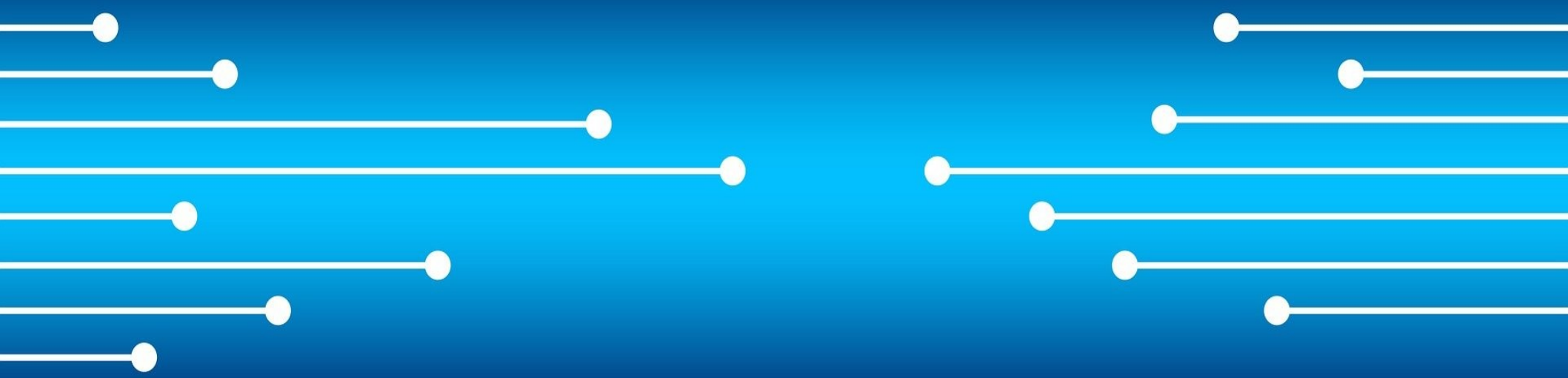


Datenverarbeitung

Teil des Moduls 5CS-DPDL-20



Prof. Dr. Deweß

Thema 10



10

Java und XML

- Was ist XML?
- Wie kann man XML-Dateien in Java einlesen?
- Wie kann man XML-Dateien mit Java erzeugen?

Was ist XML?

- Abkürzung für „eXtensible Markup Language“, ein vom W3C 1998 eingeführter Standard zur Dokumentenauszeichnung – als Mittelding zwischen der „einfachen“ Auszeichnungssprache HTML und der komplexen Auszeichnungssprache SGML (Standard Generalized Markup Language)
- ist inzwischen (viele Erweiterungen und Ergänzungen zum ursprünglichen Standard) quasi eine XML-Technologie-Familie
- ist im Unterschied zu HTML eine Meta-Sprache, bei der:
 - zwar auch Tags und Attribute verwendet werden, bei der jedoch die Menge und Bedeutung der Tags nicht global definiert sind, sondern durch den Anwender selbst festgelegt werden können
 - durch inhaltliche und strukturelle Festlegungen im Rahmen einer DTD (Document Type Definition) oder eines XML-Schemas weitere, konkrete Sprachen definiert werden können (z.B. XHTML, GPX, RSS, XAML)
- eignet sich sehr gut für die Speicherung von Daten in Textform
- wird oft zum plattformübergreifenden Datenaustausch benutzt und kann sowohl von Menschen als auch Maschinen relativ gut gelesen werden

Wie sind XML-Dokumente aufgebaut?

XML-Dokument besteht aus

- optionaler Kopf mit optional
 - XML-Deklaration mit Attributen
 - „version“: XML-Version
 - „encoding“: benutzte Zeichenkodierung
 - „standalone“
 - Deklaration, die den Dokumenttyp festlegt (DTD)
- Daten
 - Daten werden hierarchisch in Textform zwischen „Tags“ gespeichert
 - dadurch entstehende logische Komponenten heißen „Elemente“, das alles umschließende Element ist das „Wurzelement“
 - Starttags und Leertags können über Attribute zusätzliche Eigenschaften erhalten
- Kommentare und Ausführungsanweisungen

Wie sind XML-Dokumente aufgebaut?

Beispiel: buecher.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book>
    <title>Der Gröffelo</title>
    <author>Axel Scheffler, Julia Donaldson</author>
    <year>1999</year>
    <price>13.95</price>
  </book>
  <book>
    <title>Das Gröffelokind</title>
    <author>Axel Scheffler, Julia Donaldson</author>
    <year>2007</year>
    <price>9.95</price>
  </book>
</bookstore>
```

XML-Deklaration

Starttag

Endtag

Groß-/Klein-
schreibung
relevant

Wurzelement



XML und Java: Einlesen von XML-Dokumenten

„Parser“ – Was ist ein „Parser“?

- Programm, welches die grammatikalische Struktur eines Eingabe-Dokumentes im Hinblick auf eine gegebene formale Grammatik analysiert
- bestimmt ausgehend von der gegebenen Grammatik durch Angabe möglicher kleinster Elemente und deren möglicher Beziehungen zueinander, wie ein Satz aufgebaut sein kann
- ein **XML-Prozessor („XML Parser“)** stellt für das Einlesen von XML-Dokumenten auf unterster Ebene eine Programmierschnittstelle (API) für den Zugriff einer Anwendung auf ein XML-Dokument zur Verfügung

XML und Java: Einlesen von XML-Dokumenten

„XML Parser“ – Modelle der bekanntesten, grundlegenden Schnittstellen

- „Simple API for XML“ (SAX)
 - ereignisorientiert
 - sequentielle Stream-Repräsentation (Datenmodifikation so nicht einfach möglich, da Daten nur im Moment des Lesens zur Verfügung stehen) eines XML-Dokuments
- „Document Object Modell“ (DOM)
 - objektorientiert
 - wahlfreier Zugriff durch Baumstruktur (speicherintensiv; Dokument ist damit aber gut änderbar, da der ganze Baum zur Verfügung steht und am Ende zurückgeschrieben werden kann) zur Repräsentation eines XML-Dokumentes
- **weitere Verarbeitungsmöglichkeit:**
über „XML Data-Binding“: Methoden zur Transformation von XML-Dokumenten in das Objektmodell einer Programmiersprache (z.B. Java) und umgekehrt

XML und Java: Beispiel mit DOM-Parser

Beispiel: buecher.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
  <book>
    <title>Der Grüffelo</title>
    <author>Axel Scheffler, Julia Donaldson</author>
    <year>1999</year>
    <price>13.95</price>
  </book>
  <book>
    <title>Das Grüffelokind</title>
    <author>Axel Scheffler, Julia Donaldson</author>
    <year>2007</year>
    <price>9.95</price>
  </book>
</bookstore>
```

DOM-Baumstruktur



XML und Java: Beispiel mit DOM-Parser

- DOM-Baumstruktur wird durch einen „DocumentBuilder“ erzeugt
- dieser „DocumentBuilder“ wird über eine „Factory“ erzeugt, um implementierungsunabhängig zu sein
- die „Factory“ wird abhängig von der Systemkonfiguration gewählt

```
// DocumentBuilderFactory erzeugen
DocumentBuilderFactory factory =
    DocumentBuilderFactory.newInstance();

// DocumentBuilder erzeugen
DocumentBuilder builder =
    factory.newDocumentBuilder();

// Parsed die XML-Datei und
// erzeugt ein Document-Objekt
Document doc =
    builder.parse("buecher.xml");
```

DOM-Baumstruktur



XML und Java: Beispiel mit DOM-Parser

- DOM-Baumstruktur wird durch einen „DocumentBuilder“ erzeugt
- dieser „DocumentBuilder“ wird über eine „Factory“ erzeugt, um implementierungsunabhängig zu sein
- die „Factory“ wird abhängig von der Systemkonfiguration gewählt

Nun kann jeder Knoten des DOM-Baumes näher betrachtet werden, wobei Java dafür viele Methoden bereitstellt.

DOM-Baumstruktur



XML und Java: XPath

- für strukturierte, datenbezogene Anfragen kann man eine XML-Anfragesprache nutzen
- „XML Path Language“ (XPath)
 - dient dem flexiblen Zugriff auf Teile eines XML-Dokuments, indem es die Auswahl einer Knoten-Teilmenge ermöglicht
 - kann von anderen Zugriffsmöglichkeiten wie XQuery, XSL oder XPointer genutzt werden

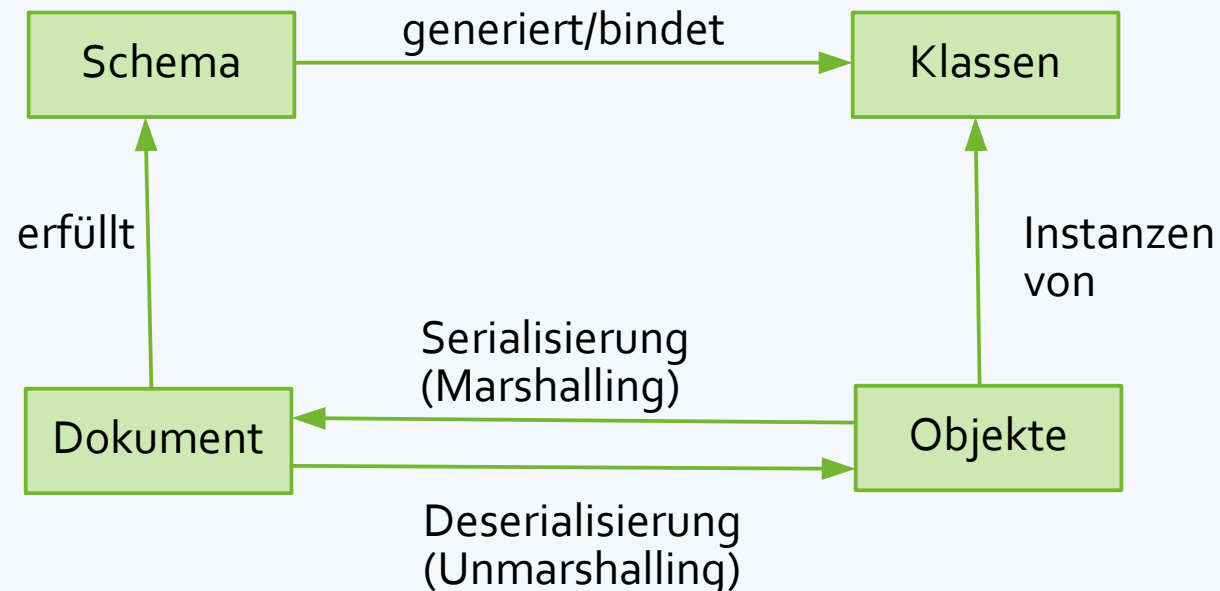
```
XPathFactory xPathFactory =  
    XPathFactory.newInstance();  
XPath xPath = xPathFactory.newXPath();  
String expression  
    = "//book[year > 2000]/title";  
  
// XPath-Ausdruck auswerten  
NodeList nodeList = (NodeList)  
    xPath.compile(expression).evaluate(document,  
    XPathConstants.NODESET);
```

DOM-Baumstruktur



XML und Java: XML Data-Binding

- Mittel zur Darstellung von Informationen in einem XML-Dokument als Objekte
- ermöglicht Anwendungen den Zugriff auf die Daten über Objekte anstatt des Datenabrufs über DOM oder SAX (oder direkt aus dem XML-Dokument)
- es gibt für Java dafür mehrere Frameworks bzw. allgemein Bibliotheken:
 - Jakarta Architecture for XML Binding (JAXB)
 - Apache XMLBeans
 - Xstream
 - Castor
 - JiBX
 - ...



XML und Java: XML Data-Binding mit JAXB

„Jakarta XML Binding“ (JAXB) [ehemals: Java Architecture for XML Binding]

- Schnittstelle, um XML-Daten in Objekte zu überführen und umgekehrt
- zur Benutzung wird eine JAXB-Implementierung benötigt, die folgende Komponenten beinhaltet:
 - „schema compiler“:
bindet ein XML-Schema an eine Menge davon abgeleiteter Programmelemente
 - „schema generator“:
bildet eine Menge von Programmelementen auf ein XML-Schema ab
 - „binding runtime framework“:
stellt „marshalling“- Operationen (Operationen, um aus einem Baum von Java-Objekten ein XML-Dokument zu erstellen [spez. Form der Serialisierung]) und „unmarshalling“-Operationen (Operationen, um aus einem XML-Dokument einen Baum aus Java-Objekten zu erzeugen [spez. Form der Deserialisierung]) für Zugriff, Validierung und Veränderung von XML-Inhalten zur Verfügung

Java und XML: JAXB - Vorbereitung

1. JAXB-Dateien (seit Java SE 11 nicht mehr standardmäßig enthalten) besorgen:
z.B.: <https://mvnrepository.com/artifact/com.sun.xml.bind/jaxb-ri/4.0.2>

← → ↻ <https://mvnrepository.com/artifact/com.sun.xml.bind/jaxb-ri/4.0.2>

Lesezeichen importier... Serialization and Deser... Outlook ChatGPT Erste Schritte Wget Wizard - Whatls... Einfue

MVN REPOSITORY

Search for groups, artifacts, categories

Indexed Artifacts (33.6M)

Projects (millions)

Year

Popular Categories

- Testing Frameworks & Tools
- Android Packages
- Logging Frameworks
- Java Specifications
- JSON Libraries
- JVM Languages
- Core Utilities

Home » com.sun.xml.bind » jaxb-ri » 4.0.2

JAXB RI » 4.0.2

JAXB RI standalone zipped bundle

License	EDL 1.0
Tags	binding jaxb xml
HomePage	https://eclipse-ee4j.github.io/jaxb-ri/
Date	Feb 06, 2023
Files	pom (5 KB) View All
Repositories	Central
Ranking	#95388 in MvnRepository (See Top Artifacts)
Used By	4 artifacts

Maven Gradle Gradle (Short) Gradle (Kotlin) SBT Ivy Grape Leini

„View all“
anklicken

Java und XML: JAXB - Vorbereitung

1. JAXB-Dateien (seit Java SE 11 nicht mehr standardmäßig enthalten) besorgen:
z.B.: <https://mvnrepository.com/artifact/com.sun.xml.bind/jaxb-ri/4.0.2>

com/sun/xml/bind/jaxb-ri/4.0.2

../		
jaxb-ri-4.0.2.pom	2023-01-27 06:36	5803
jaxb-ri-4.0.2.pom.asc	2023-01-27 06:36	833
jaxb-ri-4.0.2.pom.md5	2023-01-27 06:36	32
jaxb-ri-4.0.2.pom.sha1	2023-01-27 06:36	40
jaxb-ri-4.0.2.zip	2023-01-27 06:36	3881791
jaxb-ri-4.0.2.zip.asc	2023-01-27 06:36	833
jaxb-ri-4.0.2.zip.md5	2023-01-27 06:36	32
jaxb-ri-4.0.2.zip.sha1	2023-01-27 06:36	40

Datei „jaxb-ri-4.0.2.zip“
herunterladen

Java und XML: JAXB - Vorbereitung

2. JAXB-Dateien (eben heruntergeladene API und Implementation) verfügbar machen:
 - a) Bei Verwendung von „Eclipse“ ohne weitere Build-Tools (sonst ggf. andere Dateistruktur benutzen):
im verwendeten Java-Projekt im Arbeitsverzeichnis neben „src“ und „bin“ ein Verzeichnis „lib“ anlegen
 - b) „zip“-Datei entpacken
 - c) „*.jar“-Dateien aus dem Verzeichnis „mod“ ins unter a) angelegte „lib“-Verzeichnis kopieren
 - d) Erforderliche Abhängigkeiten im Projekt hinzufügen – bei „Eclipse“:
 1. Projekt mit rechter Maustaste anklicken
 2. „Erstellungspfad“ → „Erstellungspfad konfigurieren...“ auswählen
 3. Tab „Bibliotheken“ auswählen
 4. „Classpath“ auswählen
 5. „Externe JARs hinzufügen...“ auswählen
 6. Alle „*.jar“-Datei im „lib“-Ordner markieren und „Öffnen“ auswählen
 7. „Apply and Close“ anklicken

Java und XML: Aufgaben

Achtung, es wird
dazu keine Lösung
bereitgestellt

Laden Sie sich für die folgenden Aufgaben von OPAL die Datei „XMLAufgaben.zip“ herunter und entpacken Sie diese.

Aufgabe 1

Integrieren Sie die in der zip-Datei enthaltenen java-Dateien in eines Ihrer Java-Projekte, verschieben Sie ggf. andere dafür benötigte Dateien in Ihr Java-Arbeitsverzeichnis und passen Sie die java-Datei so an, dass die Klasse „XMLTestClass.java“ bei Ihnen lauffähig wird. Beachten Sie dabei, dass für die Verwendung von JAXB noch einige Vorbereitungen nötig sind.

Aufgabe 2

Erweitern Sie die gegebene XML-Datei um mindestens ein weiteres Buch.

Aufgabe 3

Bisher wird mittels JAXB eine ganze Bücherliste in ein XML-Dokument geschrieben. Versuchen Sie, ein XML-Dokument zu erzeugen, in dem nur ein Buch enthalten ist und entsprechend „book“ als Wurzelement benutzt wird.

Zusatzaufgabe: Schaffen Sie es, dass eine weitere Bucheigenschaft „publisher“ mit verarbeitet wird und ergänzen Sie die gegebene XML-Datei entsprechend.

Vielen Dank für Ihre
Aufmerksamkeit.

