

Übungsblatt 4 WT:III

Bis zum 05.06.2018, 23:59, sind Lösungen zu folgenden Aufgaben abzugeben: 1, 2, 3, 4, 5a,b, 6a,b,c,d.

Hinweis: Testen Sie Ihre XPath's mittels xpathtester.com.

Aufgabe 1 : XPath: Hotel (1+1+1+1 Punkte)

Gegeben sei ein XML-Schema für Listen von Hotels (Beispiel zum Testen von a,c,d: hotel.xml):

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="unqualified" attributeFormDefault="unqualified">
  <xsd:element name="hotel">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="etage" maxOccurs="20">
          <xsd:complexType>
            <xsd:sequence>
              <xsd:element name="gaestezimmer" maxOccurs="30">
                <xsd:complexType>
                  <xsd:all>
                    <xsd:element name="fernseher" type="xsd:boolean"/>
                    <xsd:element name="gastname" type="xsd:string"/>
                  </xsd:all>
                  <xsd:attribute name="nummer" type="xsd:int" use="required"/>
                </xsd:complexType>
              </xsd:element>
              <xsd:element name="putzraum" minOccurs="0"/>
            </xsd:sequence>
            <xsd:attribute name="nummer" type="xsd:int" use="required"/>
          </xsd:complexType>
        </xsd:element>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
</xsd:schema>
```

- Geben Sie einen XPath-Ausdruck an, der alle Knoten für Putzräume selektiert.
- Welche Knoten selektiert der folgende XPath-Ausdruck?
`//etage[count(gaestezimmer)>10]/@nummer`
- Geben Sie einen XPath-Ausdruck an, der die Nummern aller Gästezimmer auf Etage 1 selektiert.
- Geben Sie einen XPath-Ausdruck an, der die Nummern aller Gästezimmer mit Fernseher selektiert.

Aufgabe 2 : XSLT: Built-in Templates (1+1+1 Punkte)

Geben Sie für jede der folgenden Template-Regeln an (1) auf welche Knoten eines Dokumentbaumes die Template-Regel angewandt werden kann und (2) was für diese Knoten getan werden soll.

- `<xsl:template match="*/"/>`
`<xsl:apply-templates/>`
`</xsl:template>`
- `<xsl:template match="text()|@*">`
`<xsl:value-of select="."/>`
`</xsl:template>`
- `<xsl:template match="processing-instruction()|comment()"/>`

Aufgabe 3 : XSLT: Peanuts (2+2 Punkte)

Lade die Datei `peanuts.xml` von der Homepage herunter. In dieser Datei sind einige Namen enthalten und weitere Informationen. Stellen Sie die Informationen aus der Datei als HTML-Dokument dar. Das HTML-Dokument soll die Überschrift „Darsteller der Peanuts“ enthalten, sowie eine dreispaltige Tabelle mit den Spaltennamen „Vorname“, „Name“ und „Einführungsjahr“. Erstellen Sie ein eigenes XSLT-Stylesheet für jede Unteraufgabe.

- (a) Geben Sie ein XSLT-Stylesheet an, dass die Tabelle zusätzlich nach Nachnamen sortiert.
- (b) Geben Sie ein XSLT-Stylesheet an, in der die Tabelle nur Darsteller enthält, die nach dem Jahr 1960 eingeführt wurden.

Aufgabe 4 : Programmiermodul: XSL-Transformation P (3 Punkte)

Schreiben Sie eine XSL-Transformation um einen XML-Feed in eine HTML-Oberfläche zu überführen.

- Referenzieren Sie Ihre XSLT-Datei in der `feed.xml` der letzten Übung.
- Die Transformation soll die `feed.xml` in Ihr HTML aus dem Programmiermodul GUI oder in das der entsprechenden Musterlösung (Kursseite: in `feed-reader-gui-final.zip`) überführen.
- Entnehmen Sie so viele Informationen wie möglich der `feed.xml`, zum Beispiel auch die Seitenbeschreibung aus `<title>` und `<subtitle>`.
- Achten Sie auf eine detailgetreue Umsetzung, zum Beispiel bei der Ausgabe des Datums.
- Es soll immer der komplette Feed angezeigt werden.

Hinweis: Nicht alle Browser laden XSL-Transformationen von lokalen XML-Dateien. Nutzen Sie im Zweifel Firefox um Ihre Abgabe zu testen.

Aufgabe 5 : Wikipedia-XML-Dump (2+2+0 Punkte)

Laden Sie sich die Datei `ndswiki-latest-pages-articles-multistream.xml.zip` von der Kursseite herunter und entpacken Sie sie. Das XML-Instanzdokument (Schema: <https://www.mediawiki.org/xml/export-0.10.xsd>) enthält alle Plattdeutschen Wikipedia-Artikel.

Schreiben Sie für jede der folgenden Fragen ein SAX-Programm. Geben Sie in ihrer Abgabe jeweils die Antwort an und fügen Sie den Code bei, der die Antwort liefert.

- (a) Wie viele `page`-Elemente sind im Wikipedia-Dump enthalten?
- (b) Welche Revisions-ID hat der Artikel mit Titel „Erfurt“ im Wikipedia-Dump?
- (c) Wie viele verschiedene angemeldete Autoren hat der Wikipedia-Dump?
Hinweis: Verwenden Sie die Klasse `java.util.HashSet`.

Aufgabe 6 : Programmiermodul: Java-API P (1+2+1+3+0+0 Punkte)

Schreiben Sie eine Java-Klasse, um einen XML-Feed mittels JAXB als Objekt zu repräsentieren.

- (a) Generieren Sie Java Klassen zu Ihrem XML-Schema (falls `feed.xml` und `entry.xml` dazu valide waren; alternativ die Musterlösung `atom.xsd` von der Kursseite) mit Hilfe von `xjc`.

Beispiel: `xjc -p jaxb atom.xsd`

Wichtig: Kompilieren Sie nach der Generierung immer die erzeugte `package-info.java`!

Beispiel (bis Java 8): `javac jaxb/package-info.java`

Beispiel (Java 9, 10): `javac --add-modules java.xml.bind jaxb/package-info.java`

- (b) Erstellen Sie in der `FeedModel.java` von der Kursseite einen Konstruktor, der das XML-Instanzdokument lädt.
- Unmarshallen Sie `feed.xml` und setzen Sie den Feed als Member von `FeedModel`.
 - Validieren Sie den Feed. Werfen Sie bei Invalidität eine `RuntimeException`.¹
- (c) Erweitern Sie die `FeedModel.java` um eine Methode, die einen Eintrag per ID sucht und zurückgibt. Wurde kein Eintrag mit der ID gefunden, soll `null` zurückgegeben werden.
- (d) Erweitern Sie die `FeedModel.java` um eine Methode, die einen Eintrag hinzufügt.
- Parameter der Methode sollen die Strings `title`, `url`, `summary` und `author` sein.
 - Generieren Sie die ID aus dem `hashCode()` der URL.
 - Nutzen Sie die `getXMLGregorianCalendarNow`-Methode für den Zeitpunkt der letzten Änderung.
 - Validieren Sie den Eintrag vor dem Hinzufügen. Werfen Sie bei Invalidität eine `RuntimeException`. Überprüfen Sie dabei mittels der Methode aus Aufgabe (c) ob die ID bereits verwendet wurde.
 - Fügen Sie den Eintrag an erster Stelle hinzu.
 - Aktualisieren Sie den Wert `updated` des Feeds nach der Validierung.
 - Aktualisieren Sie zuletzt das XML-Instanzdokument mittels `Marshalling`. Nicht erschrecken: Eine etwaige `Stylesheet`-Anweisung aus Aufgabe 4 wird dabei mit dem Wert der Konstante `XSL_SHEET` überschrieben.
 - Die Methode soll die ID des neuen Eintrags zurückgeben.
- (e) Erweitern Sie die `FeedModel.java` um eine Methode, die den Feed auf einen gegebenen `java.io.PrintWriter` schreibt.²
- (f) Erweitern Sie die `FeedModel.java` um eine Methode, die einen gegebenen Eintrag auf einen gegebenen `java.io.PrintWriter` schreibt.

Hinweise:

Testen Sie mit einer `main`-Methode das Lesen und Ändern des Feeds. Testen Sie auch invalide XML-Instanzdokumente und Einträge (mit Nullwerten).

In Java 9 und 10 müssen Sie zusätzlich `--add-modules java.xml.bind` bei Aufrufen von `javac` und `java` angeben.

¹Für Java-Profis: Das ist nicht die richtige Exception für diesen Fall, aber sie erlaubt uns hier im Sinne der Übung auf die Einführung von `throws`-Auszeichnungen zu verzichten.

²Diese Methode wird im nächsten Beleg vom Server genutzt werden um den Feed in eine HTTP Response zu schreiben.

Hinweise zur Abgabe

- Erstellen Sie *eine* PDF-Datei, in der Texte und Grafiken zu den Aufgaben enthalten sind.
- Quellcode (Java, JavaScript, PHP, HTML, CSS, XML, XSL, XSD, etc.) ist als Textdatei mit entsprechender Dateiendung zu übersenden; Quellcode innerhalb des PDF-Dokuments wird nicht korrigiert.
- Referenzieren Sie Ihre Quellcode-Dateien in dem PDF-Dokument, so dass sie einer Aufgabe eindeutig zuzuordnen sind.
- Verpacken Sie Ihre Abgabe in *einer* ZIP-Datei die wie folgt benannt ist:
<Nachname>-<MatrikelNr>-webtec-blatt<Übungsblattnummer>.zip.
- Gruppenabgaben mit bis zu drei Personen sind erlaubt; pro Gruppe genügt dann eine Abgabe, in der alle Gruppenmitglieder mit Namen, Matrikel, und E-Mail-Adresse verzeichnet sind
- Senden Sie die ZIP-Datei an Johannes Kiesel (johannes.kiesel@uni-weimar.de).