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**			23. 01. 2008
Kennnr.	Matrikelnr.	Familienname	Vorname

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

Aufgabe 1: (9)

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

```
<!ELEMENT A ((B+,C+)|(C+,B*))>
<!ELEMENT B (#PCDATA|C)*>
<!ELEMENT C (#PCDATA|D|E)*>
<!ELEMENT D (#PCDATA)>
<!ATTLIST D pr ID #IMPLIED>
<!ELEMENT E (#PCDATA)>
<!ATTLIST E fo IDREF #REQUIRED>
```

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><B>abc</B><B>def</B></A>` gültig ◯ ungültig ⊗
2. `<A><C>abc</C><C>def</C></A>` gültig ⊗ ungültig ◯
3. `<A><B>abc<C>def</C>ghi</B><C>jkl</C></A>` gültig ⊗ ungültig ◯
4. `<A><C>abc<B>def</B>ghi</C><B>jkl</B></A>` gültig ◯ ungültig ⊗
5. `<A><C><D>abc</D></C>def<C>ghi</C></A>` gültig ◯ ungültig ⊗
6. `<A><C><D pr="_1">abc</D><E fo="_1">def</E></C></A>` gültig ⊗ ungültig ◯

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

Aufgabe 2: (9)

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

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="A">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:choice minOccurs="0" maxOccurs="1">
          <xsd:element name="B" type="xsd:string"/>
          <xsd:element name="C" type="xsd:string"/>
        </xsd:choice>
        <xsd:element name = "D" type="Dtype" minOccurs="0" maxOccurs="1"/>
       </xsd:sequence>
    </xsd:complexType>
  </xsd:element>
  <xsd:complexType name="Dtype">
    <xsd:all>
       <xsd:element name="E" type="xsd:string"/>
       <xsd:element name="F" type="xsd:string"/>
    </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><C>abc</C><D><E>def</E></D></A>` gültig ◯ ungültig ⊗
2. `<A><D><F>abc</F><E>def</E></D></A>` gültig ⊗ ungültig ◯
3. `<A><B>abc</B><C>def</C><D><E>ghi</E><F>jkl</F></D></A>` gültig ◯ ungültig ⊗
4. `<A><D><E>abc</E><F>def</F><E>ghi</E></D></A>` gültig ◯ ungültig ⊗
5. `<A/>` gültig ⊗ ungültig ◯
6. `<A><B>abc</B><D>def</D></A>` gültig ◯ ungültig ⊗

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

Aufgabe 3: (9)

Geben Sie eine DTD **dancing.dtd** an, sodass das XML-Dokument **dancing.xml** (siehe Anhang) bezüglich dieser DTD gültig ist. Berücksichtigen Sie beim Erstellen der DTD folgende Punkte:

- Das Wurzelelement beinhaltet eine beliebige Anzahl von "paar"-Elementen (jedoch zumindest eines).
- Das "paar"-Element beschreibt ein Tanzpaar. Ein solches besteht aus genau einem "promi" sowie entweder einem "partner" oder einer "partnerin". Beachten Sie, dass die Reihenfolge jedoch nicht fixiert ist!
- Alle Attribute sollen verpflichtend sein.

Datei **dancing.dtd**:

```
<!ELEMENT dancingstars (paar)+>
<!ELEMENT paar
   (((partnerin|partner),promi)|(promi,(partnerin|partner)))>
<!ATTLIST paar
    nr CDATA #REQUIRED
    ergebnis CDATA #REQUIRED>
<!ELEMENT partnerin (#PCDATA)>
<!ELEMENT partner (#PCDATA)>
<!ELEMENT promi (#PCDATA)>
```

Aufgabe 4: (12)

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

Anmerkung: Pro Teilaufgabe sind jeweils 4 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="//paar">
      <xsl:if test="position()!=last()">
         <xsl:value-of select="@nr"/>
      </xsl:if>
    </xsl:for-each>
</xsl:template>

</xsl:stylesheet>
```

Geben Sie hier den Output von **query1.xsl** angewandt auf **dancing.xml** an:

```
123456789
```

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:apply-templates select="//paar[last()]"/>
</xsl:template>

</xsl:stylesheet>
```

Geben Sie hier den Output von **query2.xsl** angewandt auf **dancing.xml** an:

```
Andy Kainz
Elke Winkens
```

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="/dancingstars/paar[@nr!=10]">
</xsl:template>

</xsl:stylesheet>
```

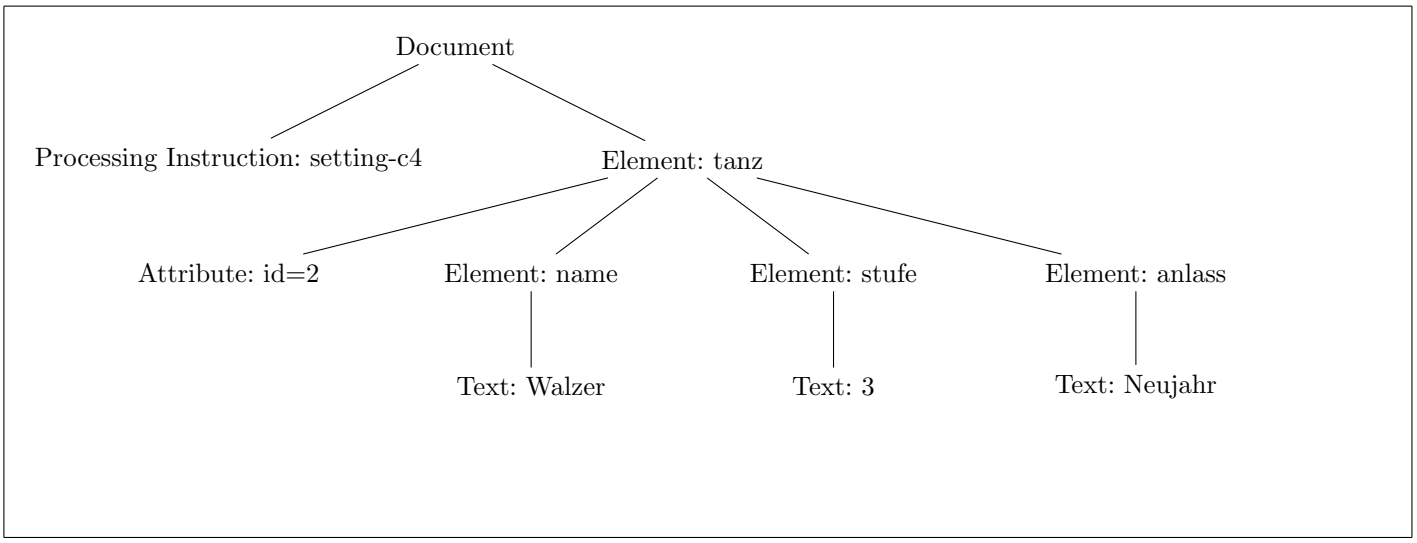
Geben Sie hier den Output von **query3.xsl** angewandt auf **dancing.xml** an:

```
Andy Kainz
Elke Winkens
```

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: Whatever").

```
<?setting-c4?><tanz id="2"><name>Walzer</name><stufe>3</stufe><anlass>Neujahr</anlass></tanz>
```



Aufgabe 6: (8)

Vervollständigen Sie die folgende Java Klasse sodass ein SAX Content-Handler herauskommt, der zählt wieviele Elemente und Attribute im Dokument vorkommen.

Die Anzahl der Elemente und Attribute soll am Ende der Verarbeitung auf die Konsole ausgegeben werden.

Das XML Dokument aus dem vorigen Beispiel sollte somit folgende Ausgabe produzieren:

```
Elemente: 4, Attribute: 1
```

```
class ElementCountHandler extends DefaultHandler {

  private int elemCount=0;
  private int attCount=0;

  public void startElement(String namespaceURI, String localName, String qName, Attributes atts)
  throws SAXException {
    elemCount++;
    attCount += atts.getLength();
  }

  public void endDocument() throws SAXException {
    System.out.println("Elemente: "+elemCount+", Attribute: "+attCount);
  }

}
```

Aufgabe 7: (8)

Schreiben Sie XPath-Anfragen um für XML-Dokumente wie **dancing.xml** (siehe Anhang) folgende Informationen zu selektieren:

1. Geben Sie die Anzahl der Paare, die eine ungerade Startnummer haben aus:

   ```
   count(//paar[@nr mod 2 = 1])
   ```

2. Finden Sie die Tanzpartnerin von Promi "Marc Pirchner":

   ```
   //partnerin[../promi = 'Marc Pirchner']
   ```

3. Geben Sie die Startnummern jener Paare aus, in denen mind. eine Person tanzt die "Alexander" im Namen hat (dh. das Element `promi`, `partner` oder `partnerin` enhalten den Text "Alexander"):

   ```
   //paar[contains(partner,'Alexander') or contains(partnerin,'Alexander') or contains(promi,'Alexander') ]/@nr
   ```

4. Geben Sie die Startnummern jener Paare aus, die *nicht* in Runde 7 ausgeschieden sind:

   ```
   //paar[not(@ergebnis='ausgeschieden in Runde 7')]/@nr
   ```

Aufgabe 8: (12)

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

1. SAX erlaubt es, ein XML Dokument in beliebiger Reihenfolge zu durchlaufen. wahr ◯ falsch ⊗
2. Wenn ein XML Dokument nicht wohlgeformt ist, muss der Parser die Verarbeitung abbrechen. wahr ⊗ falsch ◯
3. Wenn ein XML Dokument "gültig" ist, muss es nicht zwingend "wohlgeformt" sein. wahr ◯ falsch ⊗
4. Die DTD Elementdeklaration `(A?|B?|C?)+` ist gleichwertig mit `(A|B|C)*`. wahr ⊗ falsch ◯
5. XML ist eine Verallgemeinerung von SGML (XML ist "mächtiger" als SGML). wahr ◯ falsch ⊗
6. Folgende zwei Ausdrücke sind äquivalent:
   `//element[last()]`
   `//element[position()=last()]` wahr ⊗ falsch ◯
7. Folgende zwei Elemente sind äquivalent:
   `<el att1='a1' att2='a2'/>`
   `<el att2='a2' att1='a1'></el>` wahr ⊗ falsch ◯
8. `<der text>text</der text>` ist wohlgeformt. wahr ◯ falsch ⊗

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

Sie können diese Seite abtrennen!

Das folgende XML-Dokument **dancing.xml** gilt für Aufgaben 3,4 und 7:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE dancingstars SYSTEM "dancing.dtd">
<dancingstars>
   <paar nr="1" ergebnis="ausgeschieden in Runde 1">
      <partner>Alexander Kreissl</partner>
      <promi>Claudia Stoeckl</promi>
   </paar>
   <paar nr="2" ergebnis="ausgeschieden in Runde 3">
      <partnerin>Julia Polai</partnerin>
      <promi>Oliver Stamm</promi>
   </paar>
   <paar nr="3" ergebnis="Finale verloren">
      <promi>Elisabeth Engstler</promi>
      <partner>Alexander Zaglmaier</partner>
   </paar>
   <paar nr="4" ergebnis="ausgeschieden in Runde 5">
      <partnerin>Kelly Kainz </partnerin>
      <promi>Marc Pirchner</promi>
   </paar>
   <paar nr="5" ergebnis="ausgeschieden in Runde 4">
      <promi>Christine Reiler</promi>
      <partner>Manfred Zehender</partner>
   </paar>
   <paar nr="6" ergebnis="ausgeschieden in Runde 7">
      <partnerin>Alice Guschelbauer</partnerin>
      <promi>Waterloo</promi>
   </paar>
   <paar nr="7" ergebnis="Finale gewonnen">
      <partnerin>Nicole Kuntner</partnerin>
      <promi>Dorian Steidl</promi>
   </paar>
   <paar nr="8" ergebnis="ausgeschieden in Runde 6">
      <partner>Balazs Ekker</partner>
      <promi>Jeannine Schiller</promi>
   </paar>
   <paar nr="9" ergebnis="ausgeschieden in Runde 2">
      <partner>Michaela Heintzinger</partner>
      <promi>Peter Tichatschek</promi>
   </paar>
   <paar nr="10" ergebnis="ausgeschieden im Semifinale">
      <partner>Andy Kainz</partner>
      <promi>Elke Winkens</promi>
   </paar>
</dancingstars>
```

Gesamtpunkte: 75