

Übung: Zähl-Threads

Schritt 1

Erstellen Sie eine Klasse `CounterExt`, deren Objekte die Aufgabe haben, einen gemeinsamen Zähler (Klassenvariable) bis zu einer bestimmten Obergrenze (Festgesetzt über Parameter des Konstruktors) in verschiedenen Threads hoch zu zählen. Das Zählen wird natürlich in der Methode `run()` implementiert. Zählen Sie zusätzlich auch in dem Thread die Anzahl an Schleifenwiederholungen mit (Variable `myCount`). Jeder Thread soll einen Namen zur Identifikation erhalten, der ebenfalls über den Konstruktor gesetzt werden kann.

Starten Sie dann 3 Threads (zB Tick, Trick und Track) und teilen Sie die Zählarbeit somit auf.

Sobald der Zählwert überschritten wurde soll folgendes ausgegeben werden: ThreadName AnzahlDurchgänge GlobalerZählwert Dauer(ms)



Deklarieren Sie die Variable `myCount` mit dem Schlüsselwort `volatile`. Dies bewirkt dass die Variable immer direkt im Hauptspeicher gelesen/geschrieben wird und nicht über den CPU Cache.



Nutzen Sie für die Ermittlung der Dauer die Klassen `Instant` (`Instant.now()`) und `Duration` (`Duration.between(beginn, ende)`)



Sehen Sie sich das Ergebnis genau an. Eigentlich würden wir erwarten, dass die Summe der Schleifendurchläufe aller Threads dem Zielwert entspricht. Warum ist das nicht der Fall?

Schritt 2

Wandeln Sie Ihr Programm dahingehend ab, dass anstatt der Vererbung das Interface `Runnable` zum Einsatz kommt.

Schritt 3

Experimentieren Sie ein wenig mit den Threads:

- Was passiert bei sehr geringer Obergrenze (zB 50)?
- Was passiert bei sehr hoher Obergrenze?
- Erspart man sich in dem Beispiel Zeit durch die Parallelisierung?

Klassendiagramm Schritt 1, Schritt 2

