Übungen zum Programmieren in Java

Aufgabe 1 Auswertung von Ausdrücken

Gegeben seien folgende Ausdrücke:

(1) `10 == 1 + a ++ * 3`

(2) `a <= b & a > b && true`

(3) `a < b / a ? a *= a : b`

a) Geben Sie die Ausdrücke in vollständiger Klammerung an. Achten Sie dabei auf die Operatorpräzedenzen.

b) Welchen Wert haben die Ausdrücke für die Variablenbelegungen

(i) `a=2` und `b=6` bzw.
(ii) `a=3` und `b=10` ?

c) Zeichnen Sie Operatorbäume für die Ausdrücke.

Aufgabe 2 rekursive Programmierung

Ein Beispiel für rekursiv definierte Zahlenfolgen sind die Fibonacci-Zahlen: Die n-te Fibonacci-Zahl f_n ergibt sich aus der Summe ihrer beiden Vorgänger f_{n-1} und f_{n-2}. Die nullte Fibonacci-Zahl ist 0 und die erste ist 1.

a) Geben Sie eine rekursive Methodendefinition für die Berechnung der n-ten Fibonacci-Zahl an.

b) Geben Sie den Aufrufbaum für die fünfte Fibonacci-Zahl an.

Aufgabe 3 (P) iterative Programmierung

a) Definieren Sie eine iterative Methode, die die n-te Fibonacci-Zahl berechnet. Nutzen Sie dabei auftretende Zwischenergebnisse aus.

b) Geben Sie in einer Schleife die errechneten Funktionswerte sowohl der rekursiven als auch der iterativen Methode bis zur 20-ten Fibonacci-Zahl aus.

Aufgabe 4 (P) rekursiver Sortier-Algorithmus

Geben Sie einen Algorithmus an, der auf folgendem Prinzip basiert:

Gegeben sei ein (eindimensionales) Feld von ganzen Zahlen (`int`). Man wähle eine (beliebige) Zahl, die zwischen der kleinsten und der größten im Feld vorkommenden Zahl liegt (auch die kleinste bzw. größte Zahl selbst kann ausgewählt werden); diese sei im folgenden als Schnittzahl bezeichnet. Mit Hilfe dieser Zahl teile man das Feld in folgende Teile auf:

- einen Teil t_{kl}, in dem sich die Elemente des Felds befinden, die kleiner als die Schnittzahl sind,
- einen Teil t_{gl}, in dem sich die Elemente des Felds befinden, die den Wert der Schnittzahl haben, und
- einen Teil t_{gr}, in dem sich die Elemente des Felds befinden, die größer als die Schnittzahl sind.

Nun wende man dieses Prinzip rekursiv auf die Teile t_{kl} und t_{gr} an und reihe schließlich die Teile wieder aneinander. Dadurch erhält man ein sortiertes Feld.

a) Realisieren Sie diesen Algorithmus in einer rekursiven Methode

```
static void sortiere ( int[] feld ),
```

die die Elemente des eingegebenen Felds aufsteigend sortiert.

b) Verändern Sie die Methode dahingehend, daß als Ergebnis ein sortiertes Feld geliefert wird, wobei das eingegebene Feld nicht verändert werden soll.

c) Testen Sie Ihre Methoden mit einem Feld, das folgende Elemente enthält:

```
9, 7, 3, 4, 5, 6, 8, 0, 1, 2
```

Aufgabe 5 Zugriffe auf Felder

a) Wie lassen sich zwei gleichlange Felder a und b auf inhaltliche Gleichheit (Gleichheit bzgl. der Feldelemente) überprüfen?

b) Gegeben sei eine Darstellung eines $8*8$-Spielbrettes als Feld. Auf diesem Brett bewegt sich eine Spielfigur im Rösselsprung[1]. Linker und rechter Rand des Feldes sowie oberer und unterer Rand des Feldes seien miteinander verbunden, d.h. wenn die Spielfigur nach oben hinaushüpft, kommt sie im unteren Bereich des Spielfeldes wieder heraus. Wie läßt sich die neue Position der Figur aus der alten Position mit Hilfe des "`%`"-Operators errechnen? Welche Probleme treten dabei auf?

[1] Als Rösselsprung bezeichnet man die Bewegung der Springerfigur beim Schach. Dabei wird die Figur um zwei Felder "geradeaus" und ein Feld "zur Seite" bewegt.