Übung 10 zur OOS SS16 Bearbeitung bis 02.06.2016

Aufgabe 36: (Thread-Verhalten)

a) Implementieren Sie eine Thread-Klasse `CountDown`, deren `run`-Methode ein Attribut bei 5 beginnend herunterzählt und dabei jeweils die Thread-Nummer und den Attributwert durch Doppelpunkt getrennt ausgibt. Hierzu soll diese Klasse ein statisches Attribut enthalten, das bei der Erzeugung eines neuen Threads dieser Klasse inkrementiert wird. Ein Thread dieser Klasse soll beendet werden, wenn der CountDown bei Null ist.

b) Implementieren Sie eine `main`-Methode für die Klasse aus a), die 5 Threads von der Klasse aus a) erzeugt, durchnummeriert und startet und überprüfen Sie, welche Ausgabe geliefert wird.

c) Ändert sich die Ausgabe, wenn Sie den CountDown bei 2000 beginnen?

d) Ändert sich die Ausgabe, wenn Sie vor der Ausgabe, den Thread eine MilliSekunde ruhen lassen?

e) Geben Sie dem i-ten erzeugten Thread die Priorität 2 * i. Ändert sich dann die Ausgabe?

Aufgabe 37: (Synchronisation)

Überprüfen Sie, ob das Programm auf Folie "Threads-6" auf Ihrem System alle Zahlen in der richtigen Reihenfolge ausgibt, falls die `for`-Schleife nur ein Hunderstel mal so oft durchlaufen wird..

Aufgabe 38: (join)

a) Implementieren Sie eine Thread-Klasse `Zähler`, deren `run`-Methode eine statische Variable solange inkrementiert, bis diese durch 50 teilbar ist.

b) Implementieren Sie eine Thread-Klasse `CntAusgabe`, deren `run`-Methode alle Vielfachen von 50 ausgibt, indem sie jeweils solange wartet, bis ein Thread vom Typ `Zähler` fertig ist und dann die statische Variable der Thread-Klasse `Zähler` ausgibt.

c) Implementieren Sie eine `main`-Methode, die unter Verwendung der Klassen aus a) und b) die 50er Reihe ausgibt.

Aufgabe 39: (mehrere Methoden synchronisiert)

Überprüfen Sie, wie sich das System verhält, wenn Sie mehrere Methoden der Klasse als `synchronized` deklarieren.

Übung 10 zur OOS SS16 Bearbeitung bis 02.06.2016

Design-Aufgabe 3: (Framework, Template-Method)

a) Entwerfen Sie ein Design in UML, so dass ein Framework implementiert wird, das eine Methode zur Verfügung stellt, die zwei Methoden von noch zu implementierenden Klassen hinter einander ausführt und zwischen deren Ausführung den String „Die erste ist jetzt fertig und ich mache mit der zweiten weiter." auf der Standardausgabe ausgibt.

b) Geben Sie eine Implementierung Ihres Designs aus a) an.