

Übungsblatt 7

Abgabe: 02. - 04.01.06 in den Übungen

Die Lösung dieses Übungsblatts soll in Gruppen von je 2 Personen erfolgen.

Die Lösungen sind inklusive Lösungsweg auf Papier abzugeben. Auf jedem Lösungsblatt sind die Matrikelnummern beider Gruppenmitglieder anzugeben!

Aufgabe 1 (insgesamt 8 Punkte)**(a) (3 Punkte)**

Schreiben Sie eine Java-Methode

```
int[][] transponiere(int[][] matrix);
```

welche eine Matrix transponiert und das Ergebnis zurückliefert.

Beispiel: $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \mapsto \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$

(b) (5 Punkte)

Jeder quadratischen Matrix kann man auf bestimmte Weise einen Skalar zuordnen, der die Determinante der Matrix genannt wird. Die Determinante einer n -quadratischen Matrix ist *von der Ordnung n* .

Schreiben Sie eine Java-Methode

```
int determinante(int[][] matrix);
```

welche die Determinante der Ordnung 3 berechnet und das Ergebnis zurückliefert.

Aufgabe 2 (4 Punkte)

Vergleichen Sie die folgenden vier Java-Methoden zum Vertauschen von zwei Zahlen und erläutern Sie die unterschiedliche Wirkung.

(a)

```
void tausche(int a, int b) {
    int c = a;
    a = b;
    b = c;
}
```

tauscht 2 integer Zahlen durch Zuweisung mit einer Hilfsvariable

(b)

```
void tausche(int[] a) {
    int c = a[0];
    a[0] = a[1];
    a[1] = c;
}
```

tauscht das erste und das zweite Element aus einem Feld, das übergeben wird. Benutzung einer Hilfsvariable

(c)

```
void tausche(int[] a) {
    a[0] = a[1] - a[0];
    a[1] = a[1] - a[0];
    a[0] = a[1] + a[0];
}
```

1) a[0] ist die Differenz von a[1] zu a[0]. 2) a[1] wird sich -(minus) die Differenz zugeordnet. 3) a[0] wird a[1] + die Differenz zugeordnet

(d)

```
void tausche(int[] a) {
    a[0] = a[0] ^ a[1];
    a[1] = a[0] ^ a[1];
    a[0] = a[0] ^ a[1];
}
```

coole Methode mit exklusivem Oder (aufgeführt in meiner Textdatei)

Aufgabe 3 (insgesamt 8 Punkte)

Gegeben sei die rekursive Methode `bewege` in der Datei `Hanoi.java` aus der Vorlesung (II.11).

- (a) **(5 Punkte)** Nehmen Sie folgende Änderung vor: Es sollen nur Bewegungen zwischen benachbarten Türmen erlaubt sein, d.h. für Türme Start, Hilfe, Ziel kann keine Scheibe direkt vom Start zum Ziel bewegt werden.
(Hinweis: Drucken Sie sich das Programm `Hanoi.java` aus, ändern Sie es und kennzeichnen Sie Ihre Änderungen bei der Abgabe. Relevant sind dabei nur die geänderten Zeilen.)
- (b) **(3 Punkte)** Bestimmen Sie das Laufzeitverhalten Ihres Algorithmus' bezogen auf n Scheiben und beweisen Sie Ihr Ergebnis.