

Manfred Schappe
Matrikel-Nr.: 13668
U-Nummer: 22242
Hochschule Harz (FH)
Fachbereich AI
Studiengang Medieninformatik 07

**XML to HTML
PlaylistConverter**



Programmierung 3 Semesterprojekt

Wintersemester 2008/2009
Prof. J. K. Singer, Ph.D. (USA)

Aufgabenstellung

Die Aufgabe besteht in der Implementierung einer Softwareapplikation. Es soll ein Java-Programm erstellt werden, welches aus einer iTunes-Playlist, also einer XML-Datei, die Musikstücke ausliest.

Diese sollen anschließend in ein selbst entwickeltes XML-Format umstrukturiert werden, welches gegen ein ebenfalls selbst erstelltes XML-Schema verifiziert werden muss. Das von iTunes genutzte XML-Format besteht nur aus Schlüssel-Wert-Paaren und sollte so nicht benutzt werden. Die neuen XML-Tags sollen Artisten, Musikstücke usw. identifizieren.

Des Weiteren soll eine Transformationen erstellt werden, welche aus den im neu erstellten XML-Format gespeicherten Daten eine Website generiert. Die Web-Site sollte mit allen gängigen Browsern darstellbar sein und kann auch aus mehr als einer HTML-Datei bestehen.

Auf der Website sollen die Musikstücke sortiert dargestellt werden. Die Art und Weise der Webdarstellung soll vom Benutzer in dem Java-Programm durch Auswahlmöglichkeiten beeinflusst werden können. Falls möglich, können auch die entsprechenden Album-Covers auf der Web-Seite dargestellt werden.

Definition der Kernfunktionen

Nach Analyse der Aufgabe ergeben sich folgende Funktionen, die das Java-Programm mindestens zur Aufgabenerfüllung beinhalten muss:

User Interface

Es muss über ein GUI verfügen, das dem User die Bedienung ermöglicht. Die Erstellung kann über Java Swing erfolgen. Das Interface sollte dem Nutzer ermöglichen, die zu transformierende XML-Datei auszuwählen, der Einfachheit halber würde sich hier ein Dateiauswahl-Dialog (FileChooser) anbieten. Zusätzlich muss der Nutzer die Möglichkeit haben, das Sortierkriterium (Interpret o.ä.) festlegen. Dies könnte mittels eines Dropdownfeldes oder aber Checkboxen in Kombination mit entsprechendem ActionListener realisiert werden. Der Start der Transformation sollte dann durch einen einfachen Klick (ActionListener) auf einen entsprechenden Button erfolgen.

Einlesen der XML-Datei

Die vom User benannte XML-Datei muss geöffnet werden. Die enthaltenen Daten müssen extrahiert und zunächst temporär gespeichert werden. Zur

Speicherung werde ich wohl ein Array verwenden.

Umstrukturierung der Daten

Die gespeicherten Daten müssen in das neue, umstrukturierte XML-Format, überführt werden. Dies kann durch Textoperationen der Array-Einträge erfolgen. Zusätzlich muss ein neuer XML-Tag zum Festlegen des Sortierkriteriums den Daten hinzugefügt werden.

Ausgabe der XML-Datei

Die im Array gespeicherten, umstrukturierten Daten müssen in eine neue XML-Datei geschrieben und die Datei anschließend gespeichert werden. Hierzu muss entweder eine neue Datei erstellt oder eine bereits vorhandene überschrieben werden.

Ausgabe der HTML-Datei

Nun muss eine vorhandene HTML-Datei geöffnet bzw. eine neue Datei erstellt werden, in welche die entstandenen XML-Daten anhand der XSL-Formatierung geschrieben werden.

Klassen und Methoden

Nach dieser ersten Analyse der Kernfunktionen können folgende benötigte Klassen und ihre Schlüsselmethoden abgeleitet werden:

- Main:**
- zum Start der GUI / Programmstart
 - Aufruf der „Interface“-Klasse
- Interface:**
- grafische Darstellung für Benutzer mittels Java Swing
- Input:**
- Öffnen der Input-XML-Datei
 - Einlesen der enthaltenen Daten
 - Umstrukturierung der Daten
 - Speichern in temporärem Array
- Output:**
- Erzeugen / Ändern der neuen XML-Datei
 - Erzeugen / Ändern der neuen HTML-Datei anhand XML-Daten und XSL-Datei

Eventuell können einige Methoden der Input- und Output-Klasse direkt in der Interface-Klasse an den ActionListener des Start-Buttons angefügt werden.