

Musterlösung 12

Aufgabe 1 (Dateibäume)

```
static Set<String> search(String base, String search, int depth)
    throws FileNotFoundException, IOException {
    if (depth < 0)
        throw new IllegalArgumentException("negative depth");
    File dir = new File(base);
    if (!dir.exists() || !dir.isDirectory())
        throw new FileNotFoundException("base dir not found");
    TreeSet<String> result = new TreeSet<String>();
    search(dir, dir.getCanonicalPath().length()+1,
        search.toLowerCase(), depth, result);
    return result;
}

/**
 * @param dir    Verzeichnis, das durchsucht werden soll
 * @param prefix Länge des abzuschneidenden Präfix
 * @param search Suchtext
 * @param depth  Rekursionstiefe
 * @param result hier werden die Treffen eingetragen
 */
static private Set<String> search(File dir, int prefix, String search,
    int depth, Set<String> result) throws IOException {
    File[] list = dir.listFiles();
    for (File f : list) {
        if (f.getName().toLowerCase().contains(search))
            result.add(f.getCanonicalPath().substring(prefix));
        if (f.isDirectory() && depth > 0)
            search(f, prefix, search, depth - 1, result);
    }
    return result;
}
```

Aufgabe 2 (Bäume einsetzen)

Pseudo-Code:

- Lese alle Angestellten ein und speichere Abbildung von ID auf Angestellten.
- Konstruiere Baum mittels des Manager-Attributs jedes Angestellten.
- Suche CEO als Wurzel-Knoten.
- Den größten Partywert kann man bestimmen, in dem man für jeden Angestellten folgende zwei Dinge berechnet:
 - Den Partywert, wenn man selbst teilnimmt.
 - Den Partywert, wenn man selbst nicht teilnimmt.
- Diese Werte lassen sich rekursiv aus den Partywerten der Kinder ermitteln.

- Ergebnis ist der Partywert, wenn der CEO teilnimmt.

<http://www.inf.fu-berlin.de/lehre/WS05/ALP3/loesungen/U12.A2.zip>

Aufgabe 3 (Praktische Javakenntnisse 4 - Swing)

Natürlich ließ sich viel aus dem Taschenrechner übernehmen. ;-)

<http://www.inf.fu-berlin.de/lehre/WS05/ALP3/loesungen/U12.A3.jar>