Prof. Dr. Franz Schweiggert — Abteilung Angewandte Informationsverarbeitung — 20.07.2006
Norbert Heidenbluth — Blatt 9

Allgemeine Informatik II (SS 2006)

für
(Wirtschafts-)Mathematiker/Physiker und E-Techniker

Abgabetermin: 13. Juli 2006

Vorbemerkung: Dieses Blatt richtet sich nur noch an diejenigen, denen noch einige (wenige) der für den Schein erforderlichen 60 Übungspunkte fehlen.

Um diesen Kandidaten eine faire Chance einzuräumen, ist diese Aufgabe relativ einfach gehalten. Im Vordergrund steht dafür aber, im Tutorium eine eigenständige Erläuterung der Lösung liefern sowie ein Gespräch über die mit diesem Übungsblatt behandelten Themen (Rekursion, Bäume, Ausgeglichenheit, Traversierung) führen zu können.

Bonusaufgabe: Baum-Klasse deluxe (10 Punkte)

Wieder einmal wollen wir eine Luxus-Version einer im Skript vorgestellten Datenstruktur haben. Diesmal geht es um binäre Bäume.

In der Vorlesung haben wir einerseits die verschiedenen Traversierungsarten sowie andererseits die Ausgeglichenheit von Bäumen (Höhen- und Gewichtsausgeglichenheit) behandelt. Nun soll eine Klasse geschrieben werden, welche einen binären Baum repräsentiert sowie diesen auf alle drei bekannten Arten traversieren und hinsichtlich der Ausgeglichenheits-Eigenschaften untersuchen kann.

Teil a:

Implementieren Sie eine minimalistische Version der Klasse `BinaryTree`, welche einen Binärbaum repräsentiert: In jedem Knoten sollen beliebige Objekte aufgenommen werden können. Sie benötigen hierzu zunächst die privaten Felder sowie einen Konstruktor.

Teil b:

Ergänzen Sie Ihre Felder um ein privates Attribut `order`, mit welchem eingestellt werden kann, ob der Baum bei Aufruf der `toString()`-Methode inorder, preorder oder postorder traversiert wird. Schreiben Sie die Methoden `setOrderToPreorder()`, `setOrderToInorder()` und `setOrderToPostorder()`, mit welchen die Reihenfolge entsprechend umgestellt wird und schreiben Sie die Methode `toString()`, welche in Abhängigkeit der momentan eingestellten Sortierung den Baum traversiert und ausgibt.

Teil c:

Schreiben Sie nun die Instanz-Methoden `isWeightBalanced()` und `isHeigthBalanced()`, welche den Wahrheitswert zurückliefern, ob der Baum die entsprechende Ausgeglichenheits-Eigenschaft besitzt.

Tip:

Sie können sich für diese Aufgabe der Lösung von Blatt 8 bedienen, um einen binären Baum aufbauen zu lassen und somit ein "Testobjekt" zu haben.

Viel Erfolg!