

# Übung 2

## Wiederholung JAVA, Arrays und Strings

1. Berechnen Sie den Speicherbedarf (in Bytes) eines „Schachprogrammes“, das alle Permutationen von Schachfiguren-Positionen (6 verschiedene Figuren je Spieler bei 2 Spielern, oder Leerfeld) auf einem Schachbrett mit 8x8 Feldern für eine Schachpartie mit maximal 100 Zügen speichern kann. Dabei werden die Schachregeln und die Tatsache, dass es jede Figur nur in einer begrenzten Anzahl in einem Spiel geben kann, und pro Feld nur maximal eine Figur aufgestellt werden kann, außer acht gelassen.

Hinweis: Ob eine bestimmte Figur auf einem Feld steht, kann durch ein Boolean (vereinfacht angenommen, ein Bit) festgelegt werden.

2. Schreiben Sie eine Funktion, die ein Polynom  $n$ -ten Grades der Form

$$P(x) = \sum_{i=0}^n a_i x^i = a_0 + a_1 \cdot x + a_2 \cdot x^2 + a_3 \cdot x^3 + \dots + a_n \cdot x^n$$

an der Stelle  $x$  berechnet, wobei die Koeffizienten  $a_0 \dots a_n$  als **double**[]-Array und die Variable  $x$  als **double** übergeben werden. Das Ergebnis soll ebenfalls vom Typ **double** sein.

3. Schreiben Sie eine Funktion, die einen **String** als Parameter erhält, der ausschließlich aus Zeichen **0 . . . 9** besteht, und die Dezimalzahlenrepräsentation als **int** zurückliefert. Benutzen Sie eine Schleife, die Zeichen für Zeichen auswertet, und nicht die in der Java-Bibliothek vorhandenen, fertigen Konvertierungsmethoden.
4. Vereinfachen Sie die in der letzten Aufgabe programmierte Funktion, indem Sie die in Java vorhandenen Typ-Konvertierungsmethoden verwenden, wobei die Signatur der Funktion gleich bleibt.
5. Programmieren Sie folgende Code-Sequenz in einem Java-Testprogramm, und überlegen Sie sich, wie und warum es zu unterschiedlichen Ergebnissen in den Zeilen 5 und 6 kommt.

```
1 String s1 = new String("123");
2 String s2 = new String("123");
3 String s3 = "123";
4 String s4 = "123";
5 System.out.println(s1 == s2);
6 System.out.println(s3 == s4);
```