

Projekte

1. Masterarbeit am DLR in Braunschweig

- Entwicklung eines interaktiven Editors für Flugzeugkonfigurationen im Vor-entwurf.
- Programmiersprache: Python
- Werkzeuge/Methoden: Eclipse, Qt, OpenGL, CPACS, TiGL, TiXI, GIT, SVN, PyUnit, Doxygen, Checkstyle
- Dauer: 12 Monate

2. Bachelorarbeit

- Identifikation von fehlerbehafteten Daten zur energiebewussten Programmierung
- Programmiersprache: Java
- Werkzeuge/Methoden: Eclipse, Hadoop Framework, RMI, JMX, Javassist, SVN
- Dauer: 2,5 Monate

3. Projektarbeit Softwareevolution

- Weiterentwicklung von JabRef, einem Literaturverwaltungssystem für \LaTeX (Bibtex)
- Programmiersprache: Java
- Werkzeuge/Methoden: Eclipse, SVN
- Gruppenarbeit von 3 Personen
- Dauer: 5 Monate
- Einarbeitung in die Anwendung
- BugFixes
- Erweiterung der Anwendung um ausgewählte Features (Preview-Fenster etc.)

4. Projektarbeit Softwarepraktikum

- Implementierung des Spiels *Einfach Genial*
- Programmiersprache: Java
- Werkzeuge/Methoden: Eclipse, SVN
- Gruppenarbeit von 4 Personen
- Dauer: 5 Monate
- Entwicklung der Spiellogik, GUI, KI und Netzwerkanbindung

5. Fortgeschrittenenpraktikum

- Tool zur Visualisierung von Ressourcenauslastung und Konfiguration eines virtualisierten Rechnerclusters
- Programmiersprache: Java, Lua
- Werkzeuge/Methoden: Eclipse, GIT, Gephi, LibVirt, Xen
- Gruppenarbeit von 2 Personen
- Dauer: 1 Jahr
- Ein wesentlicher Unterschied zu anderen Visualisierungsanwendungen ist die Darstellung der Daten in einer Graphen-Struktur anstatt der sonst häufig genutzten Tabellenform. Durch ein einfaches Treiber-Konzept können einzelne Quellen separat entwickelt und an die Anwendung gekoppelt werden.

6. Praktikum bei echternacht new media (Tipp-Spiel)

- Erweiterung des internen Tipp-Tools
<http://bundesliga.pbc.echternacht.com/public/10/welcome.do>
- Programmiersprache: Java, Webservice von OpenLigaDB
- Implementierung einer automatischen Ermittlung der Spielergebnisse (wurde vorher vom Admin nach den Spielen manuell eingetragen)
- Kleinere Änderungen, z.B Konto löschen oder Zitat-Box (von Fußballern)

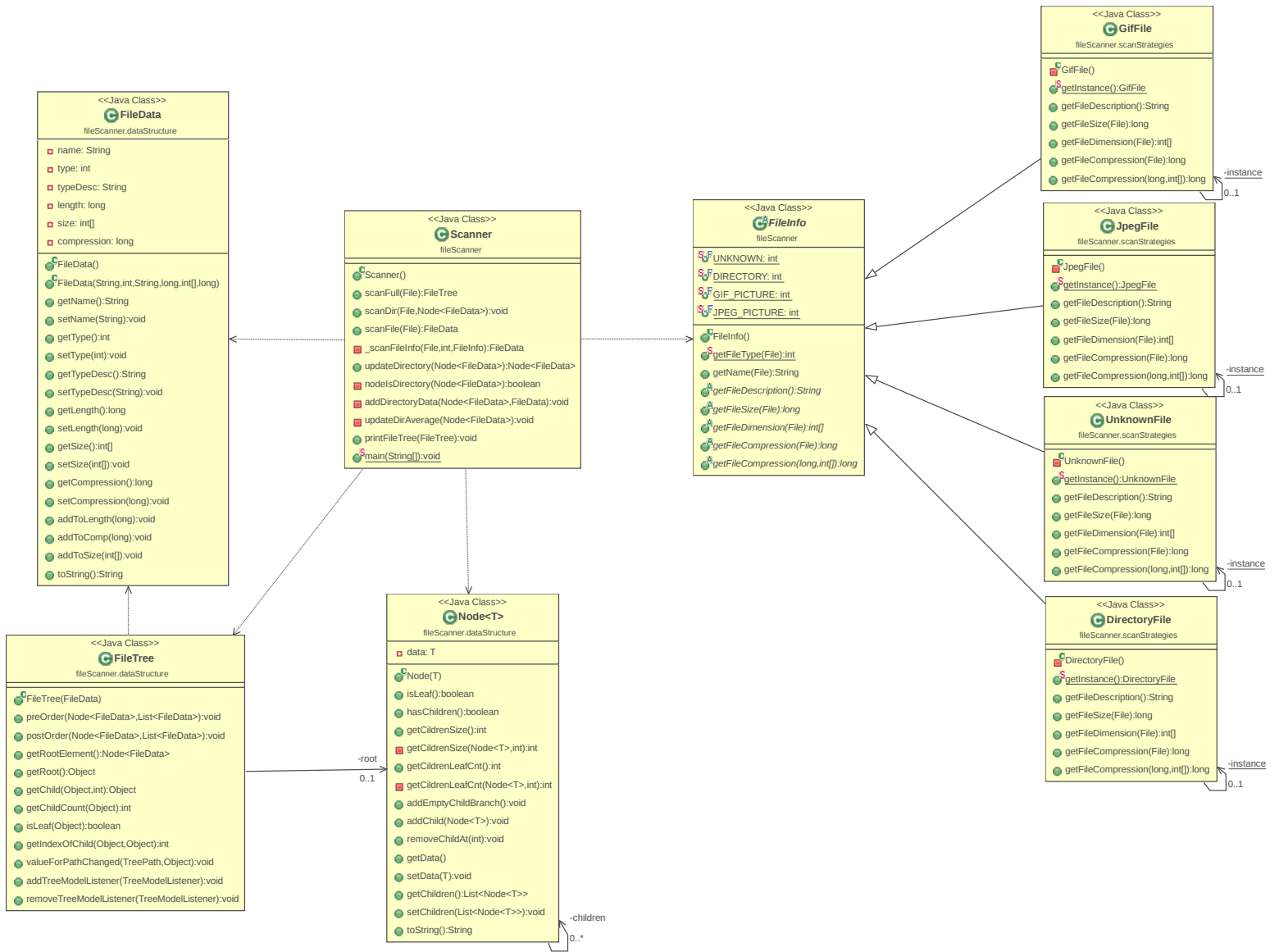
7. Praktikum bei echternacht new media (Datenanalyse)

- Entwicklung einer Anwendung zur Auswertung und Manipulation größerer Datenbestände
- Programmierung: Excel, VBA, Graphviz
- Aufgabe war das Auslesen von CSV-Dateien und die Aufbereitung in einer Excel-Anwendung mit verschiedenen Bearbeitungs- und Darstellungsfunktionen. Zudem sollten gewählte Konfigurationen mit Graphviz visualisiert werden.

8. echternacht new media (Webanwendungen der Deutschen Bank)

- Implementierung von Webanwendungen der Deutschen Bank (z.B. BAföG-Rechner)
- Programmierung: Java
- Werkzeuge/Methoden: Eclipse, SVN, Hibernate, JDBC, Maven, JUnit, Selenium
- Mitarbeiter: 6
- Dauer: 1 Jahr (2-3 in der Woche)
- hauptsächlich Bugfixes sowie JUnit- und Selenium-Tests

UML



Kurze Dokumentation

Ausführung

Der Ordner JavaTest kann als Projekt in Eclipse importiert werden. Danach einfach die main-Methode in der Scanner-Klasse ausführen. Die jar-Datei öffnet eine minimalistische Benutzeroberfläche, in der ganz intuitiv eine Datei bzw. ein Verzeichnis für den Scan ausgewählt werden kann.

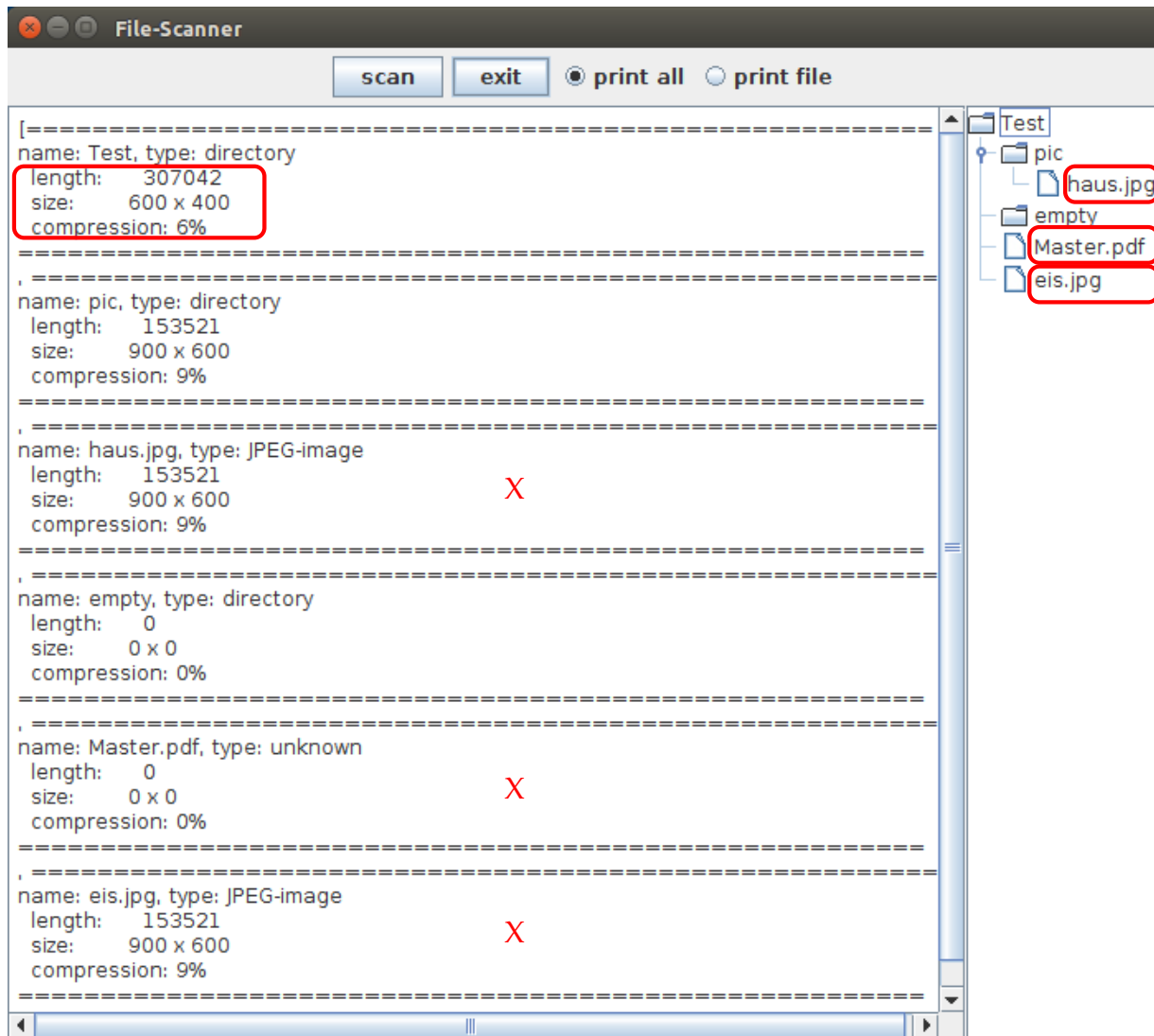


Abbildung 1: GUI-FileScanner

Die einzelnen Elemente sind anschließend in einem Verzeichnisbaum auswählbar und deren jeweilige Eigenschaften werden bei einer Selektierung in das Ausgabefeld

geschrieben. Für die Directories und Sub-Directories wurden jeweils die Komponenten *length*, *size* und *compression* anhand der darin enthaltenen Dateien berechnet. Dabei wurden ebenso unbekannte Dateien miteinbezogen. In Abbildung 1 soll dies anhand eines Beispiels verdeutlicht werden. Das rot markierte Feld in der linken, oberen Ecke zeigt die Berechnung für den obersten Ordner *Test*. Diese Berechnung resultiert aus den drei markierten Dateien auf der rechten Seite (Abb. 1), da diese irgendwo im Ordner *Test* liegen. Mit dem roten X sind die Eigenschaften der einzelnen Dateien markiert. Für *length* wurden nun alle *length*-Werte der Dateien aufsummiert. Für *size* und *compression* wurde jeweils der Durchschnitt aus allen drei Dateien genommen.

Scanner

- Steuert alle Prozesse
- Führt den Scan einer Datei durch und baut die interne Datenstruktur auf

FileInfo

- Abstrakte Oberklasse zur Beschaffung von Dateiinformationen
- Wird von der Scanner-Klasse für die jeweiligen Dateiscans verwendet
- Stellt abstrakte Methoden bereit, die jede spezifische Unterklasse (File-Strategy) für deren Dateityp implementieren muss.
- Strategien sind: GifFile, JpegFile, UnknownFile, DirectoryFile

FileData

- Ein Objekt dieser Klasse verwaltet alle angegebenen Informationen einer Datei
- FileData-Objekte werden in der internen Datenstruktur angelegt

FileTree & Node<T>

- Eine Datenstruktur, die das eingelesene Dateisystem als Baum speichert

Zeit

Alles zusammen (Code, Tests, Doku, UML) ca. 6 Stunden.