

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			19. 10. 2007
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**:

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

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

Hinweise:

- Gehen Sie davon aus, dass allen folgenden Dateien die Zeile  
    <!DOCTYPE As 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. <As><A><B>abc</B></A></As> gültig  ungültig
2. <As><A><C attr="abc">def</C></A></As> gültig  ungültig
3. <As><A><B>abc</B><C attr="def">ghj</C><B>ikl</B></A></As> gültig  ungültig
4. <As><A><B>abc</B></A><A><B><D/>def</B></A></As> gültig  ungültig
5. <As><A><B>abc<E>def</E>ghj</B></A></As> gültig  ungültig
6. <As></As> 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 in der XML Schema Definition **seminar.xsd** die complexType-Definition des Typs "VortragTyp", so dass das XML-Dokument **seminar.xml** (siehe Anhang) bezüglich dieses Schemas gültig ist. Berücksichtigen Sie dabei folgende Punkte:

- Ein "zusammenfassung"-Element hat gemischten Inhalt. Als Subelement kann nur das Element "keyword" – jedoch beliebig oft – auftreten.
- Das "bereich"-Subelement des "vortrag"-Elements soll zumindest einmal auftreten.
- Das Attribut "lva" des "vortrag"-Elements soll den Default-Wert "Seminar" haben.

Datei **seminar.xsd**:

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

  <xs:element name="seminar">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="seminar:vortrag" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:element name="vortrag" type="seminar:VortragTyp"/>

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

  </xs:complexType>
</xs:schema>
```

### Aufgabe 3:

(9)

Vervollständigen Sie die XML Schema Definition **formel1.xsd**, so dass das XML-Dokument **formel1.xml** (siehe Anhang) bezüglich dieses Schemas gültig ist. Berücksichtigen Sie beim Vervollständigen der XML Schema Definition folgende Punkte:

- Sie brauchen sich **nicht um die Definition** der Typen “teamTyp”, “laufTyp” und “saisonTyp” zu **kümmern!**
- Ergänzen Sie die Element-Definition von “formel1”, so dass folgende Schlüssel definiert werden:
  - Das Subelement “startnr” ist ein Primärschlüssel für die Fahrer.
  - Das Attribut “nr” der “lauf”-Elemente ist ein Primärschlüssel für die Läufe.
  - Die “gewinner”-Elemente enthalten folgende zwei Fremdschlüssel: Das Subelement “fahrer” ist ein Fremdschlüssel auf den Primärschlüssel für die Fahrer; das Subelement “lauf” ist ein Fremdschlüssel auf den Primärschlüssel für die Läufe.

Datei **formel1.xsd**:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="formel1">
    <xsd:complexType>
      <xsd:sequence>
        <xsd:element name="team" maxOccurs="unbounded" type="teamTyp" />
        <xsd:element name="lauf" maxOccurs="unbounded" type="laufTyp"/>
        <xsd:element name="saison" type="saisonTyp"/>
      </xsd:sequence>
    </xsd:complexType>

    <!-- Fügen Sie hier die Primär- und Fremdschlüssel-Definitionen ein. -->

  </xsd:element>

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

  <xsd:complexType name="teamTyp"> ..... </xsd:complexType>
  <xsd:complexType name="laufTyp"> ..... </xsd:complexType>
  <xsd:complexType name="saisonTyp"> .... </xsd:complexType>
</xsd:schema>
```

**Aufgabe 4:**

Vervollständigen Sie das XSLT Stylesheet **formel1.xsl**, das auf Instanz-Dokumente des Schemas **formel1.xsd** der vorigen Aufgabe angewandt werden kann und ein XML-Dokument gemäß folgender DTD liefert:

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT erfolge (team*)>
<!ELEMENT team (bez, rennen*)>
<!ELEMENT bez (#PCDATA)>
<!ELEMENT rennen (#PCDATA)>
```

**Erläuterung zu dieser DTD:**

- Das “bez”-Element dieser DTD entspricht dem “bezeichnung”-Subelement des Elements “team” in **formel1.xml**.
- Das “rennen”-Element dieser DTD entspricht dem “rennen”-Subelement des Elements “lauf” in **formel1.xml**.

Das gesuchte Stylesheet soll für jedes Team die gewonnenen Rennen ausgeben. Es sollen jedoch *nicht* die Fahrer ausgegeben werden die für das jeweilige Team die Rennen gewonnen haben.

**Beispiel:** Angewandt auf das XML-Dokument **formel1.xml** liefert dieses Stylesheet daher folgendes Ergebnis (Anmerkung: die Einrückungen dienen nur zur besseren Lesbarkeit und müssen nicht berücksichtigt werden).

```
<?xml version="1.0" encoding="UTF-8"?>
<erfolge>
  <team>
    <bez>McLaren-Mercedes</bez>
    <rennen>GP von Malaysia</rennen>
  </team>
  <team>
    <bez>Ferrari</bez>
    <rennen>GP von Bahrain</rennen>
    <rennen>GP von Australien</rennen>
  </team>
</erfolge>
```

Datei **formel1.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="formel1">
  <erfolge>
    <xsl:apply-templates select="//team" />
  </erfolge>
</xsl:template>

<xsl:template match="team">
<!-- Vervollständigen Sie dieses Template -->

</xsl:template>

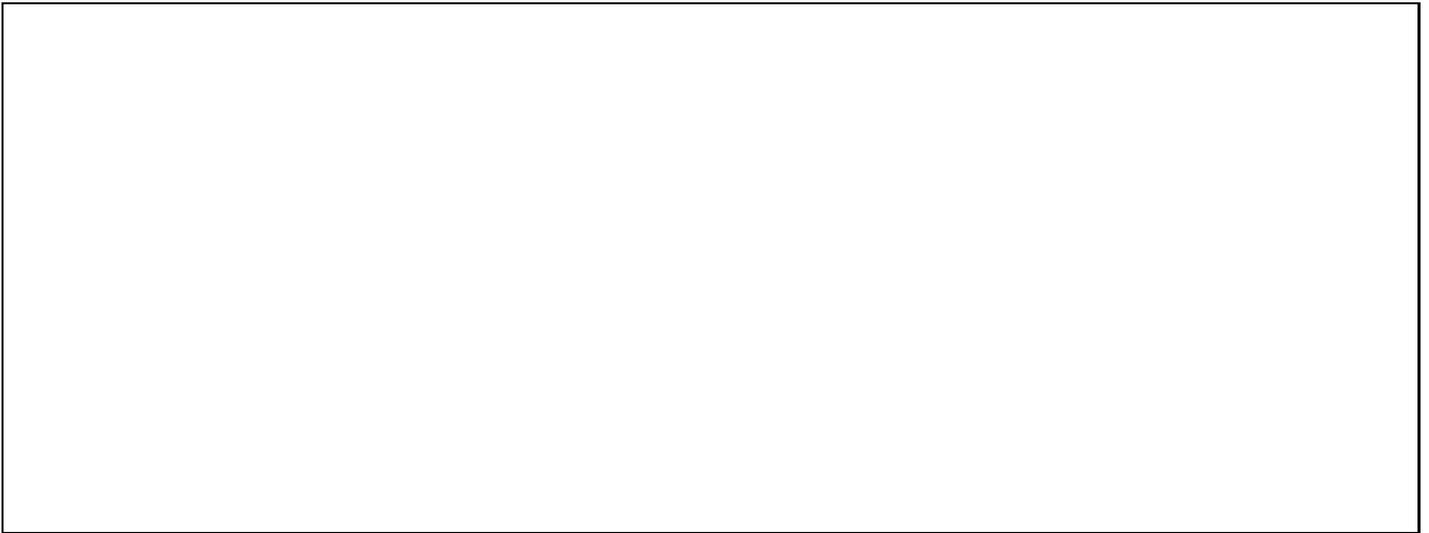
</xsl:stylesheet>
```

**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: Whatever").

```
<gebaeck><name>Sacher Torte</name><?secret?><zutaten><zutat>Ingwer</zutat><zutat>Anis</zutat></zutaten></gebaeck>
```

**Aufgabe 6:**

(8)

Vervollständigen Sie die folgende Java Klasse sodass ein SAX Content-Handler herauskommt, der alle Attribute (plus Inhalt) aller Elemente auf die Konsole ausgibt (zB "attributname: inhalt"). Sie brauchen sich nicht um eine Fehlerbehandlung kümmern; die Reihenfolge in der die Attribute ausgegeben werden ist nicht relevant.

```
class AttributePrinterHandler extends DefaultHandler {
```

```
}
```

**Aufgabe 7:**

(8)

(a) Schreiben Sie folgende XPath-Anfragen für das XML-Dokument **seminar.xml** von Aufgabe 2:

1. Alle Vorträge, in deren Zusammenfassung kein keyword vorkommt.
2. Alle keywords, die in der Zusammenfassung von Vorträgen des LVA-Typs 'Seminar' vorkommen.

(b) Schreiben Sie folgende XPath-Anfragen für das XML-Dokument **formel1.xml** der Aufgaben 3 und 4:

3. Anzahl der Teams, die zumindest einen Fahrer aus Spanien haben.
4. Alle Teams, bei denen sämtliche Fahrer eine Startnummer > 3 haben.

**Aufgabe 8:**

(12)

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

1. Eine DTD für die XML-Datei seminar.xml von Aufgabe 2 könnte mit folgender Element-Deklaration beginnen:  
`<!ELEMENT dbai:seminar (vortrag*)>`. wahr  falsch
2. In einer DTD für die XML-Datei seminar.xml von Aufgabe 2 müssten für das Element `dbai:seminar` folgende Attribute deklariert werden: `dbai`, `xsi` und `xsi:schemaLocation` wahr  falsch
3. Angenommen das XSLT-Stylesheet formel1.xsl von Aufgabe 4 würde folgendes Template enthalten:  
`<xsl:template match="text()"></xsl:template>`  
Dann ist es nicht mehr möglich, Textinhalte (z.B. den Inhalt von `bezeichnung`-Elementen) vom Quelldokument in das Zieldokument zu kopieren. wahr  falsch
4. In einem XSLT-Stylesheet ist es erlaubt, dass mittels `apply-templates` ein Knoten ausgewählt wird, auf den kein einziger `match`-Ausdruck eines `template`-Elements zutrifft. wahr  falsch
5. Für das XML-Dokument formel1.xml liefert folgende XPath-Anfrage den Boole'schen Wert `true`:  
`//team[bezeichnung='Ferrari']//startnr = //gewinner/fahrer` wahr  falsch
6. Für das XML-Dokument formel1.xml liefert folgende XPath-Anfrage den Boole'schen Wert `false`:  
`//team[bezeichnung='Ferrari']//startnr != //gewinner/fahrer` wahr  falsch
7. Ein DOM-Parser lädt normalerweise das gesamte Dokument in den Speicher. wahr  falsch
8. Attribute werden bei einem SAX Parser als eigene Events getriggert. 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 `seminar.xml` gilt für **Aufgabe 2**:

```
<?xml version="1.0" encoding="UTF-8"?>
<dbai:seminar xmlns:dbai="http://www.dbai.tuwien.ac.at/education"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.dbai.tuwien.ac.at/education/seminar.xsd">

  <dbai:vortrag typ="Seminar">
    <titel>XML-Schema</titel>
    <name>Grete Assistent</name>
    <datum>2007-03-21</datum>
    <zusammenfassung>
      Ueberblick ueber <keyword>Namespaces</keyword>, Deklarationen
      von Elementen und Attributen. Einfuehrung von simplen und
      komplexen Typen. Schliesslich werden Konzepte wie
      <keyword>Vererbung</keyword>, <keyword>Primaerschlusessel</keyword>,
      und <keyword>Fremdschlusessel</keyword> vorgestellt.
    </zusammenfassung>
    <bereich>Semistrukturierte Daten</bereich>
    <bereich>XML</bereich>
  </dbai:vortrag>

  <dbai:vortrag typ="Proseminar">
    <titel>Document Type Definitions</titel>
    <name>Hansi Student</name>
    <datum>2007-06-30</datum>
    <zusammenfassung>
      Vorgestellt werden Dokumenttyp-Deklaration, Element-Deklaration,
      Attribut-Deklaration und Entitaeten. Naeher eingegangen wird im
      speziellen auf regulaere Ausdruecke in Auftretensindikatoren.
    </zusammenfassung>
    <bereich>DTD</bereich>
  </dbai:vortrag>
</dbai:seminar>
```

Das folgende XML-Dokument **formel1.xml** gilt für die **Aufgaben 3 und 4**:

```
<?xml version="1.0" encoding="UTF-8"?>
<formel1>
  <team>
    <bezeichnung>McLaren-Mercedes</bezeichnung>
    <fahrer>
      <startnr>1</startnr>
      <name>Fernando Alonso</name>
      <land>Spanien</land>
    </fahrer>
    <fahrer>
      <startnr>2</startnr>
      <name>Lewis Hamilton</name>
      <land>Grossbritannien</land>
    </fahrer>
  </team>
  <team>
    <bezeichnung>Ferrari</bezeichnung>
    <fahrer>
      <startnr>5</startnr>
      <name>Felipe Massa</name>
      <land>Brasilien</land>
    </fahrer>
    <fahrer>
      <startnr>6</startnr>
      <name>Kimi Raikonen</name>
      <land>Finnland</land>
    </fahrer>
  </team>
  <lauf nr="7001">
    <rennen>GP von Australien</rennen>
    <ort>Melbourne</ort>
  </lauf>
  <lauf nr="7002">
    <rennen>GP von Malaysia </rennen>
    <ort>Sepang</ort>
  </lauf>
  <lauf nr="7003">
    <rennen>GP von Bahrain</rennen>
    <ort>Sakhir</ort>
  </lauf>
  <saison>
    <gewinner>
      <lauf>7001</lauf>
      <fahrer>6</fahrer>
      <datum>2007-03-18</datum>
    </gewinner>
    <gewinner>
      <lauf>7002</lauf>
      <fahrer>1</fahrer>
      <datum>2007-04-01</datum>
    </gewinner>
    <gewinner>
      <lauf>7003</lauf>
      <fahrer>5</fahrer>
      <datum>2007-04-15</datum>
    </gewinner>
  </saison>
</formel1>
```

Gesamtpunkte: 75