Übung zur Vorlesung Datenbanksysteme II

Übungsblatt 4 — Ausgabe am 25.05.2020
Abgabe der Lösungen bis 01.06.2020, 12:00 Uhr, per Mail.

Hinweis: Verwendung einer XSLT-Prozessors

- Installation von Eclipse Web Tools (WTP): `http://download.eclipse.org/webtools/updates`
- Ausführung im Browser: `https://www.freeformatter.com/xsl-transformer.html`

Aufgabe 4.1: XSLT-Grundbegriffe

Erklären Sie den Unterschied zwischen der Dokumentwurzel und dem Wurzelelementtyp.

Aufgabe 4.2: XML-Datenbankmodell

Vergleichen Sie die Syntaxbäume der beiden folgenden XML-Dateien:

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<a>
    <b>bbb</b>
    <c>ccc</c>
    <d>ddd</d>
</a>
```

und

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<a><b>bbb</b><c>ccc</c><d>ddd</d></a>
```

Die beiden Syntaxbäume sind nicht identisch. Erklären Sie die Unterschiede.

Aufgabe 4.3: XSLT-Grundbegriffe

Erklären Sie in kompakter Form (ca. 6 - 8 Zeilen Text) den Begriff *Kontextknoten* und die Bedeutung von Kontextknoten für XSL-Transformationsregeln.

Aufgabe 4.4: XSLT-Grundbegriffe

Sie möchten mit XSLT `text`-Ausgabe erzeugen, z.B. zwecks Weiterverarbeitung mit LaTeX. Dabei kann Leerraum unerwünscht sein bzw. stören. Wie entsteht solcher Leerraum, und wie können Sie ihn vermeiden?

Aufgabe 4.4: XSLT

Erklären Sie die *Unterschiede* zwischen *XPath-Pfadausdrucken* und *LocationPathPattern*, insb. Syntaxunterschiede, Bedeutungsunterschiede und die unterschiedliche Verwendung in Templates.

Aufgabe 4.6: XML-Datenbankmodell

Gegeben sei folgende XML-Datei:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<l>
<a/><i>111
</i><i>222
</i></l>
```

und folgende XSLT-Transformation:

```
<?xml version='1.0' encoding='ISO-8859-1'
    ?>
<xsl:transform version="1.0" xmlns:xsl="
    http://www.w3.org/1999/XSL/Transform" >
<xsl:output method="text" />
<xsl:template match=" / ">

  <xsl:text>========== A: &#10;</xsl:text>
  <xsl:for-each select=" // i [position()
      =2] " >
    <xsl:value-of select=" . "/>
  </xsl:for-each>

  <xsl:text>&#10;========== B: &#10;</xsl:
      text>
  <xsl:for-each select=" // * [ position()
      =2
    and self::i ] " >
    <xsl:value-of select=" . "/>
  </xsl:for-each>

  <xsl:text>&#10;========== C: &#10;</xsl:
      text>
  <xsl:for-each select=" // node() [
      position()=2
    and self::i ] " >
    <xsl:value-of select=" . "/>
  </xsl:for-each>

</xsl:template>
</xsl:transform>
```

Welche Ausgabe wird die Transformation erzeugen? Transformatieren Sie nun die Eingabedaten mit einem XSLT-Prozessor. Erklären Sie, warum die Ausgaben der `for-each`-Scheifen unterschiedlich sind, obwohl die select-Parameter sehr ähnlich aussehen.
Hinweis: Der Makroaufruf
 erzeugt ein line-feed-Zeichen. In Linux-Systemen wird dies als Zeilenvorschub angezeigt. In anderen Systemen kann es erforderlich sein, zusätzlich mit dem Makroaufruf
 ein carriage-return-Zeichen zu erzeugen (mit
) oder nur ein carriage-return-Zeichen zu erzeugen, um eine lesbare Ausgabe zu erzeugen.

Aufgabe 4.7: XSLT-Programmierung

Sei KT ein Knotentyp, der in einem Syntaxbaum auftritt. Knoten dieses Typs können entweder mit einer Transformationsregel für diesen Typ oder mit einer `xsl:for-each`-Anweisung verarbeitet werden. Erläutern Sie die Unterschiede zwischen beiden Verarbeitungsmethoden. Wann setzen Sie welche Methode ein?

Aufgabe 4.8: XSLT

In der Vorlesung haben Sie die sog. *identische Transformation* kennengelernt. Diese reproduziert den kompletten Syntaxbaum. Die textuelle Darstellung des Syntaxbaums wird hingegen nicht zu 100% reproduziert. In welchen Details können sich die eingegebene XML-Datei und die mit der identischen Transformation erzeugte XML-Datei unterscheiden?

Aufgabe 4.9: Anwendung von XSLT

In den Lehrveranstaltungsbeschreibungen aus Aufgabe 2.4 stehen die Durchführungen einer Lehrveranstaltung als Folge von Elementen <DURCHFUEHRUNG ... />. Schreiben Sie eine Transformation mit mehreren Transformationsregeln, die grundsätzlich die ganze Eingabe unverändert in die Ausgabe kopiert. Die einzige Änderung besteht darin, daß alle DURCHFUEHRUNG-Elemente in ein einziges neues Element vom Typ DURCHFUEHRUNGEN geschachtelt werden sollen, das an der Stelle des ersten DURCHFUEHRUNG-Elements stehen soll. Sofern kein einziges DURCHFUEHRUNG-Element vorhanden ist, kann man ein (dann leeres) DURCHFUEHRUNGEN-Element er-

zeugen oder auch nicht. Geben Sie zwei Lösungsvarianten an, die in diesem Fall ein leeres DURCHFUEHRUNGENElement erzeugen bzw. nicht erzeugen.

Aufgabe 4.10: Anwendung von XSLT

Bearbeiten Sie Aufgabe 3.2 erneut, aber nicht mehr mit einer einzigen Transformationsregel, sondern setzen Sie mehrere Transformationsregeln sinnvoll ein. Vergleichen Sie beide Lösungen. Benennen Sie die Vorteile einer Lösung mit mehreren Transformationsregeln.