Gruppe A	PRÜFUNG AUS "SEMISTRUKTURIERTE DATEN" **184.705**		25. 06. 2014
Kennnr.	Matrikelnr.	Familienname	Vorname

Arbeitszeit: 100 Minuten. Aufgaben sind auf den Angabeblättern zu lösen; Zusatzblätter werden nicht gewertet.
Bitte tragen Sie Namen, Studienkennzahl und Matrikelnummer ein. Halten Sie Ihren Studentenausweis bereit.

Aufgabe 1: (12)

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

```
<?xml version="1.0"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

        <xsd:element name="a" type="atype"/>

        <xsd:complexType name="atype">
                <xsd:all>
                        <xsd:element name="b" type="btype" minOccurs="0" maxOccurs="1"/>
                        <xsd:element name="a" type="atype" minOccurs="0" maxOccurs="1"/>
                </xsd:all>
        </xsd:complexType>

        <xsd:complexType name="btype">
                <xsd:choice minOccurs="0" maxOccurs="1">
                        <xsd:element name="b" type="btype"/>
                        <xsd:element name="c" type="xsd:int"/>
                </xsd:choice>
        </xsd:complexType>
</xsd:schema>
```

Betrachten Sie weiters die acht 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/>` gültig ◯ ungültig ◯
2. `<a><b/><a/></a>` gültig ◯ ungültig ◯
3. `<a><b><a/></b></a>` gültig ◯ ungültig ◯
4. `<a><a/><a/></a>` gültig ◯ ungültig ◯
5. `<a><b><c>1</c></b></a>` gültig ◯ ungültig ◯
6. `<a><b><c>1</c><c>2</c></b></a>` gültig ◯ ungültig ◯
7. `<a><b><b><b/></b></b></a>` gültig ◯ ungültig ◯
8. `<b/>` 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: (15)

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

1. XML gibt es schon wesentlich länger als das WWW. wahr ◯ falsch ◯
2. Die XML-Recommendation lässt offen, wie sich XML-Parser bei Wohlgeformtheitsfehler verhalten sollen. wahr ◯ falsch ◯
3. Wohlgeformte XML-Dokumente dürfen im Allgemeinen mehr als ein Wurzelelement haben. wahr ◯ falsch ◯
4. Der Speicherbedarf eines SAX-Parsers ist im Allgemeinen geringer als eines DOM-Parsers. wahr ◯ falsch ◯
5. Das Resultat eines XPATH Ausdrucks ist immer in Dokument-Order. wahr ◯ falsch ◯
6. Bei SAX können auch mehrere `characters` events hintereinander auftreten. wahr ◯ falsch ◯
7. Rekursive Definitionen wie z.B. `<!ELEMENT B (A,B?,C)>` sind in DTDs verboten. wahr ◯ falsch ◯
8. Der XPath-Ausdruck `//name` ist die Kurzschreibweise des XPath-Ausdrucks `/descendant::name` wahr ◯ falsch ◯
9. Syntaktisch ist JSON ein Teil von JavaScript. wahr ◯ falsch ◯
10. Das Konzept der "CDATA Sections" spielt in JSON eine wichtige Rolle. 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)

Die folgenden Aufgaben 3 – 7 beziehen sich auf das XML-Dokument gruppen.xml, das Sie auf der letzten Seite dieser Prüfungsangabe finden.

Aufgabe 3: (12)

Vervollständigen Sie das DTD Dokument **gruppe.dtd**, sodass XML-Dokumente in der Gestalt von **gruppe.xml** (siehe Anhang) bezüglich dieser DTD gültig sind. Berücksichtigen Sie beim Erstellen der DTD folgende Punkte:

- Das Element `gruppen` ist das Wurzelelement und besteht aus mindestens einem `gruppe`-Element. Jedes `gruppe`-Element besitzt ein Attribut `bezeichnung` und hat folgende Unterstruktur:
- Exakt vier `team`-Elemente, gefolgt von beliebig vielen `spiel`-Elementen. Jedes `team`-Element besitzt ein Attribut `name` sowie ein Attribut `code`. Jedes `spiel`-Element besitzt Attribute `a` `b` und `erg` und kann Kindelemente `ereignis` und `spieler` haben.
- Alle Attribute sollen verpflichtend sein. Spezifizieren Sie die Attribute (wählen Sie sinnvolle Typen aus!) sowie die nicht näher erläuterten Häufigkeiten entsprechend dem **gruppe.xml** Dokument im Anhang.
- Versuchen Sie weiters entsprechende Schlüsselbeziehungen zu finden und in der DTD abzubilden. Beachten Sie ebenso anhand von **gruppe.xml**, dass Elemente gemischten Inhalt haben können.

Datei **gruppe.dtd**:

Aufgabe 4: (10)

Betrachten Sie die folgenden XPath-Abfragen angewandt auf das Dokument **gruppe.xml** (siehe Anhang).

- Falls als Ergebnis `team` Elemente selektiert werden, geben Sie jeweils das Attribut `code` an.
- Falls als Ergebnis `spiel` Elemente selektiert werden, geben Sie jeweils das Attribut `a` an.
- Falls als Ergebnis `gruppe` Elemente selektiert werden, geben Sie jeweils das Attribut `bezeichnung` an.
- Falls als Ergebnis mehrere Elemente selektiert werden, trennen Sie die jeweiligen Ausgaben durch Leerzeichen.
- Falls der angegebene XPath Ausdruck keine Knoten selektiert, notieren Sie im entsprechenden Feld "leere Ausgabe".

Betrachten Sie dazu folgendes Beispiel:

```
//team
```

```
BRA CRO MEX KAM ESP NED CHI AUS
```

Geben Sie nun die entsprechende Ausgaben der folgenden XPath-Abfragen an.

```
count(//team)
```

```
//spiel[*]
```

```
//team[2]
```

```
(//team)[2]
```

```
//gruppe[spiel]
```

Aufgabe 5: (8)

Betrachten Sie folgende-XQuery Abfrage **gruppe.xq** angewandt auf **gruppe.xml**:

```
for $t in //team[@code=//spiel/@a or @code=//spiel/@b]

let $b := //spiel[@a=$t/@code or @b=$t/@code]

return element {$t/@name} {
   element erg {string($b/@erg)}
}
```

Geben Sie zuerst die `code` Attribute aller Knoten an, die `$t` durchläuft:

Geben Sie nun die `a` Attribute aller Knoten an, die in `$b` gespeichert werden (falls Werte mehrfach vorkommen, geben Sie diese mehrfach an).

Geben Sie nun die Ausgabe von **gruppe.xq** angewandt auf **gruppe.xml** an.
Die exakte Behandlung von Whitespaces ist dabei nicht relevant.

Aufgabe 6: (9)

Erstellen Sie ein XSLT-Stylesheet **gruppe.xsl**, das angewandt auf Dokumente der Gestalt **gruppe.xml** folgende Ausgabe erzeugt:

- Als Dokumentelement soll das Element `berichte` erzeugt werden.
- Für jedes `spiel` Element wird ein Element `bericht` mit folgendem Inhalt erzeugt:
  - dem Name des Teams, auf das sich das `a` Attribut bezieht (nicht der Code!)
  - dem Name des Teams, auf das sich das `b` Attribut bezieht
  - dem Inhalt des `erg` Attributs

Für das Dokument **gruppe.xml** soll beispielsweise folgende Ausgabe erzeugt werden:

```
<berichte>
   <bericht>Brasilien - Kroation: 3:1</bericht>
   <bericht>Mexiko - Kamerun: 1:0</bericht>
</berichte>
```

Vervollständigen Sie hier das XSLT-Stylesheet **gruppe.xsl**. Die Verwendung von Kontrollstrukturen wie `xsl:for-each`, `xsl:if`, etc. ist für die Lösung *nicht* erlaubt (und auch nicht sinnvoll)! Sie brauchen sich nicht um Whitespaces etc. zu kümmern.

Datei **gruppe.xsl**:

```
<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="                          ">









  </xsl:template>

  <xsl:template match="                          ">

















  </xsl:template>

</xsl:stylesheet>
```

Aufgabe 7: (9)

Vervollständigen Sie den folgenden SAX Handler, der angewandt auf Dokumente der Gestalt **gruppe.xml** für jede Gruppe folgende Informationen auf `System.out` ausgibt:

- den Namen der Gruppe
- die Anzahl der bisherigen Spiele in dieser Gruppe (`spiel` Elemente)

Für das Dokument **gruppe.xml** wird beispielsweise folgende Ausgabe erwartet:

```
Gruppe A: 2 Spiele
Gruppe B: 0 Spiele
```

Um die genaue Formatierung brauchen Sie sich nicht zu kümmern.

Datei **Gruppe.java**:

```
public class Gruppe extends DefaultHandler {

  // Hier wird vermutlich eine Variable benötigt




  public void startElement(String uri, String localName, String qName, Attributes atts)
    throws SAXException {

  }

  public void endElement(String uri, String localName, String qName) throws SAXException {

  }
}
```

Sie können diese Seite abtrennen!

Datei **gruppe.xml**:

```
<gruppen>
   <gruppe bezeichnung="A">
      <team name="Brasilien" code="BRA"/>
      <team name="Kroation" code="CRO"/>
      <team name="Mexiko" code="MEX"/>
      <team name="Kamerun" code="KAM"/>
      <spiel a="BRA" b="CRO" erg="3:1">Im <ereignis>Eroeffnungsspiel</ereignis> der WM 2014
                zwischen Brasilien und Kroatien schoss <spieler>Neymar</spieler>
                das entscheidende Tor.</spiel>
      <spiel a="MEX" b="KAM" erg="1:0"/>
   </gruppe>
   <gruppe bezeichnung="B">
      <team name="Spanien" code="ESP"/>
      <team name="Niederlande" code="NED"/>
      <team name="Chile" code="CHI"/>
      <team name="Australien" code="AUS"/>
   </gruppe>
</gruppen>
```

Gesamtpunkte: 75