

# Nebenläufige Programmierung in Java

Dr. Andreas Schroeder



Dieses Video behandelt

- Threads und Runnable
  - Definition und Start
  - Threads stoppen: interrupts



- Threads
  - leichtgewichtige Prozesse
  - In Java durch Klasse `java.lang.Thread`
  - Verhalten: `run()`-Methoden
- Thread ausführen
  - `Thread.start()`-Methode ausführen – nicht `Thread.run()` !
- Runnable
  - Interface mit `run()`-Methode
  - Parameter-Typ in vielen Methoden



- Threads können nicht gestoppt werden
  - `Thread.stop()` ist deprecated
- Threads können unterbrochen werden
  - `Thread.interrupt()` setzt das interruption-flag
  - Code muss selbst überprüfen ob er unterbrochen wurde
  - Aufräumarbeiten können bei interrupt vorgenommen werden
- `Thread.join()` wartet auf Beenden des Threads



- Blockierende Methoden werfen eine InterruptedException
  - z.B. von `Thread.sleep()`, `Thread.join()` und `Object.wait()`
  - Das interrupted-flag ist aber nicht gesetzt!
  - Interrupt-flag setzen: `Thread.currentThread().interrupt()`



## Was in diesem Video gezeigt wurde

- Threads und Runnable
  - Definition und Start
  - Threads stoppen: interrupts

<http://docs.oracle.com/javase/tutorial/essential/concurrency/threads.html>