

Klausur Metasprachen

PI Bachelor

Prof. Dr. Kretschmer, WS 2017/2018

Für alle Aufgaben wird das folgende XML-Beispieldokument mit Daten über ein Seminar 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 den Teilnehmern und Vorträgen eines Seminars. Bei der Anmeldung zum Seminar kann jeder Teilnehmer aus einer Liste von Themen bis zu drei Wunschthemen auswählen und diese mit einer Priorität versehen. Der Seminarleiter versucht beim Erstellen des Vortragsplan jedem Teilnehmer eines seiner Wunschthemen unter Berücksichtigung der Priorität zuzuteilen. Wenn nichts anderes gesagt ist, sind die Attribute erforderlich. Die vorkommenden XML-Elemente haben folgende Bedeutung:

Bezeichnung	Bedeutung
Seminar	Das Wurzelement seminar enthält die Kindelemente themen, teilnehmer, vortragsplan in dieser Reihenfolge. Das Attribut name gibt den Seminarnamen an.
Themen	themen hat ein oder mehrere Kindelemente thema.
Thema	thema enthält den Namen des Vortragsthemas. Das Attribut nr enthält eine Nummer, die das Thema identifiziert. Diese besteht aus dem Zeichen t, gefolgt von einer oder mehreren Ziffern.
teilnehmer	teilnehmer hat beliebig viele Kindelemente stud.
Stud	stud hat als Kindelemente erst vorname, dann nachname, gefolgt von ein, zwei oder drei wunsch-Elementen. Das Attribut matrikel von stud enthält die Matrikelnummer der/des Studierenden, die diesen identifiziert. Sie besteht aus dem Zeichen m, gefolgt von 7 Ziffern. Das Attribut email enthält eine Emailadresse der Form xy@htwsaar.de, wobei statt xy zwei beliebige Kleinbuchstaben stehen können. Der Teil hinter @ ist immer htwsaar.de
Wunsch	wunsch ist ein leeres Element. Das Attribut prio enthält die Zahl 1,2, oder 3. Diese gibt die Priorität des Wunsches an. Das Attribut thema enthält die Nummer eines vorhandenen Themas.
vortragsplan	vortragsplan enthält beliebig viele vortrag-Elemente.
Vortrag	vortrag ist ein leeres Element. Das Attribut matrikel enthält die Matrikelnummer einer/s vorhandenen Studierenden. Das Attribut thema enthält die Nummer eines vorhandenen Themas. Das Attribut datum enthält das Datum des Vortrags im Format JJJJ-MM-TT. Das Attribut uhrzeit enthält die Uhrzeit des Vortragsbeginns im Format hh:mm:ss. Wenn noch kein Termin für den Vortrag feststeht, fehlen Datum und Uhrzeit.

Beispieldokument

```
<?xml version="1.0" encoding="utf-8" ?>
<seminar name="Theoretische Informatik">
  <themen>
    <thema nr="t1">Reguläre Ausdrücke</thema>
    <thema nr="t2">Treaps</thema>
    <thema nr="t3">Skiplisten</thema>
    <thema nr="t4">Die Greibach-Normalform</thema>
    <thema nr="t5">Berechenbarkeit</thema>
    <thema nr="t6">Das Lambda-Kalkül von Church</thema>
    <thema nr="t7">Alternative Berechnungsmodelle</thema>
    <thema nr="t8">Der Rekursionssatz</thema>
    <thema nr="t9">Schaltkreiskomplexität</thema>
    <thema nr="t10">Probabilistische Algorithmen</thema>
  </themen>
  <teilnehmer>
    <stud matrikel="m1234567" email="sg@htwsaar.de">
      <vorname>Sheila</vorname>
      <nachname>Greibach</nachname>
      <wunsch prio="1" thema="t1"/>
      <wunsch prio="2" thema="t4"/>
      <wunsch prio="3" thema="t2"/>
    </stud>
    <stud matrikel="m2345678" email="ac@htwsaar.de">
      <vorname>Alonzo</vorname>
      <nachname>Church</nachname>
      <wunsch prio="1" thema="t4"/>
      <wunsch prio="2" thema="t1"/>
      <wunsch prio="3" thema="t6"/>
    </stud>
    <stud matrikel="m3456789" email="sk@htwsaar.de">
      <vorname>Stephen</vorname>
      <nachname>Kleene</nachname>
      <wunsch prio="1" thema="t1"/>
    </stud>
    <stud matrikel="m4567890" email="rm@htwsaar.de">
      <vorname>Rajeev</vorname>
      <nachname>Motwani</nachname>
      <wunsch prio="1" thema="t3"/>
      <wunsch prio="2" thema="t2"/>
    </stud>
    <stud matrikel="m5678901" email="at@htwsaar.de">
      <vorname>Alan</vorname>
      <nachname>Turing</nachname>
      <wunsch prio="1" thema="t5"/>
      <wunsch prio="2" thema="t7"/>
      <wunsch prio="3" thema="t10"/>
    </stud>
  </teilnehmer>
  <vortragsplan>
    <vortrag matrikel="m3456789" thema="t1" datum="2017-07-04" uhrzeit="13:45:00"/>
    <vortrag matrikel="m2345678" thema="t6" datum="2017-07-11" uhrzeit="13:45:00"/>
    <vortrag matrikel="m1234567" thema="t4" datum="2017-07-11" uhrzeit="15:00:00"/>
    <vortrag matrikel="m4567890" thema="t3" datum="2017-07-18" uhrzeit="13:45:00"/>
    <vortrag matrikel="m5678901" thema="t5"/>
  </vortragsplan>
</seminar>
```

Aufgabe 1

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

Aufgabe 2

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

- Studierende, die nur ein Wunschthema angegeben haben.
- Vorträge ohne Datum und Uhrzeit.
- Themen, die nicht als Vortrag vergeben sind.

Aufgabe 3

(21 Punkte) Schreiben Sie XSLT-Templates für die Erzeugung folgender Teile eines XHTML-Dokumentes: die **Tabellenzeilen** des Vortragsplans ohne die Kopfzeile.

Bei jedem Vortrag sind folgende Informationen in der genannten Reihenfolge angegeben: das Datum, die Uhrzeit im Format hh:mm, die oder der Studierende im Format Nachname, Vorname, die Nummer des Themas ohne führendes t, das Thema und die Priorität des/der Studierenden für dieses Thema.

Das Datumsfeld bleibt leer, wenn der Vortrag nicht der erste an diesem Tag ist.

Sie dürfen voraussetzen, dass die Vorträge in der XML-Datei nach Datum und dann nach Uhrzeit sortiert sind, wobei die noch nicht terminierten Vorträge zuletzt stehen.

Beispielausgabe

Vortragsplan des Seminars Theoretische Informatik

Datum	Zeit	Studierende(r)	Nr	Thema	Prio
2017-07-04	13:45	Kleene, Stephen	1	Reguläre Ausdrücke	1
2017-07-11	13:45	Church, Alonzo	6	Das Lambda-Kalkül von Church	3
	15:00	Greibach, Sheila	4	Die Greibach-Normalform	2
2017-07-18	13:45	Motwani, Rajeev	3	Skiplisten	1
		Turing, Alan	5	Berechenbarkeit	1

Aufgabe 4

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

- Die Definition eines Typs `tMatrikel` für die Matrikelnummer.
- Die Definition eines Typs `tEmail` für die Emailadresse.
- Die Definition des `stud`-Elementes unter Verwendung von a), b) und eines vorgegebenen `wunsch`-Elementes.
- Die Definition eines `key`-Elementes, das dafür sorgt, dass die Wünsche jeweils einer/s Studierenden verschiedene Priorität haben.

Aufgabe 5

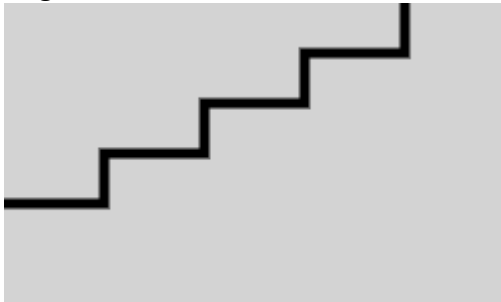
(18 Punkte) Schreiben Sie eine JavaScript-Funktion `averagePriority`. Die Funktion erhält als Parameter den Dokumentknoten des ganzen XML-Dokumentes. Die Funktion berechnet die durchschnittliche Priorität der Wünsche, die im Vortragsplan ausgewählt sind. Sie dürfen voraussetzen, dass das Dokument korrekt aufgebaut ist.

Im Beispiel ergibt sich $1.6 = (1+3+2+1+1)/5$

Aufgabe 6

(18 Punkte) Schreiben Sie eine JavaScript-Funktion `createStair`. Die Funktion erhält die Parameter `xdist`, `ydist`, `count`, `color`, `width` und erzeugt ein SVG-path-Element für eine Treppenlinie mit `count` Stufen, die am linken Rand des umfassenden svg-Elementes beginnt und am oberen Rand endet. `xdist` ist die Breite einer Stufe, `ydist` die Höhe. `color` ist die Zeichenfarbe und `width` die Strichstärke. Alle Größenangaben haben als Einheit Pixel.

Der Aufruf `createStair(50, 25, 4, 'black', 5)` erzeugt die Treppenlinie im folgenden Bild:



Die graue Fläche ist das `svg`-Element, dessen Kind das zu erzeugende `path`-Element ist, und hat in diesem Beispiel die Größe 250x150. Diese Fläche soll von der Funktion `createStair` nicht erzeugt werden.