

Übung 4: JMM: PetersonMutex

Auf dem Netzlaufwerk finden Sie das Archiv 04_JMM.zip, welches im Package `as.peterson` eine Implementierung der Klasse `PetersonMutex` enthält. Der Algorithmus von Peterson ist eine vollständige Lösung des Problems des wechselseitigen Ausschlusses. Er garantiert, dass stets nur ein Thread in einen kritischen Abschnitt gelangen kann. Die Version, die sie auf dem Netz finden, kann jedoch nur verwendet werden um *zwei* Threads gegenseitig auszuschliessen.

Infos: http://de.wikipedia.org/wiki/Algorithmus_von_Peterson

Die bereitgestellte Implementierung ist jedoch nicht korrekt bezüglich Visibility (JMM), d.h. die in einem Thread geänderten Variablen sind im zweiten Thread nicht sichtbar. Wenn Sie das Programm `peterson.Main` ausführen, beobachten Sie mit grosser Wahrscheinlichkeit folgende Ausgabe:

```
>> 1 Thread-0  
<kein Fortschritt>
```

Aufgabe:

Korrigieren Sie dieses Programm so, dass es bezüglich des JMM korrekt ist, d.h. dass *alle* Daten, die im einen Thread geändert werden, im anderen Thread sichtbar sind.

Bemerkung:

- Wenn das Programm nach ihrer Korrektur korrekt läuft bedeutet dies noch nicht, dass es bezüglich JMM korrekt ist.
- Insbesondere genügt es nicht, nur das Feld `turn` als `volatile` zu deklarieren!
- `synchronized` ist nicht erlaubt