

# PVS

## Übungsblatt 12

Luca Krüger  
luca.krueger@uni-ulm.de

zum 11.07.2018

### Aufgabe 3

- a) In einer *synchronized* Methode werden Objekte, die in mehreren Methoden verändert werden können für alle anderen Methoden für Manipulationen blockiert. Diese Sperre hält solange an bis die Methode beendet ist. In einem *synchronized* Block wird das angegebene Objekt nach dem gleichen Prinzip vor Veränderungen von außen geschützt. Allerdings hält diese Blockade nur solange an, bis der Block beendet ist. Dies bietet sich vor allem bei langen Methoden an in denen nur wenige Zeilen für die Manipulation eines Objektes verantwortlich sind.
- b) Der Deadlock im gegebenen Beispielcode tritt zufällig zu Beginn des zweiten *synchronized* Blockes auf. Dies passiert im Thread *t2*.
- c) Die Methode *join()* führt dazu, dass auf jeden angefangenen *synchronized* Block gewartet wird, bis er terminiert.
- d) Es kommt zum Deadlock, wenn die *run()*-Methode in beiden Threads jeweils bis zu Beginn des zweiten *synchronized* Block abgelaufen ist. Durch die zueinander vertauschte Übergabe der Objekte in die beiden Thread-Klassen sind so beide Objekte durch einen nicht abgeschlossenen *synchronized* Block blockiert. Die Fortsetzung des inneren *synchronized* Blockes und damit Freigabe der Objekte in einer der Threads erfordert aber die Freigabe des jeweils anderen Objektes, das auf die selbe Weise blockiert wird.
- e) Um den Deadlock zu verhindern reicht es aus, die gleichen Parameter in der selben Reihenfolge an die Threads zu übergeben.