

## Aufgabenblatt 9

### Aufgabe 1, Implementierung eines Suchbaumes (6 P)

Implementieren Sie einen binären Suchbaum, genauso, wie in der Vorlesung diskutiert. Benutzen Sie dazu bitte die Datei

<http://www-ab.informatik.uni-tuebingen.de/teaching/ws07/algo/BinarySearchTree.java>.

Schreiben Sie die folgenden Methoden: `insert`, `delete`, `search`, `getSuccessor`, `getPredecessor`, `getMinimum`, `getMaximum` und `print`.

In dem Baum sollen verschiedene Elemente mit demselben *key* vorkommen dürfen. Die Methoden `search` und `delete` sollen jeweils ein Element mit dem gegebenen *key* finden oder löschen. Die Methode `print` soll den aktuellen Baum möglichst übersichtlich ausdrucken.

Ein- und Ausgabe sind in der `main` Methode bereits implementiert und es werden dort die folgenden Kommandos unterstützt:

Insert: `insert <key>*`

Delete: `delete <key>*`

Search: `search <key>*`

Search for all occurrences: `all <key>*` (siehe Aufgabe 2)

Successor: `succ <key>*`

Predecessor: `pred <key>*`

Minimum: `min`

Maximum: `max`

Print tree: `print`

Quit: `quit`

### Aufgabe 2, Alle Elemente finden (2 P)

Implementieren Sie zusätzlich eine Methode `searchAll`, die mit Hilfe der Methoden `getSuccessor` und `getPredecessor` rekursiv *alle* Vorkommnisse eines Schlüssels als Java Liste zurück gibt.

### Aufgabe 3, Beispiel (2 P)

Zeigen Sie bitte an Hand eines Beispiels, dass die verschiedenen Funktionalitäten Ihres Programmes korrekt zu funktionieren scheinen.

Bitte senden Sie die Lösungen per Email an Daniel Huson mit Subject: Aufgabenblatt 9.