

# Vorgehensweise und Entscheidungsfindung

## class MainApp (main-Methode)

Diese Klasse setzt das Look&Feel auf den des Systems. Danach wird ein neues Objekt der Klasse ExplorerIO erstellt und im Anschluss ein weiteres Objekt der Klasse ExplorerTree, welches das ExplorerIO Objekt übergeben bekommt. Anschließend wird die Methode buildFrame() aufgerufen, die den Frame/FileExplorer aufbaut.

## class ExplorerTree

Zunächst haben wir uns überlegt, wie das fertige Produkt aussehen soll. Zu Beginn hatten wir nur ein einfaches, zweigeteiltes Fenster. Rechts sollten die Dateinfos stehen und auf der linken Seite sollte der Verzeichnisbaum stehen. Realisiert haben wir das ganze mit einer JSplitPane. Bei Verzeichnissen mit viel Inhalt wuchs der Verzeichnisbaum aber über alle Grenzen, um dieses Problem zu beheben, haben wir für den Verzeichnisbaum eine JScrollPane genommen.

Um das Programm schöner zu gestalten und um weitere Swing Komponenten zu üben, haben wir dann das Programm um ein BorderLayout und eine JMenuBar erweitert. Über das Layout können die einzelnen Komponenten einfach arrangiert werden. Über das Dateimenü kann man das Programm beenden oder ein neues Verzeichnis zum Listen wählen. Im Hilfe Menü kann man ein About-Fenster öffnen, welches zwei Bilder und einen Copyrighthinweis enthält, reiner Spielkram ☺

Zunächst hatten wir fast das ganze Programm in einer Methode, um das ganze in kleinere Teile zu zerlegen, haben wir den Aufbau des Frames, des Menüs, des Verzeichnisbaums und des Aboutfensters ausgelagert. Weiterhin gibt es Methoden um die JTextArea auf der rechten Seite mit den Dateinfos zu füllen und die Methode addNodes, die rekursiv aufgerufen wird, um die einzelnen Verzeichnisse und Dateien zu listen.

Die Dateinfos haben wir uns recht einfach ausgegeben mittels eines Stringbuilders. Um die Übersicht und das Aussehen zu verbessern, könnte man das Ganze auch in einen JTable werfen.

## class ExplorerIO

In dieser Klasse gibt es lediglich bisher eine loadDir Methode, die die Selektion eines Verzeichnisses zulässt, welches gelistet werden soll.

Den Aufbau der Dateinfos könnte bei umfangreicherer Programmgestaltung hierher ausgelagert werden.

## class ExplorerUtils

In diese Klasse gehören Hilfskonstrukte für unseren FileLister. Bisher befindet sich hier nur eine statische Methode, um die Dateigröße für die Dateinfos zu formatieren. Diese wurde bereits in einer Aufgabe aus dem vergangenen Semester benutzt, das Grundformat kommt von der Apache Foundation und wurde nur leicht verändert übernommen.

Martin Slowikowski

Matrikelnummer: 199 91 66

Jan-Tristan Rudat

Matrikelnummer: 200 78 52

Teamname: Bernie und Ert

## Quellenangaben

Class FileUtils

The Apache Software Foundation; package org.apache.commons.io

URL: <http://kickjava.com/src/org/apache/commons/io/FileUtils.java.htm>

(abgerufen am 15.10.2010)