

Aufgabe 3 (AVL-Baum)

Zur Erinnerung: AVL-Baum

- Binärbaum
- Suchbaum, d.h. Elemente geordnet: linker Sohn kleiner als der rechte Sohn
- höhenbalancierter Baum: $| \text{Höhe links} - \text{Höhe rechts} | \leq 1$

Fügen Sie nacheinander die Schlüssel 7, 20, 31, 65, 6, 22, 30, 29, 79, 24, 80, 63 in einen ursprünglich leeren AVL-Baum ein. Löschen Sie anschließend nacheinander die Schlüssel 7, 20, 80 aus dem Baum. Zeichnen Sie dabei den Baum jeweils vor und nach den Strukturänderungen auf und begründen Sie genau, welche Strukturänderungen (durch Rotationen) durchgeführt werden müssen.

Aufgabe 4 (Dictionary-ADT AVL in Java)

Programmieren Sie die AVL-Baum-Datenstruktur mit den entsprechenden ADT-Funktionen

- *Empty, IsEmpty, Insert, Delete, IsMember*
- *IsPredecessor*
- *IsSuccessor*
- *Predecessor,*
- *Successor*

in Java. Schreiben Sie eine GUI, um das Suchen, Einfügen und Löschen von Knoten in einen AVL-Baum zu testen und im Praktikum zu demonstrieren.

Hinweis: Sie können AVL-ADT-Klassen aus dem Internet verwenden, müssen aber im Praktikum zeigen, dass Sie diese auch verstanden haben.

Dokumentieren Sie alles sorgfältig!