

Suchbäume

Siehe A10Suchbaemeneu.doc

Binäre Bäume können auch genutzt werden, um Daten geordnet zu speichern. Dabei ist ein Element um so kleiner, je weiter links es steht. Solche Bäume werden häufig Suchbäume genannt, da sie im Vergleich zur Suche in Listen eine effektive Suche eines Elements ermöglichen. Denn wenn ein Element gesucht wird, genügt ein Vergleich mit der Wurzel, um zu entscheiden, ob der linke oder der rechte Teilbaum weiter durchsucht werden muss, wenn die Wurzel nicht schon das gesuchte Element enthält. Es muss maximal einer der Teilbäume geprüft werden. Den anderen Teilbaum braucht man nicht mehr zu berücksichtigen. (Lediglich bei zu Listen entarteten Bäumen ist die Suche nur genauso effektiv bzw. ineffektiv wie die Suche in einer Liste.)

TODO: Beispiel

TODO: ComparableContent

Abstrakte Klassen

In den Bäumen können beliebige Inhalte stehen. Wichtig ist nur, dass eine Ordnungsrelation definiert ist. Die Klasse Object stellt jedoch keine entsprechende Relation (kleiner/größer) zur Verfügung. Deshalb wird für die Implementation eines Suchbaums eine Klasse Item geschrieben, die die Methoden isLess, isEqual, isGreater zur Verfügung stellt. Für eine konkrete Anwendung des Suchbaums müssen die Methoden in einer Unterklasse (z.B. StringElement) zu überschrieben werden. Deshalb wird die Klasse abstrakt definiert.

```
abstract class Item {  
    public abstract boolean isLess(Item pItem);  
    public abstract boolean isEqual(Item pItem);  
    public abstract boolean isGreater(Item pItem);  
}
```

Alle (abstrakten Methoden) einer abstrakten Klasse müssen in einer Unterklasse überschrieben werden, bevor sie in einer Anwendung verwendbar sind.

Aufgaben

Aufgaben:

1. Erstellen Sie die Suchbäume, die entstehen, wenn die folgenden Daten in der angegebenen Reihenfolge eingefügt werden:
 - a) H, D, K, B, G, J, M, R, A, C, N, I, K, S, T
 - b) D, B, I, A, C, H, E
 - c) C, D, A, L, Z, X, Y
 - d) A, B, C, D, E, F, G

TODO

2. In der Anwendung Suchbaum ist die Klasse Item abstrakt. D.h. alle ihre Methoden müssen in einer Unterklasse überschrieben werden, wenn sie verwendet werden sollen. Beschreiben Sie welche Unterschiede zwischen einer abstrakten Klasse und Ihnen aus dem Unterricht bereits bekannten Java-Klassen bestehen.

TODO

3. Erklären Sie, wozu in der Methode

```
public boolean isEqual(Item e) {  
    StringElement hilf;  
    hilf=(StringElement)e;  
    if (((String)inhalt).compareTo((String) hilf.getInhalt())==0)  
        return true;  
    else return false;  
}
```

der Klasse StringElement, einer Unterklasse von Item die Variable hilf verwendet wird. (Es gibt auch eine Alternative ohne diese Hilfsvariable.)

Ergänzen sie die übrigen Methoden der Klasse StringElement.

4. Durch das Einfügen in einen Suchbaum und das anschließende Durchlaufen können Daten auch sortiert werden.
Vervollständigen Sie die Methode XY zur InOrder-Traversierung.
5. Suchbäume können genutzt werden, um die Häufigkeiten von Zeichen in Texten zu ermitteln. Zum Beispiel ist es sinnvoll für Kodierungen und Komprimierungsverfahren zu wissen, welche Zeichen häufig verwendet werden. Wenn die Kodierung der Übermittlung und nicht um Geheimhaltung dient, sollten Buchstaben, die häufig vorkommen einen einfachen Code erhalten.

Schreiben Sie für die Anwendung HaeufigkeitsbaeumeVorlage die Methode TODO, mit der die Häufigkeit des Buchstabens zeichen im Häufigkeitsbaum baum angepasst wird.

Beachten Sie, dass der Buchstabe im Baum evtl. noch gar nicht gespeichert wurde. Implementieren Sie außerdem für die Anwendung HaeufigkeitsbaeumeVorlage die Methode TODO, die den Buchstaben zeichen und dessen im Baum gespeicherte Anzahl zurückgibt, falls er enthalten ist, und sonst "Element nicht enthalten".

TODO:

Quantitative Analyse von Texten

Rap-Texte untersuchen

Sido vs Cro

TODO

<https://de.wikipedia.org/wiki/Huffman-Kodierung>

TODO

<https://de.wikipedia.org/wiki/Shannon-Fano-Kodierung> -> Präfixfreie Kodierung