

# Vorlesung „Programmieren“

Wintersemester 2007/2008

## 12. Übungsblatt

### 1. Binäre Suchbäume (4 Punkte)

Fügen Sie folgende Zeichenketten nacheinander in einen anfangs leeren binären Suchbaum ein:  
“Fra”, “Eng”, “Rus”, “Ita”, “Irl”, “Spa”, “Pol”, “Por”, “Lat”, “Swe”.

Entfernen Sie danach “Irl”, “Ita” und “Rus” aus dem Baum.

Zeichnen Sie den Baum nach jeder Änderung. (4 Punkte)

### 2. Telefonbuch mit binären Suchbäumen (Programmieraufgabe) (15 Punkte)

In dieser Aufgabe soll eine Telefonbuch-Anwendung erstellt werden, die einen binären Suchbaum zur Verwaltung der Telefonbuch-Einträge benutzt. Es sollen Name-Telefonnummer-Paare so gespeichert werden, dass die zu einem Namen zugehörige Telefonnummer schnell gefunden werden kann. Die Name-Telefonnummer-Paare sollen in Instanzen der Klasse **Eintrag** abgelegt werden, die Sie in Teilaufgabe b) erstellen sollen. Der Einfachheit halber nehmen wir an, dass Namen eindeutig sind.

Die Klasse **TelefonBuch** ist wie folgt vorgegeben:

```
public class TelefonBuch {  
  
    /* speichert die Telefonbuch-Eintraege */  
    private MyBinaryTree<Eintrag> eintraege;  
  
    public TelefonBuch() {  
        eintraege = new MyBinaryTree<Eintrag>();  
    }  
}
```

Die Klasse **MyBinaryTree** wurde in der Vorlesung vorgestellt und kann von der Web-Seite der Vorlesung (Datei: Abschnitt0403.zip) heruntergeladen werden. Entpacken Sie diese Klassen in ein Verzeichnis Abschnitt0403.

#### Aufgaben:

- Wie werden unter Linux die notwendige Umgebungsvariable für den Java-Übersetzer gesetzt, um bereits übersetzte Java-Klassen aus dem Verzeichnis Abschnitt0403 bei der Übersetzung zu berücksichtigen? (1 Punkt)
- Erstellen Sie eine Klasse **Eintrag**, welche das Interface **Comparable<Eintrag>** implementiert. Ein **Eintrag**-Objekt speichert einen Namen und eine Telefonnummer. Der Vergleich zweier Telefonbuch-Einträge erfolgt über den Namen. (2 Punkte)
- Implementieren Sie eine Methode `boolean insert(String name, String telefon)`, die die gegebene Telefonnummer unter dem gegebenen Namen im Telefonbuch speichert. Die Methode gibt `false` zurück, wenn unter diesem Namen schon eine Telefonnummer gespeichert wurde, andernfalls `true`. (1 Punkt)

d) Implementieren Sie eine Methode `String findTelefon(String name)`, die die Telefonnummer der Person mit dem gegebenen Namen zurückgibt. Falls keine Person mit diesem Namen existiert, wird `null` zurückgegeben. (2 Punkte)

e) Schreiben Sie eine Methode `boolean delete(String name)`, die die unter dem gegebenen Namen gespeicherte Telefonnummer entfernt. Diese Methode gibt `true` zurück, wenn der Eintrag erfolgreich gelöscht wurde. Nach dem Aufruf von `delete(name)` soll `findTelefon(name)` keine Telefonnummer finden. (1 Punkt)

f) Für die nächste Teilaufgabe benötigen Sie eine Möglichkeit, auf alle Elemente eines Suchbaumes zuzugreifen. Erweitern Sie die Klasse **MyBinaryTree** um eine Methode `Comparable[] toArray()`. Das von dieser Methode zurückgegebene Array enthält alle Objekte, die mit Hilfe der Methode `MyBinaryTree.insert(x o)` hinzugefügt (und nicht wieder entfernt) wurden. (2 Punkte)

g) Implementieren Sie eine Methode `String findName(String telefon)`, welche den Namen sucht, der die gegebene Telefonnummer hat. (2 Punkte)

h) Die Beispielklasse **MyBinaryTree** bietet noch keine Möglichkeit, über alle Elemente des Baumes zu iterieren. Erweitern Sie die Klasse **MyBinaryTree** um eine Methode `java.util.Iterator<Comparable> iterator()`, die eine solche Möglichkeit bietet. Der zurückgegebene Iterator braucht die Operation `remove()` nicht zu unterstützen. (4 Punkte)

#### **Bemerkungen:**

- Dieses ist eine Programmieraufgabe. Bitte geben Sie die Lösung *sowohl* auf **Papier** *als auch* den Java-Quelltext in elektronischer Form per **Email** an Ihren Tutor ab.

---

#### **Bemerkungen:**

- Jede Seite soll oben rechts den Namen der Abgebenden und die Übungsgruppennummer (wichtig!) enthalten.
- Lösungen für die Übungsaufgaben sind (in der Regel) zu zweit abzugeben.
- Kommentieren Sie Ihre Lösungen! Besteht eine Lösung aus mehreren Zetteln, so sind diese zusammen zu heften. Bitte keine Hüllen, Mappen, o.ä..
- Bitte schicken Sie *Programmieraufgaben zusätzlich zur Abgabe auf Papier in elektronischer Form per Email* an Ihren jeweiligen Tutor.
- Kommentieren Sie ihren Quelltext bei Programmieraufgaben. Dabei sollen keine Trivialitäten kommentiert werden, also bitte keine Kommentare wie

~~`x=5; // Wir weisen nun der Variablen x den Wert 5 zu`~~

sondern sinnvolle Kommentare, die Ideen des Quelltextabschnittes beschreiben oder auf Unteraufgaben (z. B. a), b), ...) hinweisen.

- **Hinreichende Bedingung für die Zulassung zur Klausur:** 50% der erreichbaren Punkte bei jedem Übungszettel (bis auf zwei) und einmaliges Vorrechnen in der Übung
- **Zertifikatskriterium:** Das Bestehen der Klausur am Ende des Semesters

**Abgabetermin: Freitag, 8.2.2008, nach der Vorlesung**