

## AP 11: AVL-Bäume (balancierte Suchbäume) - Implementierung

Erstellen Sie für eine Implementierung und grafische Ausgabe eines AVL-Baumes im Projekt mit dem Geschenkkatalog ein neues Package `avl`. Importieren Sie anschließend die Klassen `AVLbaum` und `AVLMain` in dieses Package.

Die Vorlage wird nun schrittweise für die Erstellung eines AVL-Baumes erweitert.

1. Erweitern Sie die innere Klasse `AVLNode` um zwei Variablen `hoeheRechterTeilbaum` und `hoeheLinkerTeilbaum`. Erstellen Sie in der inneren Klasse ebenfalls eine Methode `int getBalanceFaktor()`, welche mithilfe der beiden Variablen den Balance-Faktor bestimmt und zurückgibt.
2. Implementieren Sie in der Klasse `AVLbaum` eine private Hilfsmethode `int getHoehe(AVLNode node)`, welche die Höhe der übergebenen Baumstruktur mithilfe von `hoeheLinkerTeilbaum` und `hoeheRechterTeilbaum` bestimmt. Ein Knoten ohne Nachfolger hat dabei die Höhe 1, beachten Sie, falls `node null` sein sollte.
3. Passen Sie nun die Methode `add` an, so dass die Variablen `hoeheRechterTeilbaum` und `hoeheLinkerTeilbaum` beim Einfügen eines Elements in jedem betroffenen Knoten korrekt aktualisiert wird.
4. Implementieren Sie eine neue private Methode `void balance(AVLNode node)`, welche überprüft, ob beim übergebenen Knoten eine Rotation durchgeführt werden muss. Falls ja, soll die entsprechende Rotation durchgeführt werden. Dabei soll für jede Rotation eine entsprechende Methode erstellt werden, welcher Sie `node` übergeben. Die Methodenrumpfe dieser Methoden können Sie erstmal leer lassen.
5. Implementieren Sie nun die Methode für die Linksrotation wie folgt:
  - Passen Sie zuerst die Referenz des linken bzw. des rechten Nachfolgers im Elternknoten von `node` an. Beachten Sie, falls `node null` ist.



- Ändern Sie die Referenz von `parentNode` in `node` und im rechten Nachfolger von `node`.



- Tauschen Sie die Referenz des rechten Nachfolgers von `node` mit dem linken Teilbaum vom rechten Nachfolger von `node` aus und umgekehrt.



- Aktualisieren Sie die Höhe des rechten Teilbaums von `node` und die Höhe des linken Teilbaums des neuen Elternknoten von `node`. Verwenden Sie dabei die Hilfsmethode `getHoehe`.



6. Implementieren Sie nun die Methode für die Rechtsrotation analog zur Linksrotation. Sie können Ihre Implementierung mit der `Main`- Klasse testen.
7. Implementieren Sie nun die beiden Methoden für die Doppelrotationen mithilfe der Methoden für die Links- bzw. Rechtsrotation. Diese können Sie nun ebenfalls mithilfe der `Main`-Klasse testen.