

Übungsblatt 1

Ausgabe: 27.3.2018

Abgabe: 11.4.2018 (23:59 Uhr)

Aufgabe 1: ADT BinaryTree - *Programmieraufgabe* -

95 Punkte

Der in der Vorlesung vorgestellte ADT BinaryTree (binärer Suchbaum) stellt über das Interface BinaryTree die folgende Funktionalität zur Verfügung:

<code>boolean insert (int val)</code>	Fügt val in den Baum ein.
<code>boolean insert (String filename)</code>	Fügt die int-Werte, die in der Datei stehen in den Baum ein.
<code>boolean contains(int val)</code>	Testet, ob val im Baum vorhanden ist.
<code>int size()</code>	Ermittelt die Anzahl der Knoten im Baum.
<code>int height()</code>	Ermittelt die Höhe des Baums.
<code>int getMax()</code>	Liefert das größte Element im Baum.
<code>int getMin()</code>	Liefert das kleinste Element im Baum.
<code>boolean remove (int val)</code>	Entfernt val aus dem Baum.
<code>boolean isEmpty()</code>	true genau dann, wenn der Baum leer ist.
<code>void clear()</code>	Entfernt alle Elemente aus dem Baum.
<code>addAll(BinaryTree otherTree)</code>	Fügt alle Elemente des übergebenen Baums (<code>otherTree</code>) in den aktuellen Baum ein.
<code>void printInorder()</code>	Gibt Baum in Inorder aus.
<code>void printPostorder()</code>	Gibt Baum in Postorder aus.
<code>void printPreorder()</code>	Gibt Baum in Preorder aus.
<code>void printLevelorder()</code>	Gibt Baum in Levelorder aus.
<code>void saveToFile (String filename)</code>	Speichert die int-Werte des Baums in der Datei.
<code>BinaryTree clone()</code>	Erzeugt eine tiefe Kopie des Baums. Der Klon sieht exakt so aus wie das Original.

Legen Sie fest (und beschreiben Sie im Kommentar), welche Ergebnisse (Rückgabewerte) herauskommen.

Implementieren Sie in der Klasse `MyBinaryTree` die Funktionen des Interface und zusätzlich noch die folgende statische Methode:

`BinaryTree empty ()` Erzeugt einen leeren neuen Baum.

Um das Testen zu erleichtern, ist die Methode `boolean insert (String filename)` vorgesehen. Dabei werden alle Werte auf der Datei in den Baum eingefügt.

Mit der Methode `void saveToFile (String filename)` können Sie für Testzwecke den aktuellen Baum auf Datei sichern. Außerdem benötigen Sie die beiden Methoden, um die Methode `addAll(BinaryTree otherTree)` realisieren zu können.

Für die Ein- und Ausgabe auf bzw. von Datei verwenden Sie die Methoden `readIntegerArray(String filename)` und `saveIntegerArray(int[] intArray, String filename)`. Das ist etwas umständlich, aber da wir die Java-I/O noch nicht behandelt haben, müssen Sie das über diese beiden Methoden realisieren.

Wenn sie die Einfügeoperation `boolean insert (int val)` rekursiv implementieren, erhalten Sie 5 Zusatzpunkte.

Schreiben Sie ein Rahmenprogramm, das es ermöglicht, über ein geeignetes Menü die einzelnen Operationen auszuführen.

Gehen Sie so vor, dass Sie zuerst die Methode `boolean insert (int val)` implementieren und dann die Methoden zur Ausgabe der Werte des Baums. Dann lassen sich die restlichen Methoden einfacher testen.

Zum Einlesen von Werten und zur Dateiverarbeitung benutzen Sie die entsprechenden Methoden der Klasse `MakeItSimple`.

Denken Sie an Kommentare (in Englisch)!