



Übungsblatt 3

Datenstrukturen und Algorithmen (SS 2016)

Abgabe: Mittwoch, 04.05.2016, 23:59 Uhr — Besprechung: ab Montag, 09.05.2016

Bitte lösen Sie die Übungsaufgabe in **Gruppen von 3 Studenten** und wählen EINEN Studenten aus, welcher die Lösung im ILIAS als **PDF** als **Gruppenabgabe** (unter Angabe aller Gruppenmitglieder) einstellt. Bitte erstellen Sie dazu ein **Titelblatt**, welches die Namen der Studenten, die Matrikelnummern, und die E-Mail-Adressen enthält.

Die Aufgaben mit Implementierung sind mit Impl gekennzeichnet. Das entsprechende Eclipse-Projekt kann im ILIAS heruntergeladen werden. Bitte beachten Sie die Hinweise zu den Implementierungsaufgaben, die im ILIAS verfügbar sind.¹

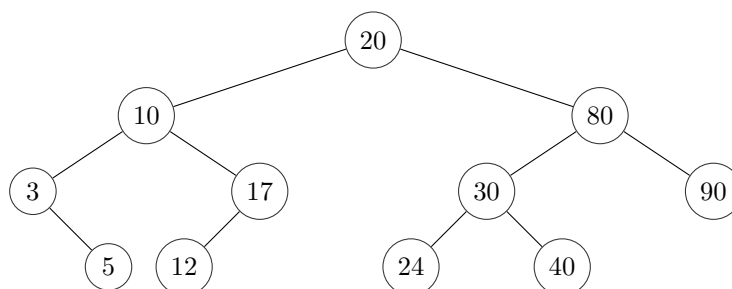
Dieses Übungsblatt beinhaltet 4 Aufgaben mit einer Gesamtzahl von 30 Punkten.

Aufgabe 1 Impl Iterator [Punkte: 10]

Gegeben im Eclipse-Projekt sind die Schnittstellen `ISimpleList` und `ISimpleListIterable`. Erweitern Sie die Klasse `SimpleList` (ohne die bestehenden Methoden zu modifizieren), damit sie die Schnittstelle `ISimpleListIterable` implementiert. Erstellen Sie hierzu zwei Iterator-Klassen als innere Klassen der Klasse `SimpleList`. **Die Verwendung existierender Iteratorimplementierungen (z.B. `ArrayList.iterator()` sowie Iteratoren anderer Datenstrukturen aus der Java-Klassenbibliothek) ist dabei nicht erlaubt.** Die Methode `remove` der Iteratoren wird nicht unterstützt und sollte immer `UnsupportedOperationException` werfen.

Aufgabe 2 Binäre Suchbäume, AVL-Bäume [Punkte: 6]

Gegeben sei der binäre Suchbaum B_1 :



- (1 Punkte) Erstellen Sie den Baum B_2 , indem Sie in B_1 die Werte 4 und 94 einfügen.
- (1 Punkte) Erstellen Sie den Baum B_3 , indem Sie aus dem ursprünglichen Baum B_1 die Werte 10 und 40 entfernen. Ersetzen Sie, wo nötig, Knoten durch ihre Inorder-Vorgänger.
- (1 Punkte) Geben Sie für jeden Knoten aus dem ursprünglichen Baum B_1 den Wert der AVL-Balance an.
- (3 Punkte) Erstellen Sie den Baum B_4 , indem Sie in den *ursprünglichen* Baum B_1 den Wert 81 einfügen. Erstellen Sie den Baum B_5 , indem Sie in den *ursprünglichen* Baum B_1 den Wert 22 einfügen. Erstellen Sie den Baum B_6 , indem Sie in den *ursprünglichen* Baum B_1 den Wert 50 einfügen. Führen Sie für jeden Baum (B_4 , B_5 , B_6) nach dem Einfügen etwaige Schritte durch, um die AVL-Balance wiederherzustellen. Geben Sie dazu auch an, welche Schritte nötig waren, um die AVL-Balance wiederherzustellen.

¹https://ilias3.uni-stuttgart.de/goto_Uni_Stuttgart_fold_997779.html

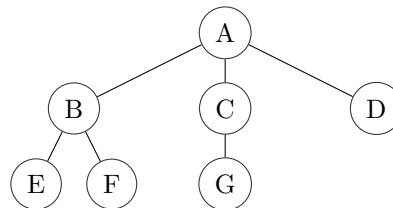
Aufgabe 3 Impl Binäre Suchbäume [Punkte: 10]

Gegeben im Eclipse-Projekt sind die Schnittstellen `IBinaryTreeNode` und `IBinarySearchTree` für binäre Suchbäume.

- (a) (2 Punkte) Implementieren Sie die Knotenklasse `BinaryTreeNode`, die die Schnittstelle `IBinaryTreeNode` implementiert. Die Knotenklasse muss einen Standard-Konstruktor (d.h. Konstruktor ohne Parameter) enthalten.
- (b) (8 Punkte) Implementieren Sie die binäre Suchbaum-Klasse `BinarySearchTree`, die die Schnittstelle `IBinarySearchTree` implementiert. **Ignorieren Sie dabei Dublikate, d.h. wenn ein Schlüsselwert eingefügt werden soll, welcher im Baum bereits vorhanden ist, dann soll nichts passieren bzw. der Versuch des Einfügens ignoriert werden.** Die Suchbaum-Klasse muss einen Standard-Konstruktor (d.h. Konstruktor ohne Parameter) enthalten.

Aufgabe 4 Binarisierung von geordneten Bäumen [Punkte: 4]

- (a) (2 Punkte) Wandeln Sie den folgenden geordneten Baum gemäß dem in der Vorlesung vorgestellten Schema (Foliensatz 5, Folie 20) in einen Binärbaum um.



- (b) (2 Punkte) Der folgende Baum wurde gemäß dem in der Vorlesung vorgestellten Schema in einen Binärbaum umgewandelt. Geben Sie den ursprünglichen Baum (vor der Umwandlung) an.

