

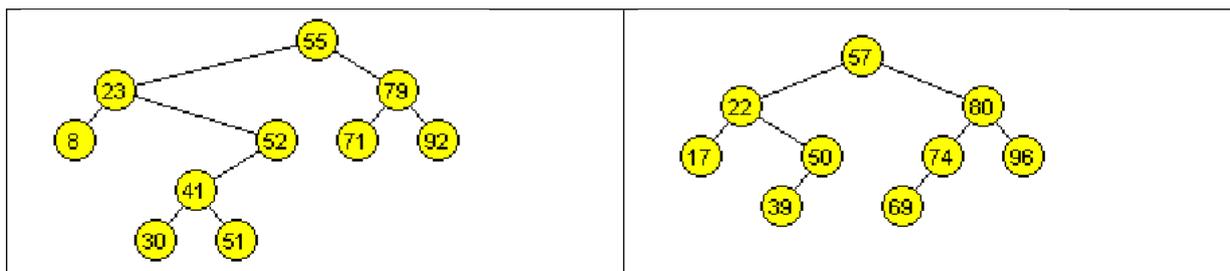
**Präsenzaufgaben 4**

**11.04.2023**

Die Lösung der Aufgaben wird am Ende der Übung von Ihnen vorgestellt.

**Aufgabe 1**

Schreiben Sie an die Knoten der beiden folgenden binären Suchbäume den AVL-Index und geben Sie anschließend an, ob es sich um AVL-Bäume handelt.



**Aufgabe 2**

In Java gibt es verschiedene Klassen und Interfaces für assoziative Felder, die das Interface `Map` implementieren. Es gibt die Klassen `HashMap`, `LinkedHashMap` und `TreeMap`, sowie die Interfaces `NavigableMap` und `SortedMap` (die weiteren Klasse und Interfaces haben ganz spezielle Einsatzzwecke).

`HashMap` benutzt eine Hashtabelle, `TreeMap` benutzt einen Rot-Schwarz-Baum (spätere Vorlesung, ist aber ähnlich wie ein AVL-Baum).

- Beschreiben Sie die Datenstruktur der Klasse `LinkedHashMap` (-> API). Welcher Nachteil der Klasse `HashMap` wird mit der `LinkedHashMap` ausgeglichen?
- Was kann eine `TreeMap` mehr als eine `HashMap`? Beachten Sie, dass die `TreeMap` zusätzlich das Interface `NavigableMap` implementiert.

**Aufgabe 3**

Zeichnen Sie jeweils die entstehenden B-Bäume.

a)		Löschen der 15
b)		Einfügen der 75
c)		Einfügen der 68

**Aufgabe 4**

Zeichnen Sie jeweils die entstehenden B-Bäume.

a)		Löschen der 43
b)		Löschen der 36
c)		Löschen der 90

**Aufgabe 5**

AVL-Suchbäume lassen sich auch als Listen verwenden. Die Grundidee ist, dass man in jedem Knoten die Anzahl der Knoten im linken und rechten Teilbaum speichert. Eine entsprechende Klasse gibt es in der Java-Standardbibliothek leider nicht. Es gibt aber zwei sehr bekannte externe Bibliotheken: *Apache Commons* und *Guava*.

- Sehen Sie sich das Video `TreeList.mp4` (ILIAS, Woche 4) an.
- Apache Commons und Guava stehen unter der Apache Commons Lizenz. Es ist wichtig, einen groben Überblick über die freien Softwarelizenzen zu haben. Lesen Sie dazu die folgende Seite:  
<https://exygy.com/blog/which-license-should-i-use-mit-vs-apache-vs-gpl/>  
 Anmerkung: Dieser Punkt ist nicht klausurrelevant. Sie sollten die Seite im eigenen Interesse trotzdem lesen.
- Lassen Sie das Testprogramm `ArrayListTest.java` (ILIAS, Woche 4) laufen und notieren Sie die Zeiten für das Einfügen vorne, hinten und in der Mitte.
- Ersetzen Sie die `ArrayList` durch eine `TreeList`.
- Notieren Sie erneut die Zeiten.
- Schreiben Sie alle Zeiten und die jeweilige O-Klasse für das Einfügen eines Elements (Average Case) in eine Tabelle.