

# Übungsblatt 1: Threads, Runnables und Thread-Zustände

Vorstellung in den Tutorien am 03.-04. Mai 2022

## Themen:

- Threads und Runnables
- Thread-Zustände

## Tutoriumsaufgaben

In diesem Tutorium wollen wir uns mit der Implementierung von Threads beschäftigen. Dazu wollen wir zunächst zwei Schleifen mittels der Java-Klasse Thread und dem Java-Interface Runnable parallelisieren. Außerdem wollen wir die Grenzen der parallelen Ausführung austesten und überprüfen, inwiefern das Amdahl'sche Gesetz korrekt ist.

### 1.1 Prozesse und Threads

Bevor wir mit der Implementierung von Threads beginnen, wollen wir ein paar theoretische Grundlagen wiederholen.

1. Grenzen sie Prozesse von Threads ab
2. Was ist die Prozessumgebung?
3. Welchen Inhalt hat ein Thread Kontrollblock (TCB)

### 1.2 Extends Thread

1. Erstellen Sie zwei for-Schleifen (A und B), die nacheinander ausgeführt werden und in jedem Durchlauf ihren Namen (A bzw. B) und die entsprechende Nummer des Durchlaufs ausgeben. **Optional:** Können Sie hier auch eine for-each-Schleife nutzen?
2. Parallelisieren Sie nun die beiden Schleifen, indem Sie diese in zwei Threads abarbeiten. Schreiben sie dafür eine Klasse, die von Thread erbt (extends Thread).
3. Frage: Können Sie auch die run() Methode aufrufen? Begründen Sie Ihre Antwort!

## 1.3 Implements Runnable

1. Schreiben Sie eine Klasse `Logger`, die eine Methode `log()` implementiert. Diese Methode bekommt zwei Parameter `id` und `number`, die auf der Konsole mit `System.out.println()` ausgegeben werden sollen.
2. Nutzen Sie nun die `Logger`-Klasse für die Ausgabe der zwei `for`-Schleifen aus **Aufgabe 2.1**, indem die von Ihnen erstellte Klasse aus **Aufgabe 2.2** von der `Logger`-Klasse erbt.  
**Frage:** Treten Probleme auf? Wenn ja, wieso?
3. Lösen Sie das Problem aus **Aufgabe 3.2** mit dem `Runnable`-Interface.

## 1.4 Thread-Zustände

1. Benennen Sie die Thread-Zustände
2. Benennen Sie die Java-Methoden mit denen man einen Thread in den jeweiligen Zustand versetzt (sofern eine Methode existiert).
3. Wenn keine Methode existiert: Von wem wird der Thread in den jeweiligen Zustand versetzt?

