Übung 5

Algorithmen, Komplexität von Algorithmen

1. Betrachten Sie den folgenden Algorithmus in Pseudocode:

```
multi := 1;
FOR i := 1 TO 10 DO
  multi := multi * i;
```

(a) Welchen Wert hat die Variable `multi` nach der Schleife?

(b) Wie verhält sich der Algorithmus bezüglich

- i. Statischer Finitheit,
- ii. Dynamischer Finitheit,
- iii. Terminierung?

(c) Stellen Sie den Algorithmus als Flußdiagramm (bzw. alternativ als UML Aktivitätsdiagramm) dar.

2. Entwerfen Sie einen Algorithmus in Pseudocode, der mittels eines Schleifenkonstruktes eine Variable beginnend von 100 bei jedem Durchlauf um 10 erniedrigt. Ist der Algorithmus **statisch finit**, **dynamisch finit**, **terminierend**?

3. Formulieren Sie den „umgangssprachlichen“ Algorithmus aus der Vorlesung, der den größten gemeinsamen Teiler zweier Zahlen nach Euklid berechnet, in Pseudocode.

- (a) Sei A die Größere der beiden Zahlen A und B (entsprechend vertauschen, falls dies nicht bereits so ist)
- (b) Setze A auf den Wert $A - B$
- (c) Wenn A und B ungleich sind, dann fahre fort mit Schritt 1, wenn sie gleich sind, dann beende den Algorithmus: Diese Zahl ist der größte gemeinsame Teiler.

4. Der folgende Algorithmus soll den Mittelwert von drei Zahlen a , b und c berechnen.

```
a := 1;
b := 3;
c := 5;
mittel := 0;
mittel := mittel + a + b + c;
mittel := mittel / 4;
```

Ist der Algorithmus korrekt? Warum [nicht]? Wie müsste er ggf. geändert werden?

5. Schreiben Sie einen Algorithmus, der einen Zahlenwert (a) n -mal zu einer Variable *sum*, die zu Beginn den Wert 0 hat, hinzuaddiert. Welcher Komplexitätsklasse gehört der Algorithmus an?
6. Verbessern Sie die Effizienz des Algorithmus aus der vorigen Aufgabe, indem Sie den Algorithmus syntaktisch so umformulieren, dass das Ergebnis bei gegebenem a und n zwar immer noch das gleiche ist, aber eine niedrigere Komplexitätsklasse erreicht wird.