Prof. Dr. A. Weber
Dr. Hassan Errami

Universität Bonn
Institut für Informatik II
30.05.2019

Sommersemester 2019

Übungen zur Vorlesung
Objektorientierte Softwareentwicklung (BA-INF-024)
Aufgabenblatt 7 (20 Punkte)

Zu bearbeiten bis: 07.06.2019

Aufgabe 1 *(Threads - 2+2+1=5 Punkte)*

a) Nennen Sie die beiden Möglichkeiten, die Java anbietet um Threads zu erzeugen, und geben Sie für jede Methode einen Vorteil an.

b) Die Klasse `ThreadTest` sei vorgegeben:

```
public class ThreadTest {
  public static void main(String args[]){
    Thread t1 = new Thread(new DateCommand());
    t1.start();
    Thread t2 = new Thread(new CounterCommand());
    t2.start();
  }
}
```

Erstellen Sie die Klassen `DateCommand` und `CounterCommand` durch Implementieren des Interfaces `Runnable`. Der parallel auszuführende Programmcode soll in beiden Klassen lediglich aus einer `for`-Schleife bestehen, die zehnmal durchlaufen wird. In der Klasse `DateCommand` soll in jedem Schleifendurchlauf das aktuelle Datum ausgeben werden, in der Klasse `CounterCommand` lediglich der aktuelle Wert der Schleifenvariablen.

c) Benutzen Sie die Methode `sleep()` der Klasse `Thread`, um die Schleifendurchläufe zu verzögern. Nutzen Sie die Funktion `new java.util.Random().nextInt(1000)` um die Länge der Verzögerung zufällig zu wählen. Hinweis: Da eine `InterruptedEception` ausgelöst wird, wenn ein Thread unterbrochen wird, muss die Verzögerung in einem `try-catch`-Block untergebracht werden.[1] Führen Sie das Programm `ThreadTest` aus und erläutern Sie das Ergebnis!

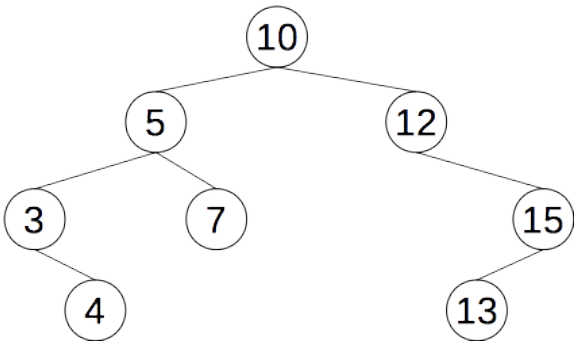
Aufgabe 2 *(Threads - 5 Punkte)*

Schreiben Sie eine Klasse *DateiBeobachter* mit Konstruktoren *DateiBeobachter(String)* und *DateiBeobachter(java.io.File)*, die das Interface *Runnable* implementiert. Die Klasse soll beobachten, ob sich eine Datei im Dateisystem ändert, und gegebenenfalls eine entsprechende Meldung ausgeben.[2] Lassen Sie sich während der Programmausführung im Abstand von einer halben Sekunde den Namen der Datei anzeigen. Legen Sie zum Testen eine Textdatei an und modifizieren Sie diese schließlich zur Laufzeit des Threads.

Aufgabe 3 *(Bäume - 2+2+3+2+1=10 Punkte)*

Gegeben sei folgender binärer Suchbau:

[1] Ausnahmebehandlungen können vernachlässigt werden.
[2] Benutzen Sie die Methode *java.io.File.lastModified()* um auf die Zeit der letzten Modifikation zuzugreifen.



a) Geben Sie die Reihenfolge der ausgegebenen Knoten an, bei folgenden Traversierungsreihenfolgen:

- Preorder
- Postorder
- Inorder

b) Schreiben Sie eine Datenstruktur für den oben abgebildeten Baum. Legen Sie hierzu eine Klasse TNode an, die einen Knoten des Baum repräsentiert. Schreiben Sie eine Weitere Klasse Baum, die den Wurzelknoten in einer Variable hält.
c) Implementieren Sie, in der Klasse Baum, die drei Traviersierungsverfahren Inorder, Preorder und Postorder. Diese sollen rekursiv aufgerufen werden und dabei die Reihenfolge der besuchten Knoten ausgeben.
d) Implementieren Sie eine Methode, die den Baum schichtweise ausgibt (Levelorder). Dabei soll mit der Wurzel begonnen werden. Nutzen Sie hierzu eine Warteschlange (java.util.LinkedList<E>).
e) Testen Sie Ihre Implementation, indem Sie die verschiedenen Methoden auf dem oben gegebenen Baum ausführen.