

Klausur Metasprachen

PI Bachelor

Prof. Dr. Kretschmer, WS 2015/2016

Für alle Aufgaben wird das folgende XML-Beispieldokument mit Daten über eine Aufgabenliste benötigt. Diese Daten sind als unvollständiges Beispiel anzusehen, d.h. **Ihre Lösungen sollten auch bei Dokumenten, die andere Daten mit der beschriebenen Struktur enthalten, gültig bleiben.**

Das XML-Dokument enthält Daten zu Aufgaben, die Kategorien zugeordnet sind. Wenn nichts anderes gesagt ist, sind die Attribute erforderlich. Die vorkommenden XML-Elemente haben folgende Bedeutung:

Bezeichnung	Bedeutung
todo	Das Wurzelement todo enthält ein oder mehrere Kindelemente <code>kategorie</code> .
kategorie	<code>kategorie</code> hat kein, ein oder mehrere Kindelemente <code>aufgabe</code> . Die enthaltenen Aufgaben haben diese Kategorie als Hauptkategorie. Das Attribut <code>id</code> enthält die eindeutige ID der Kategorie.
aufgabe	<code>aufgabe</code> hat die optionalen Kindelemente <code>notiz</code> und <code>wiederholung</code> in dieser Reihenfolge. Dahinter folgt kein, ein oder mehrere Elemente <code>zusatzkategorie</code> . Das Attribut <code>id</code> enthält die eindeutige ID der Aufgabe. Diese ID besteht aus dem Zeichen <code>a</code> , gefolgt von einer oder mehreren Ziffern. Das Attribut <code>name</code> enthält den Namen der Aufgabe. Das optionale Attribut <code>dauer</code> enthält den für die Erledigung der Aufgabe geschätzten Zeitbedarf in ganzen Stunden. Das optionale Attribut <code>fällig</code> enthält das Datum, an dem die Aufgabe fällig wird bzw. wurde, im Format <code>YYYY-MM-TT</code> . Das Attribut <code>erledigt</code> enthält entweder <code>ja</code> oder <code>nein</code> . Dieses Attribut darf fehlen. Dann ist der Wert <code>nein</code> . Das optionale Attribut <code>vorher</code> enthält eine oder mehrere IDs von Aufgaben, die erledigt sein müssen, bevor diese Aufgabe erledigt werden kann. Die IDs sind durch Leerzeichen getrennt.
wiederholung	<code>wiederholung</code> ist ein leeres Element. Wenn es vorhanden ist, wird die Aufgabe regelmäßig wiederholt. Das Attribut <code>periode</code> enthält den Zeitabstand zwischen zwei Wiederholungen im Format <code>PnYnMnDTnHnMmS</code> (ISO 8601). Das Attribut <code>ende</code> enthält ein Datum im Format <code>YYYY-MM-TT</code> . Wenn die nächste Ausführung der Aufgabe nach diesem Datum liegen würde, wird die Aufgabe nicht mehr ausgeführt. Dieses Attribut kann fehlen.
notiz	Dieses Element enthält Text, der nähere Informationen zu der jeweiligen Aufgabe enthält.
zusatzkategorie	<code>zusatzkategorie</code> enthält die ID einer zusätzlichen Kategorie, der die jeweilige Aufgabe zugeordnet ist.

Beispieldokument

```
<?xml version="1.0" encoding="utf-8"?>
<todo>
  <kategorie id="Technik">
    <aufgabe id="a4" name="Inspektion" erledigt="ja" fällig="2016-02-01">
      <notiz>Auto zur großen Inspektion bringen</notiz>
    </aufgabe>
    <aufgabe id="a10" name="TÜV" vorher="a4" dauer="2">
      <notiz>Auto zum TÜV bringen</notiz>
      <wiederholung periode="P2Y"/>
    </aufgabe>
  </kategorie>
  <kategorie id="Garten">
    <aufgabe id="a7" name="Rasen düngen" fällig="2016-04-15">
      <notiz>keinen Blandünger verwenden</notiz>
      <wiederholung periode="P1M" ende="2015-10-01"/>
    </aufgabe>
    <aufgabe id="a12" name="Schlauch reparieren" dauer="1">
      <notiz>Loch finden und flicken</notiz>
      <zusatzkategorie>Technik</zusatzkategorie>
    </aufgabe>
    <aufgabe id="a13" name="Unkraut jäten" dauer="2"></aufgabe>
    <aufgabe id="a3" name="Bewässern" dauer="3" vorher="a13 a12"/>
  </kategorie>
  <kategorie id="Freizeit"/>
  <kategorie id="Finanzen">
    <aufgabe id="a1" name="Rechnung bezahlen" fällig="2016-03-08">
      <notiz>für elektrische Heckenschere</notiz>
      <zusatzkategorie>Garten</zusatzkategorie>
      <zusatzkategorie>Technik</zusatzkategorie>
    </aufgabe>
  </kategorie>
</todo>
```

Aufgabe 1

(15 Punkte) Geben Sie eine möglichst gut zutreffende DTD für das obige XML-Dokument an.

Aufgabe 2

(15 Punkte) Geben Sie je einen XPath-Ausdruck für folgende Mengen an.

- Aufgaben, die Garten als Haupt- oder Zusatzkategorie haben.
- Kategorien, die erledigte Aufgaben enthalten.
- Aufgaben, die vor Aufgaben mit mindestens 3 Stunden Dauer erledigt werden müssen.

Aufgabe 3

(27 Punkte) Schreiben Sie XSLT-Templates, die (wie in der Beispielausgabe gezeigt) die Aufgabenliste in einer XHTML-Tabelle ausgeben. Die Aufgaben werden in einer zweispaltigen Tabelle mit den Überschriften *Kategorie* und *Aufgabe* angezeigt und sind alphabetisch nach der Kategorie sortiert. Für jede Kategorie wird in der Tabelle ein eigenes `tbody`-Element verwendet. Der Name der Kategorie wird nur einmal in einer Zelle angezeigt, die sich über alle Aufgabenzeilen dieser Kategorie erstreckt. Wenn eine Kategorie keine Aufgaben enthält, wird in der Spalte *Aufgabe* der Text *keine Aufgaben* angezeigt. Der Name erledigter Aufgaben wird *kursiv* angezeigt. Hierzu wird für die Zelle eine entsprechende Klasse (`class`) gesetzt. Sie können das folgende Template für `todo` als gegeben voraussetzen. Schreiben Sie nur das Template für `kategorie` und weitere Templates, die dort verwendet werden. Verwenden Sie weder `for-each` noch `call-template`.

```
<xsl:template match="todo">
  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="de" lang="de">
    <head>
      <meta charset="utf-8" />
      <title>Aufgaben</title>
      <style>
        .erledigt {
          font-style: italic;
        }
        table, td, th {
          border: 1px solid black;
        }
      </style>
    </head>
    <body>
      <h1>Aufgaben</h1>
      <table>
        <thead>
          <th>Kategorie</th>
          <th>Aufgabe</th>
        </thead>
        <xsl:apply-templates select="kategorie">
          <xsl:sort select="@id"/>
        </xsl:apply-templates>
      </table>
    </body>
  </html>
</xsl:template>
```

Beispielausgabe

Aufgaben

Kategorie	Aufgabe
Finanzen	Rechnung bezahlen
Freizeit	keine Aufgaben
Garten	Rasen düngen
	Schlauch reparieren
	Unkraut jäten
	Bewässern
Technik	<i>Inspektion</i>
	TÜV

Aufgabe 4

(27 Punkte) Geben Sie folgende Teile eines möglichst gut zutreffenden XML-Schemas für das obige XML-Dokument an.

- Die Definition eines Typs `tAufgabeId` für die Angabe der ID einer Aufgabe.
- Die Definition des wiederholung-Elementes.
- Die Definition eines Typs `tErledigt` für die Angabe, ob eine Aufgabe erledigt wurde oder nicht.
- Die Definition eines Typs `tAufgaben` für die Angabe der Aufgaben, die vor einer Aufgabe erledigt werden müssen (Typ des Attributes vorher)
- Die Definition des `aufgabe`-Elementes unter Verwendung von a)-d).

Aufgabe 5

(16 Punkte) Schreiben Sie eine JavaScript-Funktion `ermittleVorgänger`. Die Funktion erhält als Parameter den Dokumentknoten des ganzen XML-Dokumentes und einen String mit der ID einer Aufgabe. Die Methode ermittelt für alle Aufgaben, die laut Attribut `vorher` vor der angegebenen Aufgabe erledigt werden müssen, den Namen und gibt ein Array mit diesen Namen zurück. Die Reihenfolge der Namen ergibt sich aus der Reihenfolge im Attribut `vorher`. Wenn das Attribut `vorher` fehlt, wird ein leeres Array zurückgegeben.

Sie dürfen voraussetzen, dass das Dokument korrekt aufgebaut ist.

Beispiel:

`ermittleVorgänger(xmlDoc, "a3")` gibt das Array `["Unkraut jäten", "Schlauch reparieren"]` zurück.