Abgabe und Beurteilungskriterien

Jede Hausübung wird mit 100% bewertet, wobei diese 9% der Gesamtbewertung der ILV ausmachen.

Geben Sie die Hausübung bis zum 2.11.2025 um 23:59 ab.

Abzugeben ist ein ZIP-File, welches das Java-Projekt beinhaltet.

Benennung: **nachname_uebung3_bswe_algo_ws2025.zip**

Kann ein Java-Projekt nicht kompiliert werden, gibt es dafür Punkteabzug.

Bitte verwenden Sie das bereitgestellte Template.

Bewertet wird vorrangig die korrekte Funktionalität. Sind laut Angabe bestimmte Vorgaben zu erfüllen (beispielsweise das Verwenden einer bestimmten Kontrollstruktur), gibt es Punkteabzug, wenn diese nicht erfüllt werden.

Sollten Fragen oder Unklarheiten zu der Hausübung auftauchen, bitte ein E-Mail an patrizia.sailer@hochschule-burgenland.at oder sebastian.saip@hochschule-burgenland.at senden.

Die Bewertung wird zeitnah nach der Deadline erfolgen.

Aufgabenstellung

Aufgabenstellung: AVL-Baum traversieren

Ziel:

In dieser Übung geht es darum, ein Java-Programm zu schreiben, das die Traversierung eines AVL-Baums ermöglicht. Der Baum wird auf Basis einer eingegebenen Zahlenreihe erstellt. Das Programm muss die Zahlen entsprechend der AVL-Regeln einordnen. Achten Sie auf die korrekten Rotationen.

Anforderungen:

1. **Eingabe:** Es kann über die Konsole eine beliebig lange Reihe von Ganzzahlen, durch Komma getrennt, eingegeben werden, aus der in weiterer Folge der Baum entstehen soll.

   Beispiel: 8, 4, 9, 7, 2, 13, 11, 46

   Das Programm erstellt daraus dann den Baum und prüft, Zahl für Zahl, ob die AVL-Bedingungen erfüllt sind, oder ob/welche Rotation angewandt wird.

   Dieser Schritt soll über eine entsprechende Ausgabe transparent gemacht werden.

   Weiters soll es möglich sein, mittels String (case-sensitive, lowercase) die Traversierungsart anzugeben. Es gibt folgende Werte:

   „preorder“
   „levelorder“
   „inorder“
   „postorder“

   Der Einstiegspunkt für die Traversierungs-Klasse soll eine public static Methode mit ZWEI Parametern (int[] numbers, String order) sein. Returnwert int[].

2. **Ausgabe:** Während dem Erstellen des Baums sollen die AVL-Prüfungen **ausgegeben** werden. Anschließend soll der Baum dargestellt werden, um zu sehen, ob das Einfügen korrekt funktioniert:

   Anschließend sollen die Knoten abhängig von der gewählten Traversierungsart korrekt **ausgegeben** werden und von der Methode als int[] returned werden.

Aufgaben:

- AVL-Regeln prüfen: Während der Erstellung des Baumes soll das Programm prüfen, ob der Baum die AVL-Baum-Eigenschaften erfüllt.
- Ausgabe der AVL-Prüfung
- Baum darstellen: Der Baum soll nach dem Einfügen der Zahlen in einer baumartigen Struktur angezeigt werden.
- Traversierung: Je nach Eingabe soll der Baum unterschiedlich traversiert und die Knoten entsprechend ausgegeben und von der Methode zurückgegeben werden.

Abgabe:

- **Code:** Geben Sie den vollständigen Quellcode in Java ab.
- **Komplexität:** Best- und Worst-Case der Algorithmenkomplexität beschreiben (als README.md im Java Projekt)
- **Testfälle:** Stellen Sie sicher, dass Ihre Lösung mit verschiedenen Testfällen getestet wird.

Hinweise:

- Für das Zeichnen des Baums und die Ausgabe der Knotenfolge soll eine textbasierte Ausgabe verwendet werden, z. B. in einer hierarchischen Form wie skizziert.
- Es kann mittels System.out oder Logging-Framework über die console geloggt werden.
- Für die AVL-Prüfung müssen die Balancefaktoren jedes Knotens berechnet und überprüft werden.
- Zahlen können nur einmal im Baum vorkommen.