

## Übungsblatt 12 - Abgabe 24.01.05

*Achtung: Zu lösen sind zwei der folgenden drei Aufgaben.*

### Aufgabe 1 (Dateibäume)

Entwickeln Sie die folgende Java-Methode search:

```
/**
 * Sucht ausgehend vom Verzeichnis "base" nach Dateien (also auch
 * Verzeichnissen), deren Name "search" enthält und liefert deren
 * Pfadnamen relativ zum "base"-Verzeichnis. Groß-/Kleinschreibung wird
 * ignoriert. Die Suche wird rekursiv (d.h. Tiefensuche) mit maximaler
 * Tiefe "depth" in Unterverzeichnissen fortgesetzt, z.B. liefert
 * ein Aufruf mit den Parametern
 *
 * ".", ".txt", 2     eine Menge mit den Elementen
 *
 * urls.txt
 * ablage/diesunddas/notizen.txt
 * ablage/diesunddas/urlaub.txt
 *
 * (. bezeichnet das aktuelle Verzeichnis)
 *
 * @throws FileNotFoundException falls base kein gültiges Verzeichnis
 * @throws IllegalArgumentException falls depth negativ
 * @throws NullPointerException falls base oder search null
 */
static Set<String> search(String base, String search, int depth)
    throws FileNotFoundException;
```

*Hinweis:* API-Dokumentation der Klasse `java.io.File`.

### Aufgabe 2 (Bäume)

Unter <http://www.inf.fu-berlin.de/lehre/WS05/ALP3/uebungen/tree.pdf> finden Sie eine Aufgabe, in welcher Sie ihre Kenntnisse zum Thema Bäume einsetzen können. Verwenden Sie als konkrete Repräsentation des markierten Vielwegbaums eine Implementierung aus der Vorlesung oder versuchen Sie sich an einer eigenen.

Wer Probleme mit dem Einlesen der Daten hat, sollte auf die Vorlagen der Java-Wiederholung zurückgreifen (<http://www.inf.fu-berlin.de/lehre/WS05/ALP3/material/Speed.java>).

### Aufgabe 3 (Praktische Javakenntnisse 4 - Swing)

Falls Sie noch keine Erfahrung mit Java Swing gesammelt haben, sehen Sie sich zuerst die Dokumentation folgender Klassen an:

- JComponent
- JFrame
- JButton
- ActionListener
- Implementationen von LayoutManager - z.B BorderLayout, FlowLayout, GridLayout, etc

Des Weiteren, gibt es auch einige empfehlenswerte Einführungen zu AWT/Swing im Java Tutorial zu finden unter:

<http://java.sun.com/docs/books/tutorial/index.html>.

- a.) Schreiben Sie nun eine Swing-Anwendung, bestehend aus einem JFrame und einem editierbaren Textfeld (experimentieren Sie mit verschiedenen Komponenten), welche sich durch Klicken auf das X-Symbol schließen lässt (passiert nicht automatisch).
- b.) Fügen Sie nun 2 Knöpfe hinzu, die es erlauben, den geschriebenen Text in eine Datei zu speichern, bzw. diesen wieder zu laden. Speichert man auf eine bestehende Datei, soll eine Warnmeldung erscheinen. Im Fenstertitel soll der Name der aktuell geladenen Datei stehen.  
*Tipp:* Für die Dateiauswahl ist die Klasse JFileChooser hilfreich, bei Standard-Dialogen hilft JOptionPane.
- c.) [optional] Um auch längere Texte komfortabel editieren zu können, sollte die Text-Komponente mit Rollbalken versehen werden, sobald der Text zu groß wird; siehe dazu JScrollPane.
- d.) [optional] Erweitern Sie die Anwendung, so dass mehrere Dateien gleichzeitig geöffnet sein können und diese über Reiter erreichbar sind. Siehe dazu die Klasse JTabbedPane.

#### **Abgabe:**

- Relevante Quellcodefragmente mit Angabe der zugehörigen Datei auf Papier.
- Vollständiger Quellcode und ausführbare JAR-Datei per Mail an den Tutor.