

Übungen zu Informatik B

Sommersemester 2018

Blatt 10

Dieses Blatt ist optional und dient zur zusätzlichen Übung. Es werden keine Testate stattfinden und die Musterlösung wird parallel veröffentlicht.

Aufgabe 10.1: Monitore und Locks (0 Punkte)

Betrachten Sie die Klassen `util.Queue`, `threads.Sleeper`, `threads.RandomGenerator` und `threads.MakeRandomSleeps`. Beim Ausführen der `main`-Methode von `threads.MakeRandomSleeps` kommt es zu zahlreichen Fehlern, da u.a. die verwendete Implementation des ADT Schlange `util.Queue` nur eine vorgegebene Anzahl von Objekten aufnehmen kann.

Beheben Sie die Probleme dadurch, dass Sie dafür sorgen das:

- beim Löschen von Elementen aus einer leeren Schlange solange gewartet wird, bis die Schlange nicht mehr leer ist.
- beim Einfügen von Elementen in eine volle Schlange solange gewartet wird, bis die Schlange nicht mehr voll ist.

Nutzen Sie dafür zwei verschiedene Ansätze, die sie unabhängig voneinander implementieren.

1. Verändern Sie nur die Klassen `threads.RandomGenerator` und `threads.Sleeper` und beheben Sie die Probleme durch die Synchronisation über ein geeignetes Monitor-Objekt.
2. Verändern Sie nur die Klasse `util.Queue` und beheben Sie die Probleme, indem Sie möglichst kleine Blöcke durch `java.util.concurrent.locks.Lock`-Instanzen sperren. Nutzen Sie eine oder mehrere `java.util.concurrent.locks.Condition`-Instanzen um die verlangte Synchronisation zu gewährleisten.

Aufgabe 10.2: Java Swing (0 Punkte)

Um mögliche Täuschungsversuche zu unterbinden, darf während einer Klausur nur jeweils eine Person die Herren- oder Damentoilette benutzen. Zur besseren Koordination wird ein Programm benötigt, mit dem für $n > 0$ Toiletten festgehalten werden kann, welche der Toiletten gerade besetzt ist und welche nicht.

Das Programm soll mit den jeweiligen Bezeichnern der Toiletten oder einer ganzen Zahl, die der Anzahl der überwachten Toiletten entspricht, aufgerufen werden. Als Anzeige dient ein einziges Fenster mit jeweils einem Element pro zu überwachender Toilette. Jedes dieser Elemente ist immer gleich angeordnet: Oben steht der Bezeichner der Toilette, darunter die Zeit seit dem die Toilette besetzt ist und darunter ein Button, mit dem zwischen den Zuständen *besetzt* und *frei* umgeschaltet werden kann. Ein Element, das eine freie Toilette repräsentiert, hat einen grünen Hintergrund. Ist die repräsentierte Toilette besetzt, soll der Hintergrund des Elements rot sein.

Implementieren Sie ein Programm mit Java-Swing und/oder AWT Oberfläche, die das geforderte Verhalten implementiert. Richten Sie sich beim Anordnen der Elemente nach Abbildung 1.

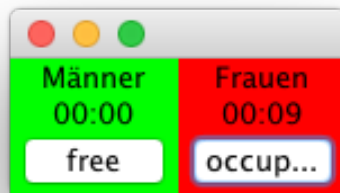


Abbildung 1: Programm zur Überwachung von zwei Toiletten. Eine davon ist seit 9 Sekunden besetzt und hat einen roten Hintergrund.