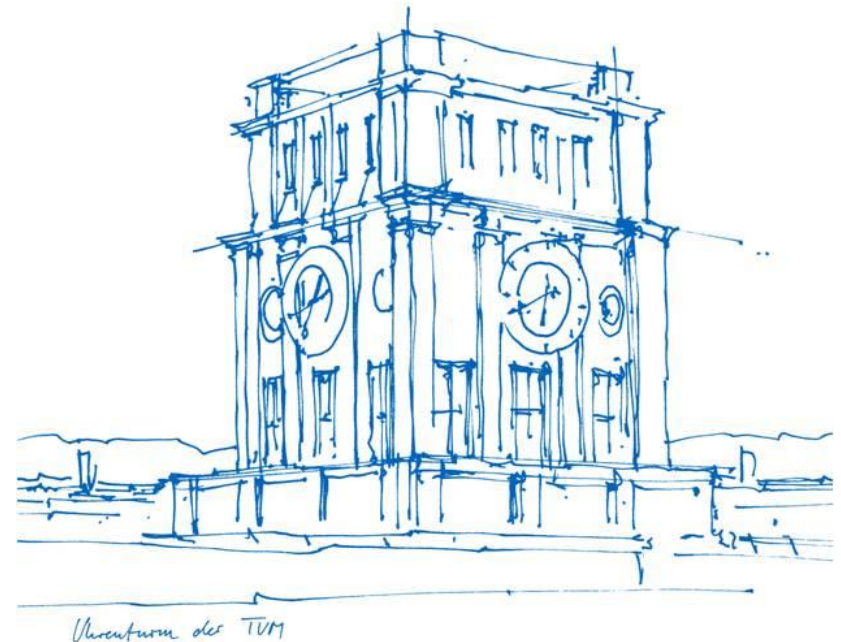


PGdP Woche #12 – Threads & parallele Streams

Sebastian Oßner – ossner@in.tum.de

Technische Universität München

Garching, 20. Januar 2020



Synchronisierte Threads

Problem letzter Woche:

Peter



balance += ...

balance += ...

Paul

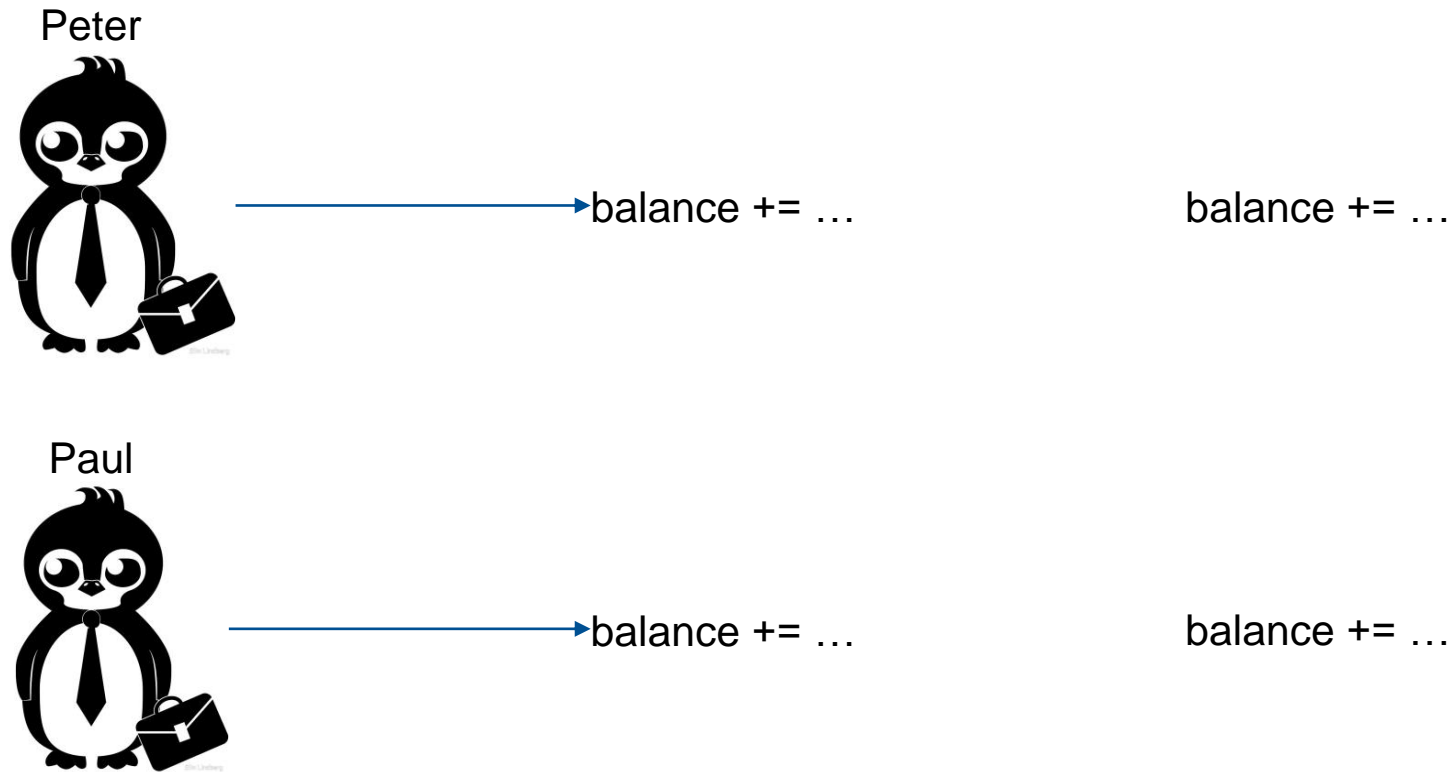


balance += ...

balance += ...

Synchronisierte Threads

Problem letzter Woche:



Synchronisierte Threads

Problem letzter Woche:



→ balance += ...



→ balance += ...

Durch diesen Parallelismus kann es passieren,
dass ein Thread schneller ist als der andere,
und die Synchronisation aus dem Schwung kommt

→ balance += ...

→ balance += ...

Synchronisierte Threads

Peter



balance += ...

balance += ...

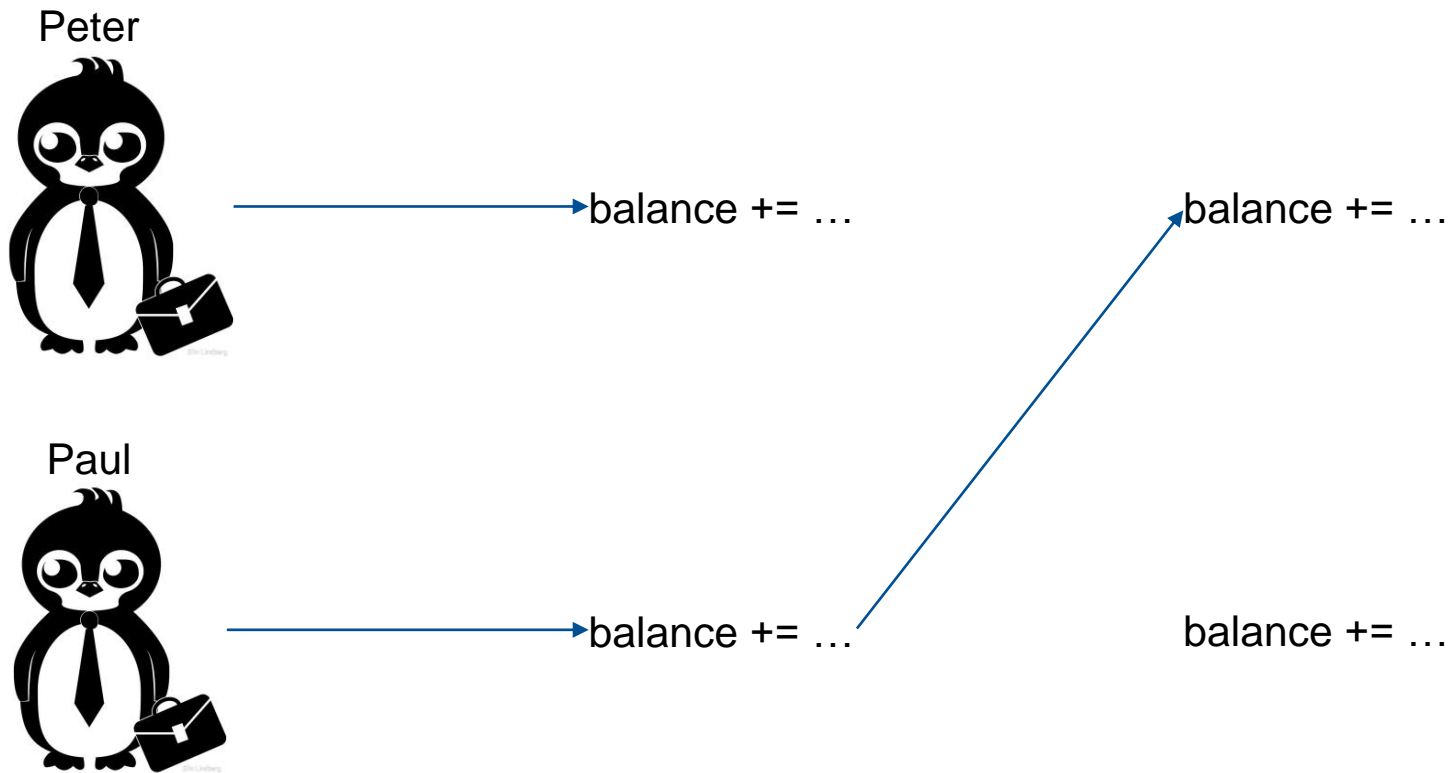
Paul



→ balance += ...

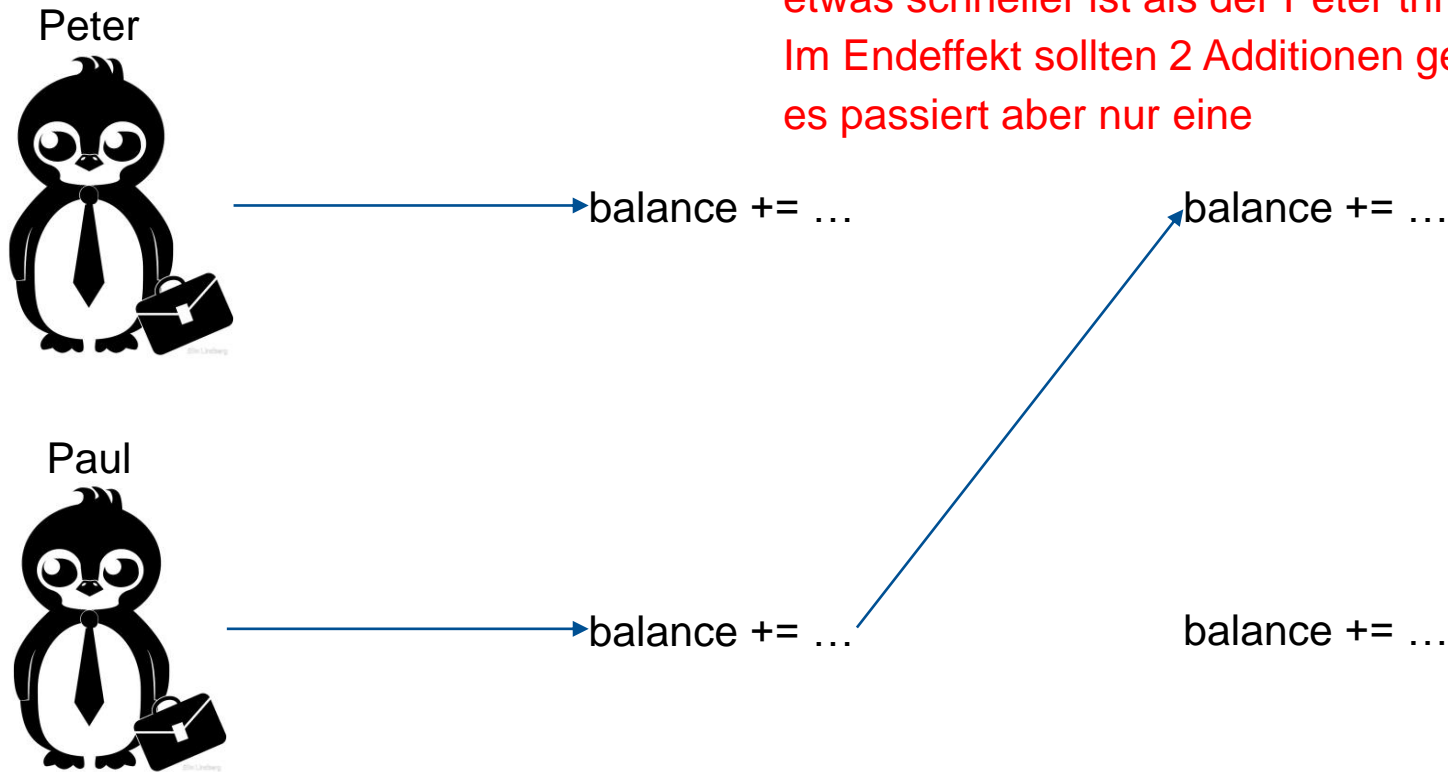
balance += ...

Synchronisierte Threads



Synchronisierte Threads

Hier will Peter gerade sein balance überschreiben, während Paul das selbe macht, da der Paul-Thread etwas schneller ist als der Peter thread. Im Endeffekt sollten 2 Additionen geschehen, es passiert aber nur eine



=> Race condition

P01 - Synchronisierte Threads

Ihr solltet nun dieses Problem lösen, setzt dazu das keyword “synchronized” ein.

Das Template enthält einen Lösungsvorschlag, testet diesen auf Korrektheit (er ist nicht korrekt)

Warum ist die Lösung falsch?

Implementiert dann eure eigene Lösung, und testet diese (Beide ausgegebenen Zahlen sollten 10.000 sein)

P02 – Synchronisierte Listen

- Das Template enthält einen Lösungsvorschlag, dieser funktioniert allerdings (mal wieder) nicht (Es treten Race Conditions auf)
- Race Conditions müssen also mit Synchronisation und Locks gelöst werden
- Verwendet dazu sowohl `synchronized` als auch Read/Write locks (RW.java)
 - Read/Write locks zählen die Anzahl der Reader/Writer und verwenden Thread Methoden wie `notify()` oder `wait()`

P03 – Parallele Streams

Lest euch die Aufgabe auf ArTeMiS durch und parallelisiert die Streams mit der Stream-Methode:

`.parallel()`

P04 – Klausurkorrektur (Ähnlich wie Klausuraufgaben)

- Semaphoren zur Synchronisation (siehe Vorlesung)
- Versteht die Aufgabenstellung genau auf Artemis und versucht die Aufgabe größtenteils Eigenständig zu lösen (Ohne Musterlösung)

