

Gruppe A	PRÜFUNG AUS "SEMISTRUKTURIERTE DATEN" 184.705		26. 11. 2012
Kennnr.	Matrikelnr.	Familienname	Vorname

Arbeitszeit: 100 Minuten. Aufgaben sind auf den Angabeblättern zu lösen; Zusatzblätter werden nicht gewertet. Bitte tragen Sie Namen, Studienkennzahl und Matrikelnummer ein. Halten Sie Ihren Studentenausweis bereit.

Aufgabe 1:

(9)

Betrachten Sie die folgende XML-Schema Datei **test.xsd**:

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="a" type="atype"/>

  <xsd:complexType name="atype">
    <xsd:all>
      <xsd:element name="b" type="xsd:int"/>
      <xsd:element name="a" minOccurs="0" maxOccurs="1" type="atype"/>
      <xsd:element name="c" minOccurs="0" maxOccurs="1" type="xsd:int"/>
    </xsd:all>
  </xsd:complexType>

</xsd:schema>
```

Betrachten Sie weiters die sechs verschiedenen XML-Dateien, die unten angeführt sind.

Sie können davon ausgehen, dass alle folgenden XML-Dateien wohlgeformt sind. Es geht also lediglich darum, ihre Gültigkeit bezüglich **test.xsd** zu entscheiden.

Kreuzen Sie an, welche der folgenden XML-Dateien gültig bezüglich **test.xsd** sind.

- | | | |
|--|---|---|
| 1. <a/> | gültig <input type="radio"/> | ungültig <input checked="" type="radio"/> |
| 2. <a>1 | gültig <input checked="" type="radio"/> | ungültig <input type="radio"/> |
| 3. <a><a><c>0</c>0 | gültig <input type="radio"/> | ungültig <input checked="" type="radio"/> |
| 4. <a>0<a>0 | gültig <input checked="" type="radio"/> | ungültig <input type="radio"/> |
| 5. <a>0<c>0</c>0 | gültig <input type="radio"/> | ungültig <input checked="" type="radio"/> |
| 6. <a><a><a>000 | gültig <input checked="" type="radio"/> | ungültig <input type="radio"/> |

(Pro korrekter Antwort 1.5 Punkte, **pro inkorrektter Antwort -1.5 Punkte**, pro nicht beantworteter Frage 0 Punkte, für die gesamte Aufgabe mindestens 0 Punkte)

Aufgabe 2:

(12)

Kreuzen Sie an, ob die folgenden Aussagen wahr oder falsch sind.

1. Der XPath-Ausdruck `//station` ist die Kurzschreibweise des XPath-Ausdrucks `/descendant::station` wahr ☐ falsch ☒
2. XQuery ist keine W3C-Recommendation, aber ein de-facto Standard. wahr ☐ falsch ☒
3. Eine XML Schema Definition muss ein wohlgeformtes XML-Dokument sein. wahr ☒ falsch ☐
4. Um die Wohlgeformtheit eines XML-Dokuments zu überprüfen wird eine DTD oder ein XML Schema benötigt. wahr ☐ falsch ☒
5. Bei SAX kann maximal ein Filter zwischen Reader und Applikation geschaltet werden. wahr ☐ falsch ☒
6. SAX Textevents treten niemals direkt hintereinander auf. wahr ☐ falsch ☒
7. Rekursive Definitionen wie z.B. `<!ELEMENT B (A,B?,C)>` sind in DTDs verboten. wahr ☐ falsch ☒
8. HTML ist eine Weiterentwicklung von XML. wahr ☐ falsch ☒

(Pro korrekter Antwort 1.5 Punkte, **pro inkorrektter Antwort -1.5 Punkte**, pro nicht beantworteter Frage 0 Punkte, für die gesamte Aufgabe mindestens 0 Punkte)

Die folgenden Aufgaben 3 – 7 beziehen sich auf das XML-Dokument **ubahn.xml**, das Sie auf der letzten Seite dieser Prüfungsangabe finden.

Aufgabe 3:

(12)

Vervollständigen Sie das DTD Dokument **ubahn.dtd**, sodass XML-Dokumente in der Gestalt von **ubahn.xml** (siehe Anhang) bezüglich dieser DTD gültig sind. Berücksichtigen Sie beim Erstellen der DTD folgende Punkte:

- Das Element **ubahn** ist das Wurzelement und besteht aus mindestens einem **linie**-Element gefolgt von genau einem **stationen**-Element.
- Das Element **linie** besteht aus mindestens zwei **haltestelle**-Elementen; diese sollen leer sein.
- Das Element **stationen** besteht aus **station**-Elementen welche zu jeder Station **sehenswürdigkeit**-Elemente auflisten (bestimmen Sie selbst die Häufigkeiten entsprechend dem **ubahn.xml** Dokument im Anhang). **sehenswürdigkeit**-Elemente haben gemischten Inhalt und können ihrerseits leere Elemente der Form **ulinie** und **slinie** enthalten.
- Spezifizieren Sie die Attribute der angegebenen Elemente gemäß dem **ubahn.xml** file und wählen Sie selbst sinnvolle Typen aus. Das **name**-Attribut des **linie**-Elements und das **kürzel**-Attribut des **station**-Elements sind außerdem als Primärschlüssel zu deklarieren. Versuchen Sie die entsprechenden Fremdschlüssel zu finden und in der DTD abzubilden.

Datei **ubahn.dtd**:

```
<!ELEMENT ubahn (linie+, stationen)>

<!ELEMENT linie (haltestelle,haltestelle+)>
<!ATTLIST linie
  name ID #REQUIRED>

<!ELEMENT haltestelle EMPTY>
<!ATTLIST haltestelle
  kürzel IDREF #REQUIRED>

<!ELEMENT stationen (station+)>

<!ELEMENT station (sehenswürdigkeit*)>
<!ATTLIST station
  kürzel ID #REQUIRED
  name CDATA #REQUIRED>

<!ELEMENT sehenswürdigkeit (#PCDATA | ulinie | slinie)*>
<!ATTLIST sehenswürdigkeit
  name CDATA #REQUIRED>

<!ELEMENT ulinie EMPTY>
<!ATTLIST ulinie
  nr IDREF #REQUIRED>

<!ELEMENT slinie EMPTY>
<!ATTLIST slinie
  nr CDATA #REQUIRED>
```

Aufgabe 4:

(6)

Stellen Sie sich vor, Sie definieren eine XML-Schema Datei für Dokumente in der Gestalt von **ubahn.xml**. Welche Aussagen würden auf ein solches XML-Schema, bzw. allgemein zum Thema XML-Schema, zutreffen:

1. Zum Spezifizieren des Inhalts des **ubahn**-Elements eignet sich das **xsd:all**-Konstrukt. wahr ☐ falsch ☒
2. Betrachten Sie das **nr**-Attribut des Elements **slinie** (= Straßenbahnlinie).
In XML-Schema können Sie (ohne explizite Aufzählung) spezifizieren, dass Attributwerte eine Zahl zwischen 1 und 99 oder ein Buchstabe sein sollen. wahr ☒ falsch ☐
3. Betrachten Sie das gemischte Element **sehenswürdigkeit**. In XML-Schema ist es möglich zu spezifizieren, dass innerhalb dieses Elements zuerst **ulinie**, dann erst **slinie** Elemente auftreten. wahr ☒ falsch ☐
4. In XML-Schema dürfen gemischte Elemente keine Schlüsselattribute besitzen (Dokumenten- vs. Datensicht). wahr ☐ falsch ☒

(Pro korrekter Antwort 1.5 Punkte, **pro inkorrektter Antwort -1.5 Punkte**, pro nicht beantworteter Frage 0 Punkte, für die gesamte Aufgabe mindestens 0 Punkte)

Aufgabe 5:

(10)

Betrachten Sie die folgenden XPath-Abfragen angewandt auf das Dokument **ubahn.xml** (siehe Anhang).

- Falls der angegebene XPath Ausdruck keine Knoten selektiert, notieren Sie im entsprechenden Feld “leere Ausgabe”.
- Falls als Ergebnis **station** oder **haltestelle** Elemente selektiert werden, geben Sie jeweils das Attribut **kürzel** an.
- Falls als Ergebnis mehrere Elemente selektiert werden, trennen Sie die jeweiligen Ausgaben durch Leerzeichen.
- Falls als Ergebnis eine Zahl ausgegeben wird, geben Sie diese an.

Betrachten Sie dazu folgendes Beispiel:

```
//station
```

```
KP SZ VT LA SR HZ SP PR
```

Geben Sie nun die entsprechende Ausgaben der folgenden XPath-Abfragen an.

```
//station[sehenswürdigkeit]
```

```
KP SZ VT PR
```

```
//station[sehenswürdigkeit[*]]
```

```
SZ VT
```

```
count(//haltestelle/@kürzel)
```

```
16
```

```
//haltestelle[../@name='U3']
```

```
LA SZ HZ VT
```

```
/station[@kürzel=//linie[@name='U1']/haltestelle/@kürzel  
[ @kürzel=//linie[@name='U2']/haltestelle/@kürzel]
```

```
KP PR
```

Aufgabe 6:

(8)

Betrachten Sie folgende-XQuery Abfrage **ubahn.xq**:

```
for $l in //linie
let $h := $l/*/@kürzel
order by $l/@name descending
return element{string($l/@name)} {
  count(//station[@kürzel=$h]/*)
}
```

Geben Sie nun die Ausgabe von **ubahn.xq** angewandt auf **ubahn.xml** an.

Die exakte Behandlung von Whitespaces ist für dieses Beispiel nicht relevant.

```
<U4>3</U4>
<U3>4</U3>
<U2>7</U2>
<U1>5</U1>
```

Erstellen Sie ein XSLT-Stylesheet **ubahn.xsl**, das angewandt auf Dokumente der Gestalt **ubahn.xml** zu jeder Station die U-Bahn Linien anführt, die dort ihre Haltestellen haben. Für das Dokument **ubahn.xml** soll folgende Ausgabe erzeugt werden:

```
<umsteigemöglichkeiten>
  <station name="Karlsplatz"    linien="U1 U2 U4"/>
  <station name="Stephansplatz" linien="U1 U3"/>
  <station name="Volkstheater"   linien="U2 U3"/>
  <station name="Landstraße"    linien="U3 U4"/>
  <station name="Schottenring"  linien="U2 U4"/>
  <station name="Herrengasse"   linien="U3"/>
  <station name="Schwedenplatz" linien="U1 U4"/>
  <station name="Praterstern"   linien="U1 U2"/>
</umsteigemöglichkeiten>
```

Um aus einer Sequenz von Strings doppelte Vorkommen herauszufiltern, ist die XPath 2.0 Funktion **distinct-values** hilfreich!

Vervollständigen Sie hier das XSLT-Stylesheet **ubahn.xsl**. Die Verwendung von Kontrollstrukturen wie **xsl:for-each** ist für die Lösung grundsätzlich erlaubt, aber nicht erforderlich! Sie brauchen sich nicht um Whitespaces etc. zu kümmern.

Datei **ubahn.xsl**:

```
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="/">
    <umsteigemöglichkeiten>
      <xsl:apply-templates select="//station"/>
    </umsteigemöglichkeiten>
  </xsl:template>

  <xsl:template match="station">
    <station name="{@name}" linien="{distinct-values(
      //linie[haltestelle/@kürzel=current()/@kürzel]/@name) }"/>
  </xsl:template>

</xsl:stylesheet>
```

Vervollständigen Sie den folgenden SAX-Handler, der angewandt auf Dokumente der Gestalt **ubahn.xml** zu jeder U-Bahn Linie die entsprechenden Stationsnamen ausgibt. Die Reihenfolge der Stationsnamen in der Ausgabe soll der Reihenfolge der **haltestelle** Elemente entsprechen. Die Ausgabe soll in die Java-Standardausgabe erfolgen. Für das Dokument **ubahn.xml** soll folgende Ausgabe erzeugt werden:

U1: Karlsplatz Stephansplatz Schwedenplatz Praterstern

U2: Karlsplatz Volkstheater Schottenring Praterstern

U3: Landstraße Stephansplatz Herrengasse Volkstheater

U4: Karlsplatz Landstraße Schwedenplatz Schottenring

```
public class UbahnHandler extends DefaultHandler {
    Map<String,List<String>> linieHaltestellen = new HashMap<String,List<String>>();
    Map<String,String> stationsNamen = new HashMap<String,String>();

    String linie;

    public void startElement(String uri, String localName, String qName,
        Attributes atts) throws SAXException {
        if ("linie".equals(localName)) {
            linie = atts.getValue("name");
            linieHaltestellen.put(linie, new LinkedList<String>());
        }
        if ("haltestelle".equals(localName)) {
            linieHaltestellen.get(linie).add(atts.getValue("kürzel"));
        }
        if ("station".equals(localName)) {
            stationsNamen.put(atts.getValue("kürzel"), atts.getValue("name"));
        }
    }

    public void endDocument() throws SAXException {
        for (String linie : linieHaltestellen.keySet()) {
            System.out.print(linie + ": ");
            for (String kürzel: linieHaltestellen.get(linie)) {
                System.out.print(stationsNamen.get(kürzel) + " ");
            }
            System.out.println();
        }
    }
}
```


Sie können diese Seite abtrennen!

Datei **ubahn.xml**:

```
<ubahn>
  <linie name="U1">
    <haltestelle kürzel="KP"/>
    <haltestelle kürzel="SZ"/>
    <haltestelle kürzel="SP"/>
    <haltestelle kürzel="PR"/>
  </linie>
  <linie name="U2">
    <haltestelle kürzel="KP"/>
    <haltestelle kürzel="VT"/>
    <haltestelle kürzel="SR"/>
    <haltestelle kürzel="PR"/>
  </linie>
  <linie name="U3">
    <haltestelle kürzel="LA"/>
    <haltestelle kürzel="SZ"/>
    <haltestelle kürzel="HZ"/>
    <haltestelle kürzel="VT"/>
  </linie>
  <linie name="U4">
    <haltestelle kürzel="KP"/>
    <haltestelle kürzel="LA"/>
    <haltestelle kürzel="SP"/>
    <haltestelle kürzel="SR"/>
  </linie>
  <stationen>
    <station kürzel="KP" name="Karlsplatz">
      <sehenswürdigkeit name="Karlskirche"/>
      <sehenswürdigkeit name="Staatsoper"/>
      <sehenswürdigkeit name="TU Wien ; )"/>
    </station>
    <station kürzel="SZ" name="Stephansplatz">
      <sehenswürdigkeit name="Stephansdom">
        Direkt an den U-Bahnlinien <ulinie nr="U1"/> und <ulinie nr="U3"/> gelegen.
      </sehenswürdigkeit>
    </station>
    <station kürzel="VT" name="Volkstheater">
      <sehenswürdigkeit name="Museumsquartier">
        Auch von der Station "Museumsquartier" der Linie <ulinie nr="U2"/> leicht zu erreichen,
        oder mit den Straßenbahnlinien <slinie nr="1"/>, <slinie nr="2"/> oder <slinie nr="D"/>.
      </sehenswürdigkeit>
      <sehenswürdigkeit name="Naturhistorisches Museum"/>
      <sehenswürdigkeit name="Kunsthistorisches Museum"/>
    </station>
    <station kürzel="LA" name="Landstraße"/>
    <station kürzel="SR" name="Schottenring"/>
    <station kürzel="HZ" name="Herrengasse"/>
    <station kürzel="SP" name="Schwedenplatz"/>
    <station kürzel="PR" name="Praterstern">
      <sehenswürdigkeit name="Riesenrad"/>
    </station>
  </stationen>
</ubahn>
```