Praktische Informatik II SS 2007

Übungsblatt 4

Abgabe: 21.05.2007

Aufgabe 1 So lasst uns einen Binärbaum pflanzen...

Da in einem binären Baum (fast) genauso effizient gesucht werden kann wie in einem Array, der Baum aber genauso flexibel erweiterbar ist wie eine Liste, sollt Ihr also den Erklärungen der Vorlesung jetzt Taten folgen lassen und binäre Bäume implementieren.

Aufgabe 1.1 Ein Baum ist ein Blatt ist ein Baum

Eure Implementierung soll den Prinzipien der Objektorientierung folgen. Beachtet dabei, dass es nicht nötig ist, eine Knotenklasse zu definieren, da jeder Knoten in einem Baum auch selbst ein Baum ist.

Die zu implementierenden Aufgaben sind

1. Einfügen eines Datensatzes an der richtigen Stelle entsprechend der Sortierung.
2. Suchen nach einem Datensatz.
3. Löschen eines Datensatzes.
4. Sortiertes Ausgeben aller Datensätze.
5. Angabe der Anzahl der Einträge im Baum und der Höhe des Baums.
6. Erstellen einer *tiefen* Kopie des Baumes. Dabei werden nicht nur Knoten neu erstellt, sondern auch die Datensätze werden kopiert.

Der Binärbaum soll vielseitig einsetzbar sein, deshalb soll in der Implementierung zunächst von den Datensätzen abstrahiert werden. Um ein geordnetes Einfügen zu ermöglichen, sollen die Datensätze allerdings eine Methode `compareTo()` besitzen, und somit das Interface `Comparable` implementieren. Beachtet auch, dass die Ausgabe des Baumes nur mit Methoden erfolgen darf, die alle potentiellen Datenobjekte besitzen (`toString()` hat jedes Object, aber zeigt nur selten das gewünschte Format).

Aufgabe 1.2 Gefüllte Bäume

Um die in Aufgabe oben erstellte Struktur zu testen, sollt Ihr als Datenmaterial Immatrikulationsdaten (ja, mal ein sehr ausgefallenes Beispiel) verwenden, etwa mit Datenfeldern wie Name, Matrikelnummer, Schuhgrösse, Postleitzahl (wg. Bremenkindregelung), usw. usw. usf..

Wählt eine angemessene Datenstruktur und definiert die notwendigen Methoden, um aussagefähige Tests erstellen.