

TECHNISCHE INFORMATIK 2

SEMAPHORE

- Definiert im Paket `java.util.concurrent`
- Operationen (Methoden):
 - `acquire()` → Semaphore nehmen
 - `release()` → Semaphore zurückgeben
 - `tryAcquire()` → Nichtblockierend falls nicht erfolgreich
- Zählende Semaphore
 - Vorbelegung (auch negativ) wird im Konstruktor mitgegeben
- Faire Semaphore
 - Vergabe nach First Come First Served

Aufgabe

- Erstellen Sie ein Threadklasse (als Runnable oder Thread), die einen Producer darstellt. Die Klasse produziert die Ausgabe „Thread <name> ist aktiv #“. <name> ist der Name des Thread, # die Anzahl der Ausführungen. Threads dieser Klasse sollen eine variable Zeit lang laufen (sleep nach der Ausgabe). Bei jedem Durchlauf muss der Thread ein Semaphor (WriteSem) nehmen und nach erfolgreichem Abschluss wieder zurückgeben. Bei Abschluss (nach jedem Schreiben) erzeugt er zusätzlich ein weiteres Semaphor (ReadSem).
- Erstellen Sie eine Klasse, die einen Consumer darstellt. Dieser Thread läuft als Dämon und wartet auf das Lese-Semaphor. Bei Empfang produziert er eine Ausgabe analog zum Producer.
- Instanziiieren Sie in einem Hauptprogramm, in welchem Sie die Semaphore anlegen und die Threads starten. Variieren Sie mit der Anzahl an Threads und der Vorbelegung der Semaphore.