

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			21. 01. 2008
Kennnr.	Matrikelnr.	Familienname	Vorname

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

Aufgabe 1:

(9)

Betrachten Sie die folgende DTD **test.dtd**:

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT A (B*,C*)>
<!ELEMENT B (C+,D?)>
<!ATTLIST B F CDATA #REQUIRED>
<!ELEMENT C (#PCDATA|D|E)*>
<!ELEMENT D (#PCDATA)>
<!ELEMENT E EMPTY>
```

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

Hinweise:

- Gehen Sie davon aus, dass allen folgenden Dateien die Zeile
 <!DOCTYPE A SYSTEM "test.dtd">
 vorangestellt ist
- Sie können auch davon ausgehen, dass alle folgenden XML-Dateien wohlgeformt sind. Es geht also lediglich darum, ihre Gültigkeit bezüglich **test.dtd** zu entscheiden.

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

1. <A><C></C> gültig ungültig
2. <A><C><E/></C> gültig ungültig
3. <A><B F="abc"><C>def<D>ghj</D></C><C>abc<E>def</E>ghj</C> gültig ungültig
4. <A><C>abc<D>efg</D>hij</C> gültig ungültig
5. <A><C><D>a</D><D>b</D><D>c</D></C> gültig ungültig
6. <A><B F="abc">def gültig ungültig

(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:

(9)

Vervollständigen Sie **auf der nächsten Seite** in der XML Schema Definition **euro.xsd** die Definition des Wurzelements “info”, sodass das XML-Dokument **euro.xml** (siehe Anhang) bezüglich dieses Schemas gültig ist.

Berücksichtigen Sie beim Vervollständigen der XML Schema Definition folgende Punkte:

- Sie brauchen sich in diesem Beispiel **nicht** um die Definition der Typen “nationenTyp”, “stadienTyp” und “auslosungTyp” zu kümmern!
- Ergänzen Sie die Element-Definition von “info”, sodass die Subelemente richtig definiert werden.
 - Stellen Sie also sicher, dass innerhalb des “info”-Elements genau ein Element “euro:nationen”, genau ein Element “euro:stadien” und genau ein Element “auslosung” vorkommen. Nutzen Sie dazu die zur Verfügung stehenden Typen “nationenTyp”, “stadienTyp” und “auslosungTyp”.
 - Hinweis: Achten Sie auf das richtige Verwenden von Namespaces. Nutzen Sie hierzu das Attribut “form” von “<xs:element>”.
- Ergänzen Sie die Element-Definition von “info”, sodass folgende Schlüssel definiert werden:
 - Als Fremdschlüssel sind die Attribute “team1”, “team2”, und “spielort” des Elements “spiel” zu definieren.
 - Wählen Sie für die dazugehörigen Primärschlüssel entsprechende Attribute und/oder Elemente in “euro:nationen” und “euro:stadien”.
 - Bei der Definition der Schlüssel brauchen Sie sich **nicht** um Namespaces kümmern.

Datei euro.xsd:

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

  <xs:element name="info">

    <!-- Vervollständigen Sie hier diese Element-Definition. -->

    <xs:complexType>
      <xs:sequence>
        <xs:element name="nationen" type="euro:nationenTyp" form="qualified"/>
        <xs:element name="stadien" type="euro:stadienTyp" form="qualified"/>
        <xs:element name="auslosung" type="euro:auslosungTyp"/>
      </xs:sequence>
    </xs:complexType>

    <xs:key name="pk_teilnehmer">
      <xs:selector xpath="euro:nationen/teilnehmer"/>
      <xs:field xpath="@kuerzel"/>
    </xs:key>

    <xs:key name="pk_stadion">
      <xs:selector xpath="euro:stadien/stadion"/>
      <xs:field xpath="stadion_id"/>
    </xs:key>

    <xs:keyref name="fk_team1" refer="euro:pk_teilnehmer">
      <xs:selector xpath="auslosung/spiel"/>
      <xs:field xpath="@team1"/>
    </xs:keyref>

    <xs:keyref name="fk_team2" refer="euro:pk_teilnehmer">
      <xs:selector xpath="auslosung/spiel"/>
      <xs:field xpath="@team2"/>
    </xs:keyref>

    <xs:keyref name="fk_stadion" refer="euro:pk_stadion">
      <xs:selector xpath="auslosung/spiel"/>
      <xs:field xpath="@spielort"/>
    </xs:keyref>

  </xs:element>

  <!-- restliches File ist nicht Teil der Lösung dieser Prüfungsaufgabe! -->

  <xs:complexType name="nationenTyp"> ..... </xs:complexType>
  <xs:complexType name="stadienTyp"> ..... </xs:complexType>
  <xs:complexType name="auslosungTyp"> ..... </xs:complexType>
</xs:schema>
```

Aufgabe 3:

(9)

Vervollständigen Sie **auf der nächsten Seite** in der XML Schema Definition **euro.xsd** nun die complexType-Definition des Typs “nationenTyp”, sodass das XML-Dokument **euro.xml** (siehe Anhang) bezüglich dieses Schemas gültig ist.

Berücksichtigen Sie dabei folgende Punkte:

- Es können beliebig viele “teilnehmer”-Elemente vorkommen, zumindest jedoch eines.
- Ein Teilnehmer ist entweder fix qualifiziert (dies wird mit einem leeren Element “fix” angegeben) oder musste sich gegen 6 oder 7 “gegner” in einer Gruppe qualifizieren (diese werden als “gegner”-Elemente aufgelistet).
- Pro Teilnehmer gibt es genau ein “kurzinfo”-Element. Dieses hat gemischten Inhalt. Als Subelemente können Elemente “spieler” und “land” beliebig oft auftreten.
- Alle Attribute, die verwendet werden, sollen verpflichtend sein.

Datei euro.xsd:

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

  <xs:element name="info"> ... </xs:element>

  <xs:complexType name="nationenTyp">

  <!-- Vervollständigen Sie nur diese complexType-Definition -->

  <xs:sequence>
    <xs:element name="teilnehmer" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:choice>
            <xs:element name="fix">
              <xs:complexType/>
            </xs:element>
            <xs:element name="gegner" type="xs:string" minOccurs="6" maxOccurs="7"/>
          </xs:choice>
          <xs:element name="kurzinfo">
            <xs:complexType mixed="true">
              <xs:choice maxOccurs="unbounded">
                <xs:element name="spieler" type = "xs:string" />
                <xs:element name="land" type = "xs:string" />
              </xs:choice>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
        <xs:attribute name="bezeichnung" type="xs:string" use="required"/>
        <xs:attribute name="kuerzel" type="xs:string" use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>

  </xs:complexType>

  <xs:complexType name="stadientyp"> ..... </xs:complexType>
  <xs:complexType name="auslosungTyp"> .... </xs:complexType>
</xs:schema>
```

Aufgabe 4:

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

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT spielplan (stadion*)>
<!ELEMENT stadion (#PCDATA|spiel)*>
<!ELEMENT spiel (#PCDATA)>
```

Erläuterungen zu dieser DTD:

- Das gesuchte Stylesheet soll für jedes Stadion die dort stattfindenden Spiele ausgeben.
- Das “stadion”-Element dieser DTD hat gemischten Inhalt, wobei als Text der Inhalt des “name”-Subelements des Elements “stadion” in **euro.xml** herangezogen wird.
- Beachten Sie, dass der Text im Element “spiel” dieser DTD aus Inhalten des “teilnehmer”-Attributs “bezeichnung” aus der Input-datei **euro.xml**, sowie dem Wort “gegen” als zusätzlichem Text zusammengesetzt ist.
- Hinweis: Die benötigten Verknüpfungen zwischen Stadien und den Bezeichnungen der Teilnehmer erhalten Sie aus den “spiel”-Elementen in **euro.xml** durch Verknüpfung mit den “kuerzel”-Attributen der “teilnehmer”-Elemente.

Beispiel: Angewandt auf das XML-Dokument **euro.xml** soll **euro.xsl** daher folgendes Ergebnis liefern (Anmerkung: die Einrückungen dienen nur zur besseren Lesbarkeit und müssen nicht berücksichtigt werden).

```
<?xml version="1.0" encoding="UTF-8"?>
<spielplan>
<stadion>
  Ernst-Happel Stadion
  <spiel>Oesterreich gegen Kroatien</spiel>
  <spiel>Oesterreich gegen Polen</spiel>
  <spiel>Oesterreich gegen Deutschland</spiel>
</stadion>
<stadion>
  Woertherseestadion
  <spiel>Deutschland gegen Polen</spiel>
  <spiel>Kroatien gegen Deutschland</spiel>
  <spiel>Polen gegen Kroatien</spiel>
</stadion>
<stadion>
  Stadion Wals-Siezenheim
</stadion>
<stadion>
  Tivoli NEU
</stadion>
</spielplan>
```

Datei euro.xsl:

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

<xsl:template match="/">

<!-- Vervollständigen Sie dieses Template -->

  <spielplan>
    <xsl:for-each select="//stadion">
      <stadion>
        <xsl:value-of select="name"/>
        <xsl:variable name="id" select ="stadion_id"/>
        <xsl:for-each select ="//spiel[@spielort=$id]">
          <spiel>
            <xsl:variable name="team1" select ="@team1"/>
            <xsl:variable name="team2" select ="@team2"/>
            <xsl:value-of select="//teilnehmer[@kuerzel=$team1]/@bezeichnung"/>
            <xsl:text> gegen </xsl:text>
            <xsl:value-of select="//teilnehmer[@kuerzel=$team2]/@bezeichnung"/>
          </spiel>
        </xsl:for-each>
      </stadion>
    </xsl:for-each>
  </spielplan>

</xsl:template>

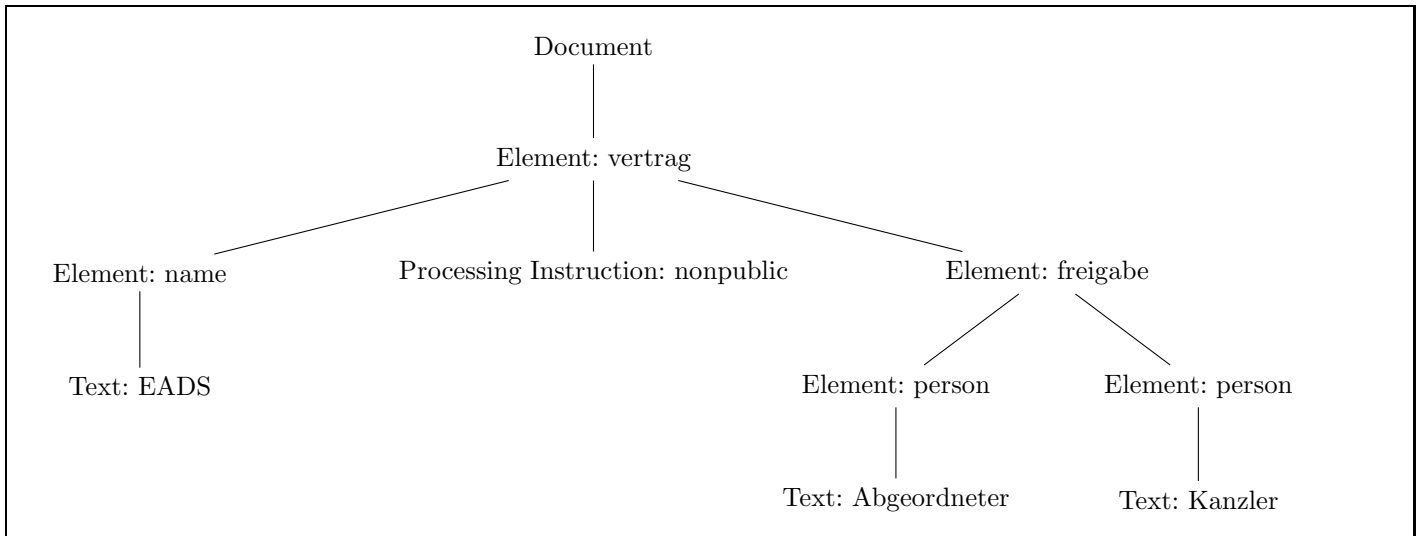
</xsl:stylesheet>
```

Aufgabe 5:

(8)

Zeichnen Sie den DOM-Baum zum folgenden XML **Dokument**. Schreiben Sie zu jedem Knoten den Knotentyp und den Inhalt (zB: "Element: elementname" oder "Text: textinhalt").

```
<vertrag><name>EADS</name><?nonpublic?><freigabe><person>Abgeordneter</person><person>Kanzler</person></freigabe></vertrag>
```

**Aufgabe 6:**

(8)

Vervollständigen Sie die folgende Java Klasse sodass ein SAX Content-Handler herauskommt, der alle *Elementnamen* (ohne Namespaces) und die *Anzahl der Attribute* dieser Elemente auf die Konsole ausgibt. Sie brauchen sich nicht um eine Fehlerbehandlung kümmern; Die Elemente sollen in Document-Order ausgegeben werden.

Beispiel: `startelement: 3`

```
class ElementPrinterHandler extends DefaultHandler {

    public void startElement(String namespaceURI,
        String localName,
        String qName,
        Attributes atts) throws SAXException {
        super.startElement(namespaceURI, localName, qName, atts);

        System.out.println(localName + ": " + atts.getLength() + "\n");
    }
}
```


Aufgabe 7:

(8)

Schreiben Sie XPath-Anfragen um für das XML-Dokument **euro.xml** (siehe Anhang) folgende Informationen zu selektieren:

1. Anzahl der Stadien, die mehr als 20000 Plätze haben.
`count(//stadion[plaetze > 20000])`
2. Alle Teilnehmer, die nicht fix qualifiziert sind (d.h., die kein Subelement "fix" enthalten).
`//teilnehmer[not(fix)]`
3. Kurzinfos in denen das Wort "EM" vorkommt.
`//kurzinfo[contains(., 'EM')]`
4. Die Namen der Stadien, für die zumindest ein Spiel (im Element "auslosung") existiert.
`//stadion[stadion_id=//spiel/@spielort]/name`

Aufgabe 8:

(12)

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

1. Angenommen ein XSLT-Stylesheet enthält folgendes Template:
`<xsl:template match="text()"></xsl:template>`
Dann ist es trotzdem möglich, Textknoten vom Quelldokument in das Zieldokument zu kopieren. wahr falsch
2. Jedes XSLT-Stylesheet muss mindestens ein Template enthalten. wahr falsch
3. Je größer das XML Dokument ist, desto größer ist der Speicherbedarf bei Verwendung eines DOM Parsers. wahr falsch
4. SAX ist im allgemeinen Speicher-effizienter als DOM. wahr falsch
5. Für das XML-Dokument **euro.xml** liefert die XPath-Anfrage `//teilnehmer/@kuerzel != //spiel/@team1` den Boole'schen Wert false. wahr falsch
6. Für das XML-Dokument **euro.xml** liefert die XPath-Anfrage `//spiel/@team1 = //spiel/@team2` den Boole'schen Wert false. wahr falsch
7. Angenommen Sie wollen in einer DTD festlegen, dass im XML-Dokument **euro.xml** die `stadion_id` Elemente eindeutige Werte haben. Dann könnte die Element-Deklaration folgendermaßen aussehen: `<!ELEMENT stadion_id ID>`. wahr falsch
8. Nehmen Sie an, dass in einer DTD 2 Primärschlüssel mittels ID-Attribut definiert wurden. Wenn Sie in derselben DTD auch einen Fremdschlüssel definieren wollen, so können Sie mittels IDREF-Attribut festlegen, auf welchen der beiden Primärschlüssel dieser Fremdschlüssel verweisen soll. 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 **euro.xml** gilt für Aufgaben 2-4, 7 und 8.

```
<?xml version="1.0" encoding="UTF-8"?>
<euro:info
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:euro = "http://www.oefb.at"
  xsi:schemaLocation="http://www.oefb.at euro.xsd">

  <euro:nationen>
    <teilnehmer bezeichnung="Oesterreich" kuerzel="AUT">
      <fix/>
      <kurzinfo>
        Als Veranstalter ist Oesterreich fix qualifiziert. Kapitaen
        des Teams wird voraussichtlich <spieler>Andreas Ivanschitz</spieler> sein.
      </kurzinfo>
    </teilnehmer>
    <teilnehmer bezeichnung="Kroatien" kuerzel="CRO">
      <gegner>Russland</gegner>
      <gegner>England</gegner>
      <gegner>Israel</gegner>
      <gegner>FYRO Mazedonien</gegner>
      <gegner>Estland</gegner>
      <gegner>Andorra</gegner>
      <kurzinfo>
        Die Kroaten konnten sich in einer starken Qualifikations-Gruppe
        durchsetzen und eliminierten dabei u.a. <land>England</land>.
        Gruppenzweiter wurde <land>Russland</land>.
      </kurzinfo>
    </teilnehmer>
    <teilnehmer bezeichnung="Deutschland" kuerzel="GER">
      <gegner>Tschechische Republik</gegner>
      <gegner>Irland</gegner>
      <gegner>Slowakei</gegner>
      <gegner>Wales</gegner>
      <gegner>Zypern</gegner>
      <gegner>San Marino</gegner>
      <kurzinfo>
        Souveraen fuer die EM qualifizierten sich auch
        <spieler>Miroslav Klose</spieler>, <spieler>Philipp Lahm</spieler>, und Co.
        Gruppensieger war allerdings die <land>Tschechische Republik</land>.
      </kurzinfo>
    </teilnehmer>
    <teilnehmer bezeichnung="Polen" kuerzel="POL">
      <gegner>Portugal</gegner>
      <gegner>Serbien</gegner>
      <gegner>Finnland</gegner>
      <gegner>Belgien</gegner>
      <gegner>Kasachstan</gegner>
      <gegner>Armenien</gegner>
      <gegner>Aserbaidshchan</gegner>
      <kurzinfo>
        Ebenfalls mit dabei ist Polen und somit auch der Austrianer
        <spieler>Jacek Bak</spieler>.
      </kurzinfo>
    </teilnehmer>
  </euro:nationen>

<!-- BITTE WENDEN. Fortsetzung auf Rueckseite -->
```

<!-- FORTSETZUNG des Dokuments euro.xml -->

```
<euro:stadien>
  <stadion>
    <stadion_id>1</stadion_id>
    <name>Ernst-Happel Stadion</name>
    <ort>Wien</ort>
    <plaetze>50000</plaetze>
  </stadion>
  <stadion>
    <stadion_id>2</stadion_id>
    <name>Woertherseestadion</name>
    <ort>Klagenfurt</ort>
    <plaetze>30000</plaetze>
  </stadion>
  <stadion>
    <stadion_id>3</stadion_id>
    <name>Stadion Wals-Siezenheim</name>
    <ort>Salzburg</ort>
    <plaetze>30000</plaetze>
  </stadion>
  <stadion>
    <stadion_id>4</stadion_id>
    <name>Tivoli NEU</name>
    <ort>Innsbruck</ort>
    <plaetze>30000</plaetze>
  </stadion>
</euro:stadien>

<auslosung>
  <spiel team1="AUT" team2="CRO" spielort="1" datum="2008-06-08" anpfiff="18:00"/>
  <spiel team1="GER" team2="POL" spielort="2" datum="2008-06-08" anpfiff="20:45"/>
  <spiel team1="CRO" team2="GER" spielort="2" datum="2008-06-12" anpfiff="18:00"/>
  <spiel team1="AUT" team2="POL" spielort="1" datum="2008-06-12" anpfiff="20:45"/>
  <spiel team1="POL" team2="CRO" spielort="2" datum="2008-06-16" anpfiff="20:45"/>
  <spiel team1="AUT" team2="GER" spielort="1" datum="2008-06-16" anpfiff="20:45"/>
</auslosung>
</euro:info>
```

Gesamtpunkte: 75