

## Übungsblatt 10

Abgabe bis Montag, 14.07.08, 16 Uhr

### Hinweis:

Programmieraufgaben immer per Email (eine Email pro Blatt und Gruppe) an den zuständigen Tutor schicken (Java Quellcode und eventuell benötigte Datendateien). Bitte werfen Sie Ihre schriftlichen Lösungen in die Briefkästen in Geb. 051, Erdgeschoss ein. Für den Erhalt von Bonuspunkten müssen Sie in wenigstens 9 Übungen anwesend sein und müssen wenigstens 9 Übungszettel bearbeitet haben.

### Aufgabe 10.1

Schreiben Sie eine Klasse `BinaryTree` für Integerzahlen. Die Klasse könnte wie folgt aussehen:

```
class BinaryTree {
    BinaryTree(int value) {
        content = value;
        right = null;
        left = null;
    }
    int content;
    BinaryTree left;
    BinaryTree right;
}
```

und sollte unter anderem folgende Methoden beinhalten:

- `boolean contains(int v)`  
Überprüft, ob ein Knoten mit dem Schlüssel `v` existiert.
- `void insert(int v)`  
Fügt einen Knoten mit Schlüssel `v` in den Baum ein, sofern noch kein Knoten mit diesem Schlüssel existiert. Dabei sind im linke Teilbaum ausschließlich Knoten mit kleineren Zahlen als der Wurzel enthalten und im rechte Teilbaum entsprechend nur größere.
- `void delete(int v)`  
Löscht den Knoten mit dem Schlüssel `v`, falls er existiert.

**Aufgabe 10.2** Erweitern Sie die Klasse `BinaryTree` um folgende Methoden:

- `void preorder()`  
Gibt zuerst den Schlüssel der Wurzel und dann die Schlüssel der linken und rechten Teilbäume aus.
- `void inorder()`  
Gibt zuerst den linken Teilbaum, dann den Schlüssel der Wurzel und danach den rechten Teilbaum aus.
- `void postorder()`  
Gibt zuerst den linken und rechten Teilbaum und danach den Schlüssel der Wurzel aus.

Überprüfen Sie ihre Implementierung mithilfe der Testklasse `TestBinaryTree.java`, die Sie auf der Homepage zur Übung finden.

1. An welchen Stellen in einem Binärbaum befinden sich die Knoten mit dem kleinsten, bzw. größten Schlüssel?
2. Anhand welcher Ausgabefunktion (`preorder`, `inorder`, `postorder`) kann auf den Aufbau des Binärbaums geschlossen werden?

### **Aufgabe 10.3**

Betrachten Sie ein Schachbrett (8x8 Felder) und einen einzelnen Springer. Schreiben Sie eine Klasse, die ein Schachbrett beliebiger Größe repräsentiert und alle möglichen Spielzüge eines Springers berechnen kann. Ein Springer bewegt sich immer um zwei Felder horizontal und eines vertikal bzw. ein Feld horizontal und zwei vertikal. Mit Ihrer Klasse soll bestimmt werden können, welche Felder ein Springer ausgehend von Startposition  $(x, y)$  in maximal  $n$  Zügen erreichen kann.

1. Führen Sie eine Aufwandsabschätzung für Ihr Programm durch und geben Sie den Aufwand in der O-Notation an. Was ist eine geeignete Problemgröße  $n$ ?
2. Welche Felder können auf einem normalen Schachbrett ausgehend von Feld  $(0, 0)$  in maximal 4 Zügen erreicht werden?
3. Wieviele Züge sind mindestens notwendig, um von Startposition  $(0, 0)$  ein beliebiges Feld auf einem Brett der Größe 18x18 zu besuchen?

Hilfestellung:

- Verwenden Sie zur Darstellung des Schachbretts ein zweidimensionales Array.
- Stellen Sie beispielsweise die Startposition des Springers durch eine 2, ein unbesuchtes Feld durch eine 0 und ein erreichbares Feld durch eine 1 dar.
- Schreiben Sie eine Methode, die ausgehend von einem Feld alle erreichbaren Felder bestimmt. Um mehrere Züge auszuführen, sollte diese Methode dann rekursiv aufgerufen werden.