Ludwig-Maximilians-Universität München WS 2011/12
Institut für Informatik Übungsblatt 14
Prof. Dr. R. Hennicker, G. Beyer, A. Klarl

Übungen zu Einführung in die Informatik: Programmierung und Software-Entwicklung

Aufgabe 14-1 **Binärbäume** Präsenz

Laden Sie von der Vorlesungswebseite die Klassen `BinTree` und `Node` herunter, und implementieren Sie folgende Methoden:

a) In der Klasse `BinTree` ist eine Methode `public Object findElement(int key)` zum Durchsuchen eines Baumes nach einem Knoten mit Schlüssel `key` implementiert. Falls ein Knoten mit einem Wert ungleich `null` vorhanden ist, wird dieser Wert zurück gegeben. Andernfalls wird eine `NoSuchElementException` geworfen. Die Methode ist wie folgt implementiert:

```
1  public Object findElement(int key)
2      throws NoSuchElementException {
3    Object o = root.findElement(key);
4    if (o != null) {
5      return o;
6    }
7    throw new NoSuchElementException("Element fuer key "
8        + key + " nicht gefunden!");
9  }
```

Die Methode greift auf eine gleichnamige Methode `public Object findElement(int key)` in der Klasse `Node` zurück, welche die eigentliche Suche nach dem Knoten mit Schlüssel `key` realisieren soll. Ergänzen Sie die Klasse `Node` um eine entsprechende Methode `public Object findElement(int key)`.

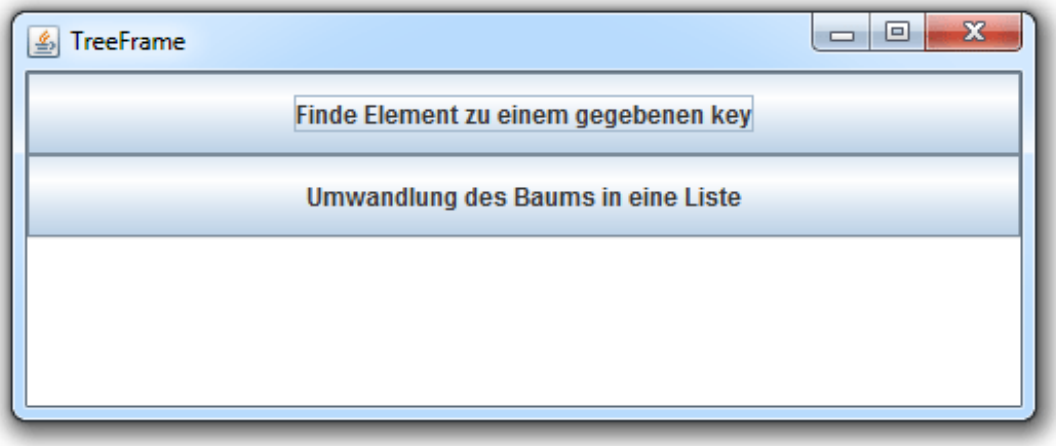
b) In der Klasse `BinTree` ist eine Methode `public LinkedList<Object> toList()` implementiert, welche für einen Baum eine verkettete Liste vom Typ `LinkedList` zurückgibt, die die Werte der Knoten enthält. Die Methode ist wie folgt implementiert:

```
1  public LinkedList<Object> toList() {
2    LinkedList<Object> res = new LinkedList<Object>();
3    if (root != null) {
4      root.buildList(res);
5    }
6    return res;
7  }
```

Die Methode greift auf eine Methode `buildList(LinkedList<Object> res)` der Klasse `Node` zurück, welche die eigentliche Ansammlung von Knotenwerten in einer verketteten Liste `res` realisieren soll. Ergänzen Sie die Klasse `Node` um eine entsprechende Methode `buildList(LinkedList<Object> res)`. Gehen Sie dabei von der Java-Klasse `LinkedList<E>` aus.

c) Implementieren Sie nun ein Programm mit einer grafischen Benutzeroberfläche, welches die Methoden `public Object findElement(int key)` und `public LinkedList<Object> toList()` testet. Der zu testende Baum soll in diesem Programm statisch vorliegen und die Einträge `(2, "Hello")` und (7, "World") haben.

Die grafische Benutzeroberfläche soll wie folgt aussehen:



Es soll zwei Buttons, einen für die Suche eines Elements zu einem gegebenen Schlüssel und einen zum Umwandeln des Baums in eine Liste, geben. Darunter soll ein Ausgabebereich platziert werden, in dem Rückmeldung über die Ergebnisse gegeben wird.

Schreiben Sie eine Klasse `TreeFrame`, die die Hauptklasse dieser grafischen Benutzeroberfläche sein soll und das Fenster erzeugt. Um Ihr Programm ausführen zu können, schreiben Sie eine weitere Klasse `TreeFrameMain`, die Sie wie gewohnt im gleichen Ordner wie Ihre Klasse `TreeFrame` abspeichern.

d) Erweitern Sie Ihre Klasse `TreeFrame` um eine Ereignisbehandlung für den Button für die Suche eines Elements zu einem gegebenen Schlüssel. Wird dieser Button gedrückt, soll der Benutzer in einer Methode `findeElement` mit Hilfe der Klasse `JOptionPane` nach dem Schlüssel des gesuchten Elements gefragt werden. Der Wert des gefundenen Elements soll anschließend im Ausgabebereich angezeigt werden. Falls kein Element mit dem gesuchten Schlüssel im Baum vorhanden ist (und somit eine `NoSuchElementException` geworfen wird) soll das Programm die Fehlermeldung „Ein Element mit diesem Schlüssel ist im vorgegebenen Baum nicht vorhanden.“ ausgeben.

e) Erweitern Sie Ihre Klasse `TreeFrame` um eine Ereignisbehandlung für den Button für das Umwandeln des vorgegebenen Baums in eine Liste der Werte der Knoten des Baums vom Typ `LinkedList<Object>`. Wird dieser Button gedrückt, so soll eine Methode `umwandelnBaum` die Umwandlung des Baums übernehmen und mit Hilfe eines Listeniterators die Werte im Ausgabebereich anzeigen.

Besprechung der Präsenzaufgabe in den Übungen vom 02.02.2012 bis 07.02.2012.