

UNIVERSITÄT LEIPZIG
LPZ E-BUSINESS
applied telematics

Übungen zum Java-Praxiskurs für Fortgeschrittene

Matthias Book
Malte Hülдер

Lehrstuhl für Angewandte Telematik / e-Business WS 2005/2006

UNIVERSITÄT LEIPZIG
LPZ E-BUSINESS
applied telematics

Wiederholung: XML und JDOM

- XML-Dokumente
 - strukturierte Speicherung von Daten im Textformat
 - müssen wohlgeformt, sollten valide sein
 - Validitätsregeln in Document Type Definition (DTD) spezifiziert
 - Lese-Strategien: DOM (Baum im Speicher), SAX (Element-Events)
- JDOM
 - Klassenbibliothek zur Arbeit mit XML-Dokumenten
 - Einlesen in JDOM-Repräsentation z.B. mit **SAXBuilder**
 - Objekte zur Arbeit mit der JDOM-Repräsentation des Dokuments:
 - Document** enthält **Content**-Abschnitte
 - Contents** sind **Elemente**, **Texte**, etc.
 - Elemente** können wieder **Content**-Abschnitte enthalten
 - Elemente** können **Attributes** haben
 - Ausgabe mit **XMLOutputter**

Der Java-Praxiskurs II 2

UNIVERSITÄT LEIPZIG
LPZ E-BUSINESS
applied telematics

Document Type Definition

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT bestand (buch*)>
<!ELEMENT buch (titel, autor+, verlag, erscheinungsjahr, isbn,
    (präsenzexemplar | ausleihe)?)>
<!ATTLIST buch
    signatur CDATA #REQUIRED
>
<!ELEMENT titel (#PCDATA)>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT verlag (#PCDATA)>
<!ELEMENT erscheinungsjahr (#PCDATA)>
<!ELEMENT isbn (#PCDATA)>
<!ELEMENT präsentexemplar EMPTY>
<!ELEMENT ausleihe (name, datum+)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT datum (#PCDATA)>
<!ATTLIST datum
    typ (ausleihe | rückgabe) #REQUIRED
>
```

Der Java-Praxiskurs II 3

UNIVERSITÄT LEIPZIG
LPZ E-BUSINESS
applied telematics

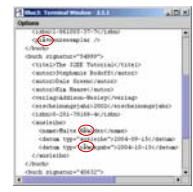
Ü2.1 XML-Dateien lesen und ausgeben

1. Einlesen und Ausgeben des XML-Dokuments mit JDOM

```
import org.jdom.*;
import org.jdom.input.*;
import org.jdom.output.*;
import java.io.*;

public class BestandLeser {

    public static void main(String[] args) {
        try {
            Document doc = new SAXBuilder(true).build("bestand.xml");
            new XMLOutputter().output(doc, System.out);
        }
        catch (JDOMException jde) { jde.printStackTrace(); }
        catch (IOException ioe) { ioe.printStackTrace(); }
    }
}
```



Der Java-Praxiskurs II 4

UNIVERSITÄT LEIPZIG
LPZ E-BUSINESS
applied telematics

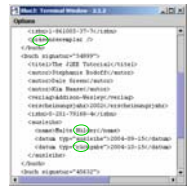
Ü2.1 XML-Dateien lesen und ausgeben

2. Korrektur der Umlaut-Ausgabe

```
import org.jdom.*;
import org.jdom.input.*;
import org.jdom.output.*;
import java.io.*;

public class BestandLeser {

    public static void main(String[] args) {
        try {
            Document doc = new SAXBuilder(true).build("bestand.xml");
            OutputStreamWriter osw = new OutputStreamWriter(System.out);
            new XMLOutputter().output(doc, osw);
            osw.close();
        }
        catch (JDOMException jde) { jde.printStackTrace(); }
        catch (IOException ioe) { ioe.printStackTrace(); }
    }
}
```



Der Java-Praxiskurs II 5

UNIVERSITÄT LEIPZIG
LPZ E-BUSINESS
applied telematics

Ü2.1 XML-Dateien lesen und ausgeben

3. Erweiterung der DTD um Vormerkungen

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT bestand (buch*)>
<!ELEMENT buch (titel, autor+, verlag, erscheinungsjahr, isbn,
    (präsenzexemplar? | (ausleihe?, vormerkung?)))>
<!ATTLIST buch
    signatur CDATA #REQUIRED
>
...
<!ELEMENT präsentexemplar EMPTY>
<!ELEMENT ausleihe (name, datum+)>
<!ELEMENT vormerkung (name)>
<!ATTLIST vormerkung
    priorität CDATA #REQUIRED
>
<!ELEMENT name (#PCDATA)>
<!ELEMENT datum (#PCDATA)>
<!ATTLIST datum
    typ (ausleihe | rückgabe) #REQUIRED
>
```

Der Java-Praxiskurs II 6

Ü2.1 XML-Dateien lesen und ausgeben

- Beispiel: Erweiterte XML-Datei

```
<bestand>
<buch signatur="54999">
  <titel>The J2EE Tutorial</titel>
  ...
  <ausleihe>
    <name>Malte Hülder</name>
    <datum typ="ausleihe">2004-09-15</datum>
    <datum typ="rückgabe">2004-10-15</datum>
  </ausleihe>
  <vormerkung priorität="3">
    <name>Matthias Book</name>
  </vormerkung>
  <vormerkung priorität="2">
    <name>Clemens Schäfer</name>
  </vormerkung>
</buch>
...
</bestand>
```

Der Java-Praxiskurs II

7

2. Manipulieren von XML-Dateien

- *gegeben*: XML-Dokument mit Buchbestand, passende DTD

1. Einlesen des XML-Dokuments mit JDOM
2. Traversierung des JDOM-Modells
 - Ausgabe der ausgeliehenen Bücher
3. Manipulation der JDOM-Elementstruktur:
 - Suchen eines bestimmten Buches
 - Löschen der Ausleihe
 - Umwandeln der höchstpriorisierten Vormerkung in eine Ausleihe
4. Zurückschreiben der geänderten Struktur in die XML-Datei

Der Java-Praxiskurs II

8

Ü2.2 XML-Dateien manipulieren

1. Einlesen des XML-Dokuments mit JDOM

```
import org.jdom.*; import org.jdom.input.*; import java.io.*;

public class BestandManipulator {
  private Document doc = null;

  public BestandManipulator(String dateiname)
    throws JDOMException, IOException {
    doc = new SAXBuilder(true).build(dateiname);
  }

  public static void main(String[] args) {
    try {
      BestandManipulator bm = new BestandManipulator("bestand.xml");
    }
    catch (JDOMException jde) { jde.printStackTrace(); }
    catch (IOException ice) { ice.printStackTrace(); }
  }
}
```

Der Java-Praxiskurs II

9

Ü2.2 XML-Dateien manipulieren

2. Ausgabe der ausgeliehenen Bücher

```
public void zeigeAusleihen() {
  Element bestandElem = doc.getRootElement();
  Iterator buchElems = bestandElem.getChildren().iterator();
  while (buchElems.hasNext()) { // Bücher durchlaufen
    Element buchElem = (Element) buchElems.next();
    Element ausleiheElem = buchElem.getChild("ausleihe");
    if (ausleiheElem != null) { // Daten der Ausleihe extrahieren
      String titel = buchElem.getChildText("titel");
      String entleiher = ausleiheElem.getChildText("name");
      String fristende = null; // Rückgabedatum suchen:
      Iterator datumElems =
        ausleiheElem.getChildren("datum").iterator();
      while (datumElems.hasNext() && (fristende == null)) {
        Element datumElem = (Element) datumElems.next();
        if (datumElem.getAttributeValue("typ").equals("rückgabe"))
          fristende = datumElem.getTextTrim();
      } // Daten ausgeben:
      System.out.println("Titel: " + titel);
      System.out.println("Entleiher: " + entleiher);
      System.out.println("Rückgabe: " + fristende);
    }
  }
}
```

Der Java-Praxiskurs II

10

Ü2.2 XML-Dateien manipulieren

3. Buch zurückgeben und an nächsten Vormerker ausleihen

```
public void rueckgabe(String signatur) {
  Element bestandElem = doc.getRootElement();
  Iterator buchElems = bestandElem.getChildren().iterator();
  while (buchElems.hasNext()) { // Bücher durchlaufen
    Element buchElem = (Element) buchElems.next();
    if (buchElem.getAttributeValue("signatur").equals(signatur)) { // Sig-Suche
      buchElem.removeChild("ausleihe"); // Ausleihe-Daten entfernen
      int minPriorität = Integer.MAX_VALUE;
      Element minVormerkungElem = null;
      int insertIndex = -1; // zur Speicherung der späteren Einfügeposition
      Iterator vormerkungElems = buchElem.getChildren("vormerkung").iterator();
      while (vormerkungElems.hasNext()) { // Vormerkungen durchlaufen
        Element vormerkungElem = (Element) vormerkungElems.next();
        if (insertIndex < 0) insertIndex = buchElem.indexOf(vormerkungElem);
        int prioritaet = Integer.parseInt(
          vormerkungElem.getAttributeValue("priorität"));
        if (prioritaet < minPriorität) { // höchstpriorisierte Vorm. suchen
          minPriorität = prioritaet;
          minVormerkungElem = vormerkungElem;
        }
      }
      ... // Fortsetzung auf nächster Folie
    }
  }
}
```

Der Java-Praxiskurs II

11

Ü2.2 XML-Dateien manipulieren

3. Buch zurückgeben und an nächsten Vormerker ausleihen

```
... // Fortsetzung von vorheriger Folie
if (minVormerkungElem != null) { // Vormerkung gefunden
  String name = minVormerkungElem.getChildText("name");
  buchElem.removeChild(minVormerkungElem); // Vormerkung entfernen
  Element ausleiheElem = new Element("ausleihe"); // Ausleihe konstr.
  Element nameElem = new Element("name"); // Benutzername
  nameElem.addContent(name);
  ausleiheElem.addContent(nameElem);
  Element ausleihdatumElem = new Element("datum"); // Ausleihdatum
  ausleihdatumElem.setAttribute("typ", "ausleihe");
  Calendar heute = Calendar.getInstance();
  ausleihdatumElem.addContent(formatDatum(heute)); // (als String einf.)
  ausleiheElem.addContent(ausleihdatumElem);
  Element rueckgabedatumElem = new Element("datum"); // Rückgabedatum
  rueckgabedatumElem.setAttribute("typ", "rückgabe");
  heute.add(Calendar.DATE, 14);
  rueckgabedatumElem.addContent(formatDatum(heute)); // (als String einf.)
  ausleiheElem.addContent(rueckgabedatumElem);
  buchElem.addContent(insertIndex, ausleiheElem); // Ausleihe einfügen
}
break; // kein weiterer Durchlauf durch Bücher notwendig
}
}
```

Der Java-Praxiskurs II

12

Ü2.2 XML-Dateien manipulieren

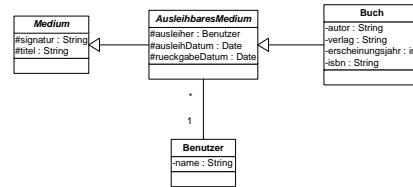
4. Elementstruktur in XML-Datei zurückschreiben

```
public void bestandSchreiben(String dateiname) throws IOException {
    FileOutputStream fos = new FileOutputStream(dateiname);
    new XMLOutputter().output(doc, fos);
    fos.close();
}
```

```
...
<buch signatur="54999">
  <titel>The J2EE Tutorial</titel>
  <autor>Stephanie Bodoff</autor>
  <autor>Dale Green</autor>
  <autor>Kim Hase</autor>
  <verlag>Addison-Wesley</verlag>
  <erscheinungsjahr>2002</erscheinungsjahr>
  <isbn>0-201-79168-4</isbn>
  <ausleihe><name>Clemens Schäfer</name><datum type="ausleihe">2004-10-03</datum><datum type="rückgabe">2004-11-03</datum></ausleihe><vormerkung
  priorität="3">
    <name>Matthias Book</name>
  </vormerkung>
</buch>
...
```

3. XML auf Objektmodell abbilden

1. Einlesen des XML-Dokuments und Speicherung in Instanzen des fachlichen Objektmodells



2. Ausgabe des Objektmodells

- Lösung: siehe Quelltext