

Gruppe A

Bitte tragen Sie **sofort** und **leserlich** Namen, Studienkennzahl und Matrikelnummer ein und legen Sie Ihren Studentenausweis bereit.

PRÜFUNG AUS "SEMISTRUKTURIERTE DATEN" 181.135			27. 6. 2007
Kennnr.	Matrikelnr.	Familienname	Vorname

Arbeitszeit: 120 Minuten. Aufgaben sind auf den Angabebättern zu lösen; Zusatzblätter werden nicht gewertet.

**Aufgabe 1:**

(9)

Vervollständigen Sie die DTD **nachrichten.dtd**, so dass das XML-Dokument **nachrichten.xml** bezüglich dieser DTD gültig ist. Berücksichtigen Sie beim Erstellen der DTD folgende Punkte:

- Die Verknüpfung mit einer XML Schema-Definition ist optional.
- Ein "nachrichten"-Element kann 0 oder beliebig viele "nachricht"-Subelemente haben.
- Ein "nachricht"-Element kann 1 oder beliebig viele "titel"-Subelemente haben.
- Das "redakteur"-Subelement des "nachricht"-Elements ist optional.
- Das Attribut "bereich" des "nachricht"-Elements soll den Default-Wert "sport" haben.
- Ein "text"-Element hat gemischten Inhalt. Als Subelemente können folgende Elemente beliebig oft und in beliebiger Reihenfolge auftreten: "schlagwort", "auto" und/oder "person".

Datei **nachrichten.dtd**:

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT auto (#PCDATA)>
<!ELEMENT person (#PCDATA)>
<!ELEMENT schlagwort (#PCDATA)>

<!ELEMENT orf:nachrichten (orf:nachricht*)>
<!ATTLIST orf:nachrichten xmlns:orf CDATA #REQUIRED
    xmlns:xsi CDATA #IMPLIED
    xsi:schemaLocation CDATA #IMPLIED>
<!ELEMENT orf:nachricht (nr, titel+, datum, redakteur?, text)>
<!ATTLIST orf:nachricht bereich CDATA "sport">
<!ELEMENT nr (#PCDATA)>
<!ELEMENT titel (#PCDATA)>
<!ELEMENT datum (#PCDATA)>
<!ELEMENT redakteur (#PCDATA)>
<!ELEMENT text (#PCDATA | schlagwort | auto | person)*>
```

**Aufgabe 2:**

Vervollständigen Sie in der XML Schema Definition **nachrichten.xsd** die complexType-Definition des Typs "Nachricht-Typ", so dass das XML-Dokument **nachrichten.xml** bezüglich dieses Schemas gültig ist. Berücksichtigen Sie dabei folgende Punkte:

- Das "redakteur"-Subelement des "nachricht"-Elements ist optional.
- Das Attribut "bereich" des "nachricht"-Elements soll den Default-Wert "sport" haben.
- Ein "text"-Element hat gemischten Inhalt. Als Subelemente können folgende Elemente beliebig oft und in beliebiger Reihenfolge auftreten: "schlagwort", "auto" und/oder "person".

Datei **nachrichten.xsd**:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="http://www.orf.at/news"
  xmlns:news="http://www.orf.at/news">

  <xs:element name="nachrichten">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="news:nachricht" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="nachricht" type="news:NachrichtTyp"/>

  <xs:complexType name="NachrichtTyp">
    <!-- Vervollständigen Sie diese complexType-Definition -->
    <xs:sequence>
      <xs:element name="nr" type="xs:integer"/>
      <xs:element name="titel" type="xs:string" maxOccurs="unbounded"/>
      <xs:element name="datum" type="xs:date" />

      <xs:element name="redakteur" type="xs:string" minOccurs="0" />
      <xs:element name="text">
        <xs:complexType mixed="true">
          <xs:choice maxOccurs="unbounded">
            <xs:element name="schlagwort" type="xs:string" />
            <xs:element name="auto" type="xs:string" />
            <xs:element name="person" type="xs:string" />
          </xs:choice>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="bereich" type="xs:string" default="sport"/>
  </xs:complexType>
</xs:schema>
```

**Aufgabe 3:**

Vervollständigen Sie die XML Schema Definition **fussball.xsd**, so dass das XML-Dokument **fussball.xml** bezüglich dieses Schemas gültig ist. Berücksichtigen Sie beim Vervollständigen der XML Schema Definition folgende Punkte:

- Sie brauchen sich **nicht um die Definition** der Typen “vereineTyp”, “bewerbeTyp” und “titelTyp” zu **kümmern!**
- Ergänzen Sie die Element-Definition von “fussball”, so dass folgende Schlüssel definiert werden:
  - Das Attribut v\_id der “verein”-Elemente ist ein Primärschlüssel für die Vereine.
  - Das Subelement b\_id der “bewerb”-Elemente ist ein Primärschlüssel für die Bewerbe.
  - Die “titel”-Elemente enthalten folgende zwei Fremdschlüssel: Das Subelement “verein” ist ein Fremdschlüssel auf den Primärschlüssel der Vereine, und das Subelement “bewerb” ist ein Fremdschlüssel auf den Primärschlüssel der Bewerbe.

Datei **fussball.xsd**:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  <xsd:element name="fussball">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="vereine" type="vereineTyp"/>
        <xsd:element name="bewerbe" type="bewerbeTyp"/>
        <xsd:element name="titel" maxOccurs="unbounded" type="titelTyp"/>
      </xsd:sequence>
    </xsd:complexType>
    <!-- Fügen Sie hier die Primär- und Fremdschlüssel-Definitionen ein. -->

    <xsd:key name="pk_verein">
      <xsd:selector xpath="vereine/verein"/>
      <xsd:field xpath="@v_id"/>
    </xsd:key>
    <xsd:key name="pk_bewerb">
      <xsd:selector xpath="bewerbe/bewerb"/>
      <xsd:field xpath="b_id"/>
    </xsd:key>
    <xsd:keyref name="fk_verein" refer="pk_verein">
      <xsd:selector xpath="titel"/>
      <xsd:field xpath="verein"/>
    </xsd:keyref>
    <xsd:keyref name="fk_bewerb" refer="pk_bewerb">
      <xsd:selector xpath="titel"/>
      <xsd:field xpath="bewerb"/>
    </xsd:keyref>
  </xsd:element>

</xsd:element>

  <!-- restliches Schema ist nicht Teil der Prüfungsaufgabe! -->
  <xsd:complexType name="vereineTyp"> ..... </xsd:complexType>
  <xsd:complexType name="bewerbeTyp"> ..... </xsd:complexType>
  <xsd:complexType name="titelTyp"> ..... </xsd:complexType>
</xsd:schema>

```

#### Aufgabe 4:

(12)

Vervollständigen Sie das XSLT Stylesheet **fussball2vereine.xsl**, das auf Instanz-Dokumente des Schemas **fussball.xsd** der vorigen Aufgabe angewandt werden kann und ein XML-Dokument gemäß folgender DTD liefert:

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT vereine (verein*)>
<!ELEMENT verein (v_id, name, titel*)>
<!ELEMENT v_id (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT titel (bezeichnung, jahr)>
<!ELEMENT bezeichnung (#PCDATA)>
<!ELEMENT jahr (#PCDATA)>
```

#### Erläuterung zu dieser DTD:

- Die “verein”-Elemente entsprechen genau den “verein”-Elementen laut fussball.xsd.
- Die Subelemente “v\_id” und “name” eines “verein”-Elements entsprechen den Informationen des entsprechenden “verein”-Elements laut fussball.xsd.
- Die Subelemente “titel” eines “verein”-Elements entsprechen jenen “titel”-Elementen laut fussball.xsd, deren Fremdschlüssel “verein” auf die “v\_id” dieses Vereins zeigt.
- Die Subelemente “bezeichnung” und “jahr” eines “titel”-Elements entsprechen der Bewerb-Bezeichnung laut “b\_id” und dem Subelement “jahr” laut fussball.xsd.

**Beispiel:** Angewandt auf das XML-Dokument fussball.xml liefert dieses Stylesheet folgendes Ergebnis:

```
<?xml version="1.0" encoding="UTF-8"?>
<vereine>
  <verein>
    <v_id>1001</v_id>
    <name>FC Liverpool</name>
    <titel>
      <bezeichnung>Champions League</bezeichnung>
      <jahr>1984</jahr>
    </titel>
    <titel>
      <bezeichnung>Champions League</bezeichnung>
      <jahr>2005</jahr>
    </titel>
  </verein>
  <verein>
    <v_id>1002</v_id>
    <name>FC Barcelona</name>
    <titel>
      <bezeichnung>Champions League</bezeichnung>
      <jahr>2006</jahr>
    </titel>
    <titel>
      <bezeichnung>Primera Division</bezeichnung>
      <jahr>2006</jahr>
    </titel>
  </verein>
</vereine>
```

Datei **fussball2vereine.xsl**:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<!-- Ausgabe als XML -->
<xsl:output method="xml" indent="yes" encoding="UTF-8" version="1.0" />

<xsl:template match="fussball">
  <vereine>
    <xsl:apply-templates select="//vereine/verein" />
  </vereine>
</xsl:template>

<xsl:template match="verein">
<!-- Vervollständigen Sie dieses Template -->
  <verein>
    <v_id><xsl:value-of select="@v_id"/></v_id>
    <xsl:copy-of select = "name"/>
    <xsl:variable name="vid" select ="@v_id"/>
    <xsl:for-each select ="/fussball/titel[verein=$vid]">
      <xsl:variable name="bew" select ="bewerb"/>
      <titel>
        <xsl:copy-of select = "//bewerb[b_id=$bew]/bezeichnung"/>
        <xsl:copy-of select ="jahr" />
      </titel>
    </xsl:for-each>
  </verein>
</xsl:template>

</xsl:stylesheet>
```

**Aufgabe 5:**

(7)

Geben Sie die Events plus Inhalt (Namen und evtl. Text), wie sie von einem SAX Parser gefeuert werden, für das folgende XML Fragment an. Gehen Sie davon aus, dass jeweils nur ein characterEvent pro Textknoten gefeuert wird. Sie können die Events auch abkürzen (zB startElement = sE). Beispiel: "startElement wein" bzw. "sE wein"

```
<wein><sorte>Shiraz</sorte><?confidential-start?><frostschutz>10%</frostschutz><?confidential-end?></wein>
```

startElement	wein
startElement	sorte
characterEvent	Shiraz
endElement	sorte
processingInstruction	confidential-start
startElement	frostschutz
characterEvent	10%
endElement	frostschutz
processingInstruction	confidential-end
endElement	wein

**Aufgabe 6:**

(9)

Vervollständigen Sie die folgende Java Methode, so dass ein Node-Objekt zurückgegeben wird, das dem folgenden XML-Fragment entspricht.

```
<wein><sorte typ="rot">Shiraz</sorte></wein>
```

```
Node constructRebe(Document doc) throws Exception {
    Element wein = doc.createElement("wein");

    Element sorte = doc.createElement("sorte");
    sorte.setAttribute("typ", "rot");
    Text shiraz = doc.createTextNode("Shiraz");
    sorte.appendChild(shiraz);
    wein.appendChild(sorte);

    return wein;
}
```

**Aufgabe 7:**

(8)

(a) Schreiben Sie folgende XPath-Anfragen für das XML-Dokument **nachrichten.xml** von Aufgabe 1:

1. Alle Nachrichten, für die kein Redakteur angegeben wurde.

```
//orf:nachricht[not(redakteur)]
```

2. Alle Elemente namens "nachricht" im Namespace "http://www.orf.at/news". Sie dürfen sich bei dieser Anfrage nur auf den Namespace und nicht auf ein bestimmtes Präfix verlassen.

```
//node()[local-name()='nachricht' and namespace-uri()='http://www.orf.at/news']
```

(b) Schreiben Sie folgende XPath-Anfragen für das XML-Dokument **fussball.xml** von Aufgabe 3:

3. Anzahl der Bewerbe, deren Organisator den Teil-String 'Football' enthält.

```
count( //bewerbe/bewerb[contains(organisator,'Football')] )
```

4. Summe der Budgets aller Vereine, deren Budget in Euro angegeben ist.

```
sum(//verein/budget[@waehrung = 'Euro'])
```

**Aufgabe 8:**

(12)

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

1. In der XML Schema Datei fussball.xsd von Aufgabe 3 hätte man die Schlüssel auch innerhalb der complexType-Definitionen definieren können, d.h.: Primärschlüssel für "vereine" innerhalb von "vereineTyp", Primärschlüssel für "bewerbe" innerhalb von "bewerbeTyp", die beiden Fremdschlüssel innerhalb von "titelTyp". wahr  falsch

2. Das Element "budget" des XML-Dokuments fussball.xml von Aufgabe 3 könnte man in einer XML Schema Definition auf folgende Weise definieren:

```
<xsd:element name="budget" type="xsd:decimal">  
  <xsd:attribute name="waehrung" type="xsd:string"/>  
</xsd:element>
```

wahr  falsch 

3. Betrachten Sie das XSLT-Stylesheet fussball2vereine.xsl von Aufgabe 4. Wenn man in der apply-templates Anweisung das Attribut `select = "//vereine/verein"` durch `select = "*"`  ersetzt, so erhält man genau das gleiche Resultat.

wahr  falsch 

4. Betrachten Sie das XSLT-Stylesheet fussball2vereine.xsl von Aufgabe 4. Wenn man beim ersten template das Attribut `match = "fussball"` durch `match = "/"` ersetzt, so erhält man genau das gleiche Resultat.

wahr  falsch 

5. Für das XML-Dokument fussball.xml liefert die XPath-Anfrage `//titel/jahr = '1984'` den Boole'schen Wert true und die XPath-Anfrage `//titel/jahr != '1984'` den Boole'schen Wert false.

wahr  falsch 

6. Für das XML-Dokument fussball.xml liefert die XPath-Anfrage `//verein/@v_id = //bewerbe/bewerb/b_id` den Boole'schen Wert true.

wahr  falsch 

7. SAX verwendet die DOM API um ein XML Dokument zu lesen.

wahr  falsch 

8. Je größer das XML Dokument ist, desto größer ist der Speicherbedarf des SAX Parsers.

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)

Das folgende XML-Dokument **nachrichten.xml** gilt für die **Aufgaben 1 und 2**:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE nachrichten SYSTEM "nachrichten.dtd">
<orf:nachrichten xmlns:orf = "http://www.orf.at/news"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.orf.at/news nachrichten.xsd">

  <orf:nachricht bereich="sport">
    <nr>10005</nr>
    <titel>Ferrari auf Fehlersuche</titel>
    <datum>2007-06-18</datum>
    <redakteur>Heinz Pruessler</redakteur>
    <text> Hamilton zermuerbt Konkurrenz. <auto>Ferrari</auto> geraet im Kampf um die
      <schlagwort>WM 2007</schlagwort> endgueltig ins Hintertreffen. In Indianapolis waren
      die Italiener am Sonntag bereits im dritten Rennen in Folge gegen <auto>McLaren</auto>
      ohne Siegeschance. Nun wird fieberhaft gearbeitet, um das Kraefteverhaeltnis bis zum
      <schlagwort>Grand Prix</schlagwort> von Frankreich am 1. Juli wieder umzudrehen.
    </text>
  </orf:nachricht>

  <orf:nachricht bereich="sport">
    <nr>10015</nr>
    <titel>Abgang mit Polizeibegleitung</titel>
    <titel>Polizei holt Trainer vom Platz</titel>
    <datum>2007-06-19</datum>
    <text> Kuriose Szenen in der brasilianischen <schlagwort>Meisterschaft</schlagwort>:
      Weil er nach einer vermeintlichen Fehlentscheidung das Spielfeld nicht verlassen wollte,
      wurde America-Trainer <person>Lori Sandri</person> von der Polizei abgefuehrt.
    </text>
  </orf:nachricht>

</orf:nachrichten>
```

Das folgende XML-Dokument `fussball.xml` gilt für die **Aufgaben 3 und 4**:

```
<?xml version="1.0" encoding="UTF-8"?>
<fussball>
  <vereine>
    <verein v_id="1001">
      <name>FC Liverpool</name>
      <gegruendet>1892</gegruendet>
      <budget waehrung="Pfund">45100342.50</budget>
    </verein>
    <verein v_id="1002">
      <name>FC Barcelona</name>
      <gegruendet>1899</gegruendet>
      <budget waehrung="Euro">80100543.00</budget>
    </verein>
  </vereine>
  <bewerbe>
    <bewerb>
      <b_id>1001</b_id>
      <bezeichnung>Premier League</bezeichnung>
      <organisator>Football Association</organisator>
    </bewerb>
    <bewerb>
      <b_id>1053</b_id>
      <bezeichnung>Primera Division</bezeichnung>
      <organisator>Real Federacion de Futbol</organisator>
    </bewerb>
    <bewerb>
      <b_id>1100</b_id>
      <bezeichnung>Champions League</bezeichnung>
      <organisator>UEFA</organisator>
    </bewerb>
  </bewerbe>
  <titel>
    <verein>1001</verein>
    <bewerb>1100</bewerb>
    <jahr>1984</jahr>
  </titel>
  <titel>
    <verein>1002</verein>
    <bewerb>1100</bewerb>
    <jahr>2006</jahr>
  </titel>
  <titel>
    <verein>1001</verein>
    <bewerb>1100</bewerb>
    <jahr>2005</jahr>
  </titel>
  <titel>
    <verein>1002</verein>
    <bewerb>1053</bewerb>
    <jahr>2006</jahr>
  </titel>
</fussball>
```

Gesamtpunkte: 75