Algorithmen und Datenstrukturen (IB) Praktikum
Aufgabenblatt 8 – Binärbäume

Ausgabe: 24. Mai 2018
Abgabe: 5. Juni 2018, 18:00 Uhr

1 Knobelaufgaben (Pflicht)

1.1 Rekonstruktion Binärbäume I

Ein Binärbaum T enthält genau die Zahlen 1 bis 7. Sie wissen, dass bei einem Durchlaufenper Breitensuche die Knoten in der folgenden Reihenfolge besucht werden: 2,1,4,3,7,5,6. Beim Durchlaufen in in-order werden die Knoten in der folgenden Reihenfolge besucht: 1, 3, 5, 2, 7, 6, 4. Zeichnen Sie T!

1.2 Rekonstruktion Binärbäume II

Ein Binärbaum T enthält genau die Zahlen 1 bis 7. Sie wissen, dass bei einem Durchlaufen in post-order die Knoten in der folgenden Reihenfolge besucht werden: 5, 6, 4, 7, 1, 2, 3. Beim Durchlaufen in in-order werden die Knoten in der folgenden Reihenfolge besucht: 5, 4, 6, 7, 3, 1, 2. Zeichnen Sie T!

1.3 Eindeutigkeit

Begründen oder widerlegen Sie, dass ein Binärbaum eindeutig beschrieben ist, wenn bekannt ist, in welcher Reihenfolge die Knoten in pre-order und in post-order durchlaufen werden!

2 Rekursive Berechnungen auf Binärbäumen (Pflicht)

Sie finden im Moodle ein kleines Maven-Projekt mit der aus der Vorlesung bekannten Implementierung eines Binärbaumes. Die Klasse `BinaryTree` repräsentiert einen allgemeinen Binärbaum. Instanzen lassen sich nur über den Konstruktor erzeugen. Die Methoden zur Traversierung (Breitensuche und Tiefensuche in allen Varianten) sind bereits implementiert, ebenso die aus der Vorlesung bekannten Methoden `getHeight()` und `getSize()`. Die Funktionsweise der Klasse können Sie den Tests in `BinaryTreeTest` oder der Klasse `BinaryTreeDemo` entnehmen.

2.1 Blätter zählen

Implementieren Sie die Methode `int getLeafCount()`, so dass Sie rekursiv die Anzahl der Blätter des Baumes zurückgibt!

2.2 Elemente finden

Implementieren Sie die Methode `boolean contains(T element)`, so dass diese rekursiv ermittelt, ob ein Element bereits im Baum enthalten ist (Vergleich mit `equals`!).