

## 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			30. 11. 2007
Kennnr.	Matrikelnr.	Familienname	Vorname

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

### Aufgabe 1:

(12)

Vervollständigen Sie die DTD **oper.dtd**, sodass das XML-Dokument **oper.xml** (im Anhang) bezüglich dieser DTD gültig ist. Berücksichtigen Sie beim Erstellen der DTD folgende Punkte:

- Die Verknüpfung mit einer XML Schema-Definition ist optional; alle anderen Attribute, die Sie verwenden, sollen verpflichtend sein.
- Es soll sichergestellt sein, dass zumindest ein "werk"-Element und zumindest ein "auffuehrung"-Element vorhanden ist. Jedes Werk hat entweder genau einen Komponisten (spezifiziert durch das Element "komposition") oder aber einen oder mehrere Choreographen (spezifiziert durch Element(e) "choreographie");
- Ein "beschreibung"-Element hat gemischten Inhalt. Als Subelemente können die Elemente "stichwort" und "rolle" beliebig oft und in beliebiger Reihenfolge auftreten.
- Pro Element "besetzung" kommen entweder ausschließlich Sängerinnen ("saengerin") und Sänger ("saenger") oder ausschließlich Tänzerinnen ("taenzerin") und Tänzer ("taenzer") vor; jeweils gefolgt von der dargestellten Rolle ("rolle"). Es soll zumindest eine solche Rollenzuteilung fuer jede Aufführung gegeben sein.

#### Datei **oper.dtd**:

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT werk_id (#PCDATA)>
<!ELEMENT name (#PCDATA)>      <!ELEMENT dirigent (#PCDATA)>
<!ELEMENT komposition (#PCDATA)> <!ELEMENT choreographie (#PCDATA)>
<!ELEMENT rolle (#PCDATA)>      <!ELEMENT stichwort (#PCDATA)>
<!ELEMENT saenger (#PCDATA)>    <!ELEMENT saengerin (#PCDATA)>
<!ELEMENT taenzer (#PCDATA)>    <!ELEMENT taenzerin (#PCDATA)>

<!ELEMENT staatsoper:programm (repertoire,spielplan)>
<!ATTLIST staatsoper:programm xmlns:staatsoper CDATA #REQUIRED
  xmlns:xsi CDATA #IMPLIED
  xsi:schemaLocation CDATA #IMPLIED>
<!ELEMENT repertoire (werk+)>
<!ELEMENT werk (name,(komposition|(choreographie+)),beschreibung)>
<!ATTLIST werk id CDATA #REQUIRED genre CDATA #REQUIRED>
<!ELEMENT beschreibung (#PCDATA | rolle | stichwort)*>
<!ELEMENT spielplan (auffuehrung+)>
<!ELEMENT auffuehrung (werk_id,dirigent,besetzung)>
<!ATTLIST auffuehrung datum CDATA #REQUIRED>
<!ELEMENT besetzung (((saenger|saengerin),rolle)+)|(((taenzer|taenzerin),rolle)+)>
```

Vervollständigen Sie auf der nächsten Seite die XML Schema Definition **oper.xsd**, sodass das XML-Dokument **oper.xml** (siehe Anhang) bezüglich dieses Schemas gültig ist. Berücksichtigen Sie dabei:

- Ergänzen Sie die Element-Definition von “programm” und verwenden Sie dafür die Typen “repertoireTyp” und “spielplanTyp”. Weiters soll das Attribut “id” des Elements “werk” als ein Primärschlüssel definiert werden.
- Ergänzen Sie die Typ-Definition von “repertoireTyp”. Es gilt (wie bei Beispiel 1):
  - Es soll sichergestellt sein, dass zumindest ein “werk”-element vorhanden ist.
  - Jedes Werk hat entweder genau einen Komponisten (spezifiziert durch das Element “komposition”) oder einen oder mehrere Choreographen (spezifiziert durch Element(e) “choreographie”);
  - Ein “beschreibung”-Element hat gemischten Inhalt. Als Subelemente können die Elemente “stichwort” und “rolle” beliebig oft und in beliebiger Reihenfolge auftreten.
  - Alle Attribute, die Sie verwenden, sollen verpflichtend sein.
- Sie brauchen sich **nicht um die Definition** des Typs “spielplanTyp” zu **kümmern!**

Datei oper.xsd:

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

  <xs:element name="programm">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="repertoire" type="oper:repertoireTyp"/>
        <xs:element name="spielplan" type="oper:spielplanTyp"/>
      </xs:sequence>
    </xs:complexType>
    <xs:key name="pk_werk">
      <xs:selector xpath="repertoire/werk"/>
      <xs:field xpath="@id"/>
    </xs:key>
  </xs:element>

  <xs:complexType name="repertoireTyp">
    <xs:sequence>
      <xs:element name="werk" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="name" type="xs:string"/>
            <xs:choice>
              <xs:element name="komposition" type="xs:string"/>
              <xs:element name="choreographie" type="xs:string" maxOccurs="unbounded"/>
            </xs:choice>
            <xs:element name="beschreibung">
              <xs:complexType mixed="true">
                <xs:choice maxOccurs="unbounded">
                  <xs:element name="stichwort" type="xs:string" />
                  <xs:element name="rolle" type="xs:string" />
                </xs:choice>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="id" type="xs:integer" use="required"/>
          <xs:attribute name="genre" type="xs:string" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>

  <!-- restliches Schema ist nicht Teil der Prüfungsaufgabe! -->

  <xsd:complexType name="spielplanTyp"> ..... </xsd:spielplanTyp>

</xs:schema>
```



Betrachten Sie die folgenden XSLT-Stylesheets. Geben Sie jeweils den Output an, den das entsprechende Stylesheet angewandt auf **oper.xml** (siehe Anhang) liefert. Sie brauchen sich dabei nicht um Whitespaces etc. kümmern.

Anmerkung: Pro Teilaufgabe sind jeweils 5 Punkte erreichbar.

Datei **query1.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="/">
    <xsl:for-each select="//werk[@genre='Oper']">
      <xsl:element name="titel"> <xsl:value-of select="name"/> </xsl:element>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

Vervollständigen Sie hier den Output von **query1.xsl** angewandt auf **oper.xml**:

```
<?xml version="1.0" encoding="UTF-8"?>

<titel>Tosca</titel>
<titel>La traviata</titel>
```

Datei **query2.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="/">
    <xsl:element name="beschreibung"> <xsl:apply-templates/> </xsl:element>
  </xsl:template>

  <xsl:template match="//werk[@id='4']">
    <xsl:value-of select="."/>
  </xsl:template>

  <xsl:template match="text()">
  </xsl:template>
</xsl:stylesheet>
```

Vervollständigen Sie hier den Output von **query2.xsl** angewandt auf **oper.xml**:

```
<?xml version="1.0" encoding="UTF-8"?>

<beschreibung> Schwanensee Marius Petipa Prinz Siegfried ist grossjaehrig geworden.... </beschreibung>
```

### Fortsetzung Beispiel 3.

Datei **query3.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="/">
    <xsl:for-each select="//auffuehrung">
      <xsl:variable name="id" select="werk_id"/>
      <xsl:if test="position()=last()">
        <xsl:value-of select="@datum"/>
        <xsl:text>: </xsl:text>
        <xsl:value-of select="//werk[@id=$id]/name"/>
        <xsl:value-of select="besetzung"/>
      </xsl:if>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>
```

Vervollständigen Sie hier den Output von **query3.xsl** angewandt auf **oper.xml**:

```
<?xml version="1.0" encoding="UTF-8"?>

2008-01-11: La traviata
Norah Amsellem Violetta Valery
Joseph Calleja Alfredo Germont
```

#### Aufgabe 4:

(7)

Geben Sie die Eventnamen plus relevanten Inhalt (z.B. 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 (also der gesamte Textinhalt auf einmal übergeben wird). Sie können die Events auch abkürzen (zB characterEvent = cE). Beispiel: "characterEvent beispieltext" bzw. "cE beispieltext"

```
<wort lang="de"><!--not widely known-->tantieme</wort><wort lang="en">royalty</wort>
```

```
startElement    wort [lang: de]
characterEvent  tantieme
endElement      wort
startElement    wort [lang: en]
characterEvent  royalty
endElement      wort
```

#### Aufgabe 5:

(9)

Vervollständigen Sie die folgende Java Methode sodass alle Attribute (plus Inhalt) eines übergebenen XML Dokuments auf die Konsole ausgegeben werden (zB "attributname: inhalt").

- Sie müssen **keine Fehlerbehandlung** implementieren.
- Die Reihenfolge in der die Attribute ausgegeben werden ist **nicht relevant**.

Tipp: Navigieren Sie **rekursiv** mithilfe der Methoden aus den Folien (DOM Seiten 19, 28, 33, 34) durch den Baum.

```
void printAttributes(Node node) throws Exception {

    NodeList nl = node.getChildNodes();
    for(int i=0; i < nl.getLength(); ++i) {
        Node n = nl.item(i);
        if(n.hasAttributes()) {
            NamedNodeMap nnm = n.getAttributes();
            for(int j=0; j < nnm.getLength(); ++j) {
                Node a = nnm.item(j);
                System.out.println(a.getNodeName() + ": " + a.getTextContent());
            }
        }
        printAttributes(n);
    }
}
```

**Aufgabe 6:**

(8)

Schreiben Sie XPath-Anfragen um für das XML-Dokument **oper.xml** von Aufgabe 1 folgende Informationen zu selektieren:

1. Alle Werke, deren Beschreibung mindestens ein "rolle"-Subelement und mindestens ein "stichwort"-Subelement enthält.  
`//werk[beschreibung[rolle and stichwort]]`
2. Alle Aufführungen, in deren Besetzung mindestens 3 TänzerInnen vorkommen (d.h.: Summe aus "taenzer"- und "taenzerin"-Elementen  $\geq 3$ ).  
`//auffuehrung[count(besetzung/taenzer) + count(besetzung/taenzerin) >= 3 ]`
3. Alle Elemente aus dem Namespace `http://www.staatsoper.at`.  
`//*[namespace-uri() = 'http://www.staatsoper.at']`
4. Alle "saenger"- und "saengerin"-Elemente bei Aufführungen im Jahr 2007 (Tipp: Nehmen Sie an, dass eine Aufführung im Jahr 2007 stattfindet, wenn das datum-Attribut den Substring '2007' enthält).  
`//auffuehrung[contains(@datum, '2007')]//saenger |  
//auffuehrung[contains(@datum, '2007')]//saengerin`

**Aufgabe 7:**

(12)

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

1. Angenommen im XSLT-Stylesheet **query1.xsl** von Aufgabe 2 wird folgendes Template *hinzugefügt*:  
`<xsl:template match="text()"></xsl:template>`  
Dann bleibt der Output bei Anwendung des Stylesheets auf die Datei **oper.xml** unverändert.    wahr  falsch
2. Angenommen im XSLT-Stylesheet **query2.xsl** von Aufgabe 2 wird folgendes Template *gelöscht*:  
`<xsl:template match="text()">  
</xsl:template>`  
Dann bleibt der Output bei Anwendung des Stylesheets auf die Datei **oper.xml** unverändert.    wahr  falsch
3. Angenommen Sie wollen das Element `werk_id` als Fremdschlüssel (auf das `id`-Attribut der `werk`-Elemente) definieren. Dann müsste in der DTD **oper.dtd** von Aufgabe 1 die Element-Deklaration von `werk_id` folgendermaßen aussehen:  
`<!ELEMENT werk_id (IDREF)>`.    wahr  falsch
4. Angenommen Sie wollen das Element `werk_id` als Fremdschlüssel (auf das `id`-Attribut der `werk`-Elemente) definieren. Dann könnte in einer XML Schema Definition die Element-Deklaration von `werk_id` folgendermaßen aussehen:  
`<xs:element name='werk_id' type='xs:integer' keyref='true'/>`.    wahr  falsch
5. Um die Datei **oper.xml** mit einer XML Schema Definition zu verknüpfen, dürften die Präfixe im `programm`-Element auch folgendermaßen umgeschrieben werden:  
`<abc:programm  
  xmlns:abc = "http://www.staatsoper.at"  
  xmlns:xyz = "http://www.w3.org/2001/XMLSchema-instance"  
  xyz:schemaLocation = "http://www.staatsoper.at oper.xsd">`    wahr  falsch
6. Für das XML-Dokument **oper.xml** liefert folgende XPath-Anfrage eine leere Knotenmenge:  
`//spielplan[contains(../dirigent, 'Haider')]`    wahr  falsch
7. Für das XML-Dokument **oper.xml** liefert folgende XPath-Anfrage den Boole'schen Wert `true`:  
`//werk/@id = //werk_id and //werk/@id != //werk_id`    wahr  falsch
8. Bei einem "Pull-Parser" kontrolliert die Applikation den Programmablauf.    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 **oper.xml** gilt für **Aufgaben 1–3**:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE programm SYSTEM "oper.dtd">
<staatsoper:programm
  xmlns:staatsoper = "http://www.staatsoper.at"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.staatsoper.at oper.xsd">
  <repertoire>
    <werk id="1" genre="Oper">
      <name>Tosca</name>
      <komposition>Giacomo Puccini</komposition>
      <beschreibung>
        Rom um 1800. Die Stadt stoehnt unter einem Schreckensregime. Der
        revolutionaer gesinnte Maler <rolle>Mario Cavaradossi</rolle> ist beauftragt in
        der <stichwort>Kirche Sant'Andrea della Valle</stichwort> ein Maria Magdalena
        Gemaelde zu schaffen.
        Der aus seiner Gefangenschaft geflohene Republikaner Angelotti aus der Familie
        der Attavanti, hat sich in dieser Kirche verborgen und macht sich nun Mario
        gegenueber bemerkbar. Dieser nennt ihm ein sicheres Versteck auf seinem Besitz.
        Kaum ist Angelotti weitergeflohen, betritt der grausame <rolle>Polizeichef
        Scarpia</rolle> die Kirche. In der offengelassenen Grabkapelle der Attavanti und
        findet er den Faecher der Schwester Angelottis. Sofort ist im klar, dass Angelotti
        in der Kirche zuflucht gesucht haben musste. Er weiss von der Freundschaft
        Cavaradossis und Angelottis. Ausserdem will er Cavaradossis Geliebte,
        die Saengerin <rolle>Floria Tosca</rolle>, fuer sich gewinnen...
      </beschreibung>
    </werk>
    <werk id="2" genre="Ballet">
      <name>Der Nussknacker</name>
      <choreographie>Gyula Harangozo</choreographie>
      <choreographie>Wassili Wainonen</choreographie>
      <beschreibung>
        Familie Stahlbaum und deren Gaeste feiern das <stichwort>Weihnachtsfest</stichwort>.
        Zur Unterhaltung seiner Patenkinder <rolle>Maria</rolle> und Fritz fuehrt
        <rolle>Drosselmeyer</rolle> das neueste Computerspiel vom Creepyman und einem
        Prinzenpaar vor und laesst Puppen tanzen. Maria erhaelt von Drosselmeyer
        einen <rolle>Nussknacker</rolle>...
      </beschreibung>
    </werk>
    <werk id="3" genre="Oper">
      <name>La traviata</name>
      <komposition>Giuseppe Verdi</komposition>
      <beschreibung>
        <rolle>Violetta Valery</rolle>, strahlender Mittelpunkt der Pariser Demimonde, gibt -
        nach einer Krankheit scheinbar wiedergenesen - in ihrem Haus ein Fest, bei dem sich
        viele Herren der guten Gesellschaft einfinden. Einer aus der Schar ihrer Verehrer,
        Gaston, fuehrt seinen Freund <rolle>Alfredo Germont</rolle>, auf den die umworbene
        Frau tiefen Eindruck gemacht hat, in deren Salon ein...
      </beschreibung>
    </werk>
    <werk id="4" genre="Ballet">
      <name>Schwanensee</name>
      <choreographie>Marius Petipa</choreographie>
      <beschreibung>
        <rolle>Prinz Siegfried</rolle> ist grossjaehrig geworden...
      </beschreibung>
    </werk>
  </repertoire>
<!-- BITTE WENDEN. Forsetzung auf Rueckseite -->
```

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

```
<spielplan>
  <auffuehrung datum="2007-12-04">
    <werk_id>1</werk_id>
    <dirigent>Paolo Carignani</dirigent>
    <besetzung>
      <saengerin>Amarilli Nizza</saengerin> <rolle>Floria Tosca</rolle>
      <saenger>Jose Cura</saenger> <rolle>Mario Cavaradossi</rolle>
      <saenger>Marco Vratogna</saenger> <rolle>Baron Scarpia</rolle>
    </besetzung>
  </auffuehrung>
  <auffuehrung datum="2007-12-08">
    <werk_id>3</werk_id>
    <dirigent>Friedrich Haider</dirigent>
    <besetzung>
      <saengerin>Norah Amsellem</saengerin> <rolle>Violetta Valery</rolle>
      <saenger>Joseph Calleja</saenger> <rolle>Alfredo Germont</rolle>
    </besetzung>
  </auffuehrung>
  <auffuehrung datum="2007-12-18">
    <werk_id>2</werk_id>
    <dirigent>Sascha Goetzel</dirigent>
    <besetzung>
      <taenzerin>Olga Esina</taenzerin> <rolle>Maria</rolle>
      <taenzer>Roman Lazik</taenzer> <rolle>Nussknacker</rolle>
      <taenzer>Kirill Kourlaev</taenzer> <rolle>Drosselmeyer</rolle>
    </besetzung>
  </auffuehrung>
  <auffuehrung datum="2007-12-28">
    <werk_id>2</werk_id>
    <dirigent>Sascha Goetzel</dirigent>
    <besetzung>
      <taenzerin>Irina Tsymbal</taenzerin> <rolle>Maria</rolle>
      <taenzer>Rainer Krenstetter</taenzer> <rolle>Nussknacker</rolle>
      <taenzer>Eno Peci</taenzer> <rolle>Drosselmeyer</rolle>
    </besetzung>
  </auffuehrung>
  <auffuehrung datum="2008-01-11">
    <werk_id>3</werk_id>
    <dirigent>Friedrich Haider</dirigent>
    <besetzung>
      <saengerin>Norah Amsellem</saengerin> <rolle>Violetta Valery</rolle>
      <saenger>Joseph Calleja</saenger> <rolle>Alfredo Germont</rolle>
    </besetzung>
  </auffuehrung>
</spielplan>
</staatsoper:programm>
```

Gesamtpunkte: 75