
Programmierstarthilfe WS 2008/09
Fakultät für Ingenieurwissenschaften und Informatik

12. Blatt

Für die Woche vom 19.1. bis zum 23.1.2009 (KW 4)

Organisatorisches

Die Webseiten zur Veranstaltung sind unter <http://www.uni-ulm.de/in/mi/lehre/2008ws/programmierstarthilfe.html> zu finden.

Diese Woche arbeiten wir mit Bäumen. Wir werden nun sehen, dass rekursive Methoden praktisch und effizient zugleich sein können.

1 Aufgaben

1.1 Binäre Suchbäume

Schreibe eine Klasse `BinBaumKnoten`, mit der sich Knoten eines binären Suchbaums erstellen lassen. Jeder Knoten enthält einen `int`-Wert und Referenzen auf einen linken und einen rechten Nachfolgerknoten vom gleichen Typ. Du kannst dich dabei an der Liste vom letzten Mal orientieren.

Schreibe außerdem eine Klasse `BinBaum`, die eine Referenz auf die Wurzel des Binärbaums enthält.

- a) Schreibe einen Konstruktor für die Klasse `BinBaumKnoten`, der einen `int`-Wert übergeben bekommt und einen Knoten mit entsprechendem Wert anlegt.
- b) Schreibe einen Konstruktor für die Klasse `BinBaum`, der einen `int`-Wert übergeben bekommt und einen neuen Baum erzeugt, dessen Wurzel ein Knoten mit dem übergebenen Wert erstellt.
- c) Schreibe eine einfache Einfügemethode für die Klasse `BinBaum`. Sie bekommt einen `int`-Wert und die Wurzel des Baums übergeben. Ist der Wert kleiner oder gleich als der der Wurzel ruft sie sich rekursiv auf, allerdings mit dem linken Nachfolgerknoten, ansonsten mit dem rechten. Ist der entsprechende Nachfolger `null` wird an dieser Stelle ein neuer Knoten mit dem entsprechenden Wert eingefügt.
- d) Schreibe in der Klasse `BinBaum` eine Methode `boolean contains(int wert)`, die überprüft ob ein Integer-Wert im Baum vorhanden ist. Bei der Suche kannst du dich an der vorherigen Teilaufgabe orientieren.
- e) Schreibe für die Klasse `BinBaum` die rekursiven Methoden `String preOrder()`, `String postOrder()` und `String inOrder()`, welche den Baum in gegebener Weise traversieren (durchlaufen) und als `String` ausgeben.

1.2 Binäre Suchbäume 2

Die Klasse `Binbaum` soll nun um weitere Methoden erweitert werden. Teste diese mit einem selbst erstellten Binärbaum.

- a) Schreibe eine Methode für den Binärbaum, welche zu einem gegebenen Knoten den Knoten mit der kleinsten ID in seinem Teilbaum sucht.
`BinBaumKnoten smallestNode(BinBaumKnoten start)`.
- b) Schreibe eine Methode `int countLeaves()`, welche die Anzahl der Blattknoten zählt und zurück gibt.
- *c) Schreibe zu Ausgabe des Binärbaum eine Methode, die diesen rekursiv auf der Konsole ausgibt: `void printOut()`.
- *d) Entwickle eine Methode `void deleteNode(int id)`, welche den Knoten mit der gegebenen ID löscht und dafür sorgt, dass die Ordnung des neuen Binärbaums gewährleistet bleibt.

2 Für das nächste Blatt

Nächste Woche werden wir die Java API behandeln. Lies dazu den Anhang im Skript.