

## In diesem Kapitel

- Um was geht's?
- Wer sind Sie?
  - Was Sie bereits wissen sollten.
  - Ihre Arbeitsumgebung.
- Ihr Vorgehen beim Durcharbeiten
- Notation

## Einleitung

### 1.1. Um was geht's?

Unterschiedliche Personen haben unterschiedliche Kenntnisse und verstehen damit auch Konzepte unterschiedlich.

Hier ein Beispiel:

- 1) Aus Sicht eines Datenbank-Spezialisten sind XML Dokumente einfache Tabellen.
- 2) Aus Sicht des OO-Programmierers sind XML Dokumente Objekte und DTD's Klassen.
- 3) Aus Sicht des Web-Designers sind XML Dokumente zusammen mit Stylesheets eine mögliche Weiterentwicklung von HTML.
- 4) Aus Sicht des Theoretikers sind XML Dokumente S-Ausdrücke.

Aber :

- 1) XML ist nicht HTML;  
obschon Web Seiten in XML beschrieben werden können.
- 2) XML Dokumente speichern Informationen;  
sie sind aber keine Datenbanken.
- 3) XML kann Objekte serialisieren;  
sie sind aber keine Objekte.
- 4) XML kann funktionale Programmiersprachen beschreiben;  
aber XML ist keine Programmiersprache.
- 5) Bücher und Dokumente kann man mithilfe von XML beschreiben;  
aber XML ist keine Publishing Sprache.

XML ist ein selbständiges Informatik-Werkzeug und kann nicht ohne Abstriche in eines der gängigen Informatik-Konzepte (Objekte, Klassen, Seiten-Beschreibungssprache) eingeordnet werden.

Deswegen werden wir im Folgenden XML mit mächtigen speziellen API's beschreiben und analysieren:

- SAX – das Simple API for XML
- DOM – das Document Object Model
- JDOM – ein Java API für XML

# XML UND JAVA

Zuerst werden wir uns mit den Grundlagen, wenn auch nur oberflächlich, beschäftigen:

- XML Syntax
- DTD's, XML Schema und XML Validierung
- XPath
- XSLT und das TrAX API
- JAXP (eine Kombination von SAX, DOM, TrAX und einigen Factory Klassen)

Zudem werden wir umfangreichere Konzepte wie XML-RPC, SOAP und ähnliche Konzepte kennen lernen.

## **1.2. Wer sind Sie**

Sie sollten bereits Java kennen. Sie haben sich zum Ziel gesetzt, XML in Ihre Java Applikationen zu integrieren:

- Die meisten XML Werkzeuge sind selber in Java geschrieben.
- Sie finden eine Unzahl Open Source Werkzeuge für XML (in Java).
- XML wird überwiegend mithilfe von Java verarbeitet.

In diesem Kurs lernen Sie:

- XML Dokumente aus Java Applikationen heraus zu speichern.
- XML Dokumente zu lesen.
- XML Dokumente zu durchsuchen, abzufragen und zu mutieren.
- „flache“ Dateien in hierarchische XML Dokumente umzuwandeln.
- Mit Netzwerk-Servern zu kommunizieren und mit diesen XML Daten zu senden und zu empfangen.
- XML Dokumente gegenüber DTD's oder XML Schemas zu validieren
- XSLT Transformationen mit Java zu kombinieren.

In Beispielen lernen Sie die allgemeinen Konzepte konkret anzuwenden.

### **1.2.1. Was Sie bereits wissen sollten**

Da wir einiges an XML und Java voraussetzen, sollten Sie bereits Grundkenntnisse in Java und XML haben. Speziell Java sollten Sie bereits recht gründlich kennen:

- Objekt-orientierte Programmierung, inklusive Vererbung und Polymorphismus.
- Java Packages : Einsatz von Packages, auch Fremdpackages.
- CLASSPATH : Einbinden unterschiedlichster Archive
- Java I/O : Readers, Writers und Streams.
- Java Collections API : Hash-Tabellen, Maps, Iteratoren und Listen.

Zudem werden wir SQL und JDBC gelegentlich einsetzen, falls es sich anbietet.

## 1.2.2. Ihre Arbeitsumgebung

XML ist noch Plattform-unabhängiger als Java: Sie können XML mithilfe von Perl, C++, ... bearbeiten. Interne Formate, LSB (Least Significant Bit) und ähnliche Konzepte, spielen beim Einsatz von XML kaum eine Rolle. Sie können XML auf unterschiedlichsten Hardware- und Software-Plattformen einsetzen und diese beliebig miteinander kombinieren.

Wir werden uns auf JDK ab 1.4 beschränken. Diese Version umfasst bereits XML Basisklassen. Aber Sie können problemlos Apache Klassenbibliotheken mit älteren JDK Version kombinieren.

## 1.3. *Ihr Vorgehen beim Durcharbeiten*

In den ersten Kapiteln befassen wir uns mit den Grundlagen von XML, SOAP, XML-RPC und ähnlichen Konzepten. Dann werden wir direkt aus Java heraus XML Dokumente generieren, plump und ohne Ehrgeiz.

Anschliessend werden wir uns mit den speziellen API's befassen:

- Dem ereignisgesteuertem SAX API.
- Dem baumartigen DOM API.
- Dem baumartigen JDOM API.
- Dem XPath API, mit dessen Hilfe wir XML Dokumente durchsuchen können.
- Dem TrAX API, mit deren Hilfe wir XSLT durchführen können.

Falls Sie eine Kurzeinführung in XML benötigen, sollten Sie die ersten fünf Kapitel der Reihe nach durcharbeiten – als aktiv am PC mitarbeiten!

- Kapitel 6 – 8 befassen sich mit SAX
- Kapitel 9 – 13 befassen sich mit DOM
- Kapitel 14 und 15 befassen sich mit JDOM
- Kapitel 16 und 17 befassen sich mit XPath und XSLT, setzen also Kenntnisse der andern API's voraus.

## 1.4. Notation

Folgende Konventionen sind allgemein gültig und werden auch hier möglichst eingehalten:

- *Kursiver Text*  
wird eingesetzt, um Text speziell hervorzuheben.
- Quellcode  
wird eingesetzt, um Programmcode, URL's und System-Meldungen zu kennzeichnen.
- *Kursiver „Quelltext“*  
beschreibt Platzhalter für Quelltext.
- **Fetter Text**  
dient der üblichen Hervorhebung von Text.
- **Fetter Text**  
dient der Beschreibung von Eingaben durch den Benutzer.

Oft werden die Ausgaben der Programmbeispiele so umformatiert, dass der Text besser lesbar wird:

```
result.append("          <Amount>" + amount + "</Amount>\r\n");
```

Damit dieser Text besser lesbar wird, werden wir in der Regel die Ausgabe auf mehrere Zeilen aufteilen :

```
result.append("          <Amount>");  
result.append(amount + "</Amount>\r\n");
```

## 1.5. Persönliche Bemerkungen

Sie sollten nicht zögern, mich zu kontaktieren, falls Sie Fehler finden, der Text unverständlich wird oder Sie Fragen haben. Sie finden meine Kontaktadresse auf dem Web, speziell meine aktuellen email Adressen.