

1. Inhaltsübersicht

2. Motivation

Wo werden Threads verwendet? Bei welchen Problemen sind sie behilflich?

3. Exkurs geschichtlicher Hintergrund von parallelem Programmieren

4. Was ist ein Thread

5. Implementierung von Threads in Java (Programmbeispiel)

6. Probleme beim Verwenden von Threads

7. Was ist Thread Sicherheit?

8. Was ist Atomicity? (Einblick in die entsprechenden Java Klassen)

9. Was verursachen Race conditions? (anhand von einem Beispiel)

10. Locking

11. Zustände mit Locks absichern

12. Lebensdauer und Performance (bezogen auf Threads mit Locks)

13. Was sind Sharing Objects?

- > Sichtbarkeit (Visibility)

- > Was sind volatile variables?

- > Thread confinement

- > Immutability

14. concurrent Collections (besser als synchronisierte Collections)

- > ConcurrentHashMap

- > CopyOnWriteArrayList

- > Blocking queues

- > serielles Thread confinement

- > Blocking und interruptable methods

15. Synchronizers

- > Latches

- > FutureTasks

- > Semaphores

- > Barriers

Rot: Einführung + Allgemeines Verständnis

Blau: Thread Sicherheit - und Exception-Umgang

Pink: Umsetzung in der Praxis