executor Services und Thread Pools
 - o Schnittstelle `java.util.concurrent.ExecutorService`
 - o Schnittstelle `java.util.concurrent.ScheduledExecutorService`
 - o Klasse `java.util.concurrent.AbstractExecutorService`
 - o Klasse `java.util.concurrent.ThreadPoolExecutor`
 - o Klasse `java.util.concurrent.ScheduledThreadPoolExecutor`
 - o Schnittstelle `java.util.concurrent.Future`
 - o Enumeration `java.util.concurrent.TimeUnit`
 - o Klasse `java.util.concurrent.Executors`
5. Atomare Zugriffe
 - o `volatile`
 - o Klasse `java.util.concurrent.atomic.AtomicBoolean`
 - o Klasse `java.util.concurrent.atomic.AtomicInteger`
 - o Klasse `java.util.concurrent.atomic.AtomicLong`
 - o Klasse `java.util.concurrent.atomic.AtomicReference`
 - o Klasse `java.util.concurrent.atomic.AtomicIntegerArray`
 - o Klasse `java.util.concurrent.atomic.AtomicLongArray`
 - o Klasse `java.util.concurrent.atomic.AtomicReferenceArray` Synchronisation
 - o Kollisionen
 - o Monitoren
 - o Klasse `java.util.concurrent.Semaphore`
6. Locks
 - o Schnittstelle `java.util.concurrent.locks.Lock`
 - o Klasse `java.util.concurrent.locks.ReentrantLock`

- Schnittstelle `java.util.concurrent.locks.ReadWriteLock`
- Klasse `java.util.concurrent.locks.ReentrantReadWriteLock`

7. Interaktion

- Warten und Benachrichtigen
- Schnittstelle `java.util.concurrent.locks.Condition`
- Klasse `java.util.concurrent.CyclicBarrier`
- Klasse `java.util.concurrent.CountDownLatch`
- Klasse `java.util.concurrent.Phaser`

8. Threadsicherheit

- Threadsichere Listen
- Threadsichere Mengen
- Threadsichere Schlangen
- Threadsichere Abbildungen

9. Kommunikation

- Pipes
- Klasse `java.io.PipedOutputStream`
- Klasse `java.io.PipedInputStream`
- Klasse `java.io.PipedWriter`
- Klasse `java.io.PipedReader`
- Klasse `java.util.concurrent.Exchanger`

10. Nebenwirkungen

- Deadlock
- Livelock
- CPU Starvation
- Lock Starvation

11. Fork / Join Framework

- Rekursion
- Klasse `java.util.concurrent.ForkJoinTask`
- Klasse `java.util.concurrent.RecursiveAction`
- Klasse `java.util.concurrent.RecursiveTask`
- Klasse `java.util.concurrent.ForkJoinPool`

Key Learnings

- Umgang mit den verschiedenen Typen im Umfeld der Nebenläufigkeit von Java

Zielpublikum

Programmierer, die vertiefte Kenntnisse zu den traditionellen Wegen des Multithreading wie auch dem Concurrency API von Java erwerben wollen

Weiterführende Kurse

- [Einführung in die Applikationsentwicklung mit Jakarta EE \(«JEE01»\)](#)
- [Java SE Application Developer Professional \(«JSEADP»\)](#)

Haben Sie Fragen oder möchten Sie einen Firmenkurs buchen?

Wir beraten Sie gerne unter 044 447 21 21 oder info@digicomp.ch. Detaillierte Infos zu den Terminen finden Sie unter www.digicomp.ch/weiterbildung-software-engineering/programmiersprachen/java-javascript/kurs-multithreading-mit-java