

Klausur zum Grundkurs Markup-Sprachen (WS 2014/2015, LV-Nr. 36663a)

Montag, 2. Februar 2015 | Dozent: Florian Meier, M.A.

Beginn: 14 Uhr, c.t. Raum: H2

Vorname:	_____
Nachname:	_____
Matrikelnummer:	_____
E-Mail:	_____
Studienfächer:	_____

Allgemeine Hinweise

1. Überprüfen Sie bitte, ob Sie alle Seiten (6) der Klausurangabe vollständig erhalten haben.
2. Bearbeitungszeit: **60 Minuten**.
3. Maximal erreichbare **Punktzahl: 56**. Zu Ihrer Orientierung sind die erreichbaren Punkte bei jeder Frage genannt – bitte teilen Sie die Arbeitszeit entsprechend ein.
4. Vergessen Sie nicht Ihren Namen, Ihre Matrikelnummer, Ihre E-Mail-Adresse sowie Ihre Studienfächer leserlich oben auf dieser Seite einzutragen bevor Sie mit der Bearbeitung beginnen.
5. Verwenden Sie nur den Klausurbogen für Ihre Lösungen. Sollte der Platz nicht ausreichen, so verwenden Sie die Rückseite des Bogens und geben Sie an, auf welche Frage sich die Lösung bezieht.
6. Benutzen Sie keine Bleistifte, keine rotschreibenden Stifte und kein TippEx (oder ähnliche Produkte).
7. Es sind keine weiteren Unterlagen (Skripte, Vorlesungsmitschriften, etc.) zugelassen.
8. Wenden Sie sich bei Unklarheiten in den Aufgabenstellungen immer an die Aufsichtsführenden. Hinweise und Hilfestellungen werden dann, falls erforderlich, offiziell für den gesamten Hörsaal durchgegeben.
9. Geben Sie keine mehrdeutigen (oder mehrere) Lösungen an. In solchen Fällen wird stets die Lösung mit der geringeren Punktzahl gewertet. Eine richtige und eine falsche Lösung zu einer Aufgabe ergeben also null Punkte.
10. Verändern Sie die Aufgabenstellung nicht, um Sie an Ihre Lösung „anzupassen“. Lösungen, die sich nicht an die vorgegebenen Aufgabenstellungen halten, werden mit null Punkten bewertet.

Viel Erfolg!

- 1. Motivation und Kontext** | Nennen Sie Gründe, die für die Verwendung von Markup-Sprachen bzw. Formaten wie XML sprechen! (4 Punkte)

- 2. Markup-Grundlagen** | Erklären Sie anhand der beiden nachfolgenden HTML5-Codebeispiele den Unterschied zwischen *generic coding* und *specific coding*! (4 Punkte)

```
Bsp-01: <p style="color:blue;">Lorem ipsum</p>  
Bsp-02: <footer>Lorem ipsum</footer>
```

- 3. Markup-Sprachen** | Erläutern Sie kurz den Zusammenhang zwischen SGML, XML, HTML , XHTML und HTML5! (5 Punkte)

- 4. Validität- und Qualitätssicherung** | Nennen und diskutieren Sie unterschiedliche Varianten, um die Validität von XML-Dokumenten sicherzustellen. Gehen Sie in der Diskussion auf mögliche Vor- und Nachteile dieser Varianten ein! (8 Punkte)

- 5. XML-Namensräume** | Erklären Sie anhand eines Beispiels (z.B. Briefe von Beethoven in XML) den Verwendungszweck von Namensräumen! (4 Punkte)

- 6. XSLT** | Bei der Verarbeitung eines XML-Dokumentes mit Hilfe eines XSLT-Prozessors unterscheidet man zwei *processing*-Ansätze. Welche Ansätze sind dies und worin besteht deren Unterschied? Geben Sie zudem pro *processing*-Ansatz jeweils ein XSLT-Element an, das diesen Ansatz vertritt! (6 Punkte)

7. XML-Schema | An welchen Stellen verstößt das nachfolgende XML-Dokument gegen das eingebundene XML-Schema? Begründen Sie jeden Verstoß kurz! (6 Punkte)

XML-Dokument: shiporder.xml

Verstöße:

```

1  <?xml version="1.0" encoding="UTF-8" ?>
2  <shiporder
3    xmlns="http://www.shiporder.de/schema"
4    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5    xsi:schemaLocation="http://www.shiporder.de/schema shiporder.xsd"
6    Version="1.0">
7    <order orderid="889923">
8      <orderperson>John Smith</orderperson>
9      <shipto kind="present wrapping">
10         <name>Ola Nordmann</name>
11         <address>Langgt 23</address>
12         <city>4000 Stavanger</city>
13         <country>Norway</country>
14       </shipto>
15       <item>
16         <title>Hide your heart</title>
17         <price>9.90</price>
18         <quantity>1</quantity>
19       </item>
20     </order>
21     <order orderid="88954">
22       <shipto>
23         <name>Mary Doe</name>
24         <address>Kellerweg 8</address>
25         <city>93053 Regensburg</city>
26         <country>Germany</country>
27       </shipto>
28       <item>
29         <title>niveau weshalb warum</title>
30         <quantity>1</quantity>
31         <preis>10.99</preis>
32       </item>
33     </order>
34 </shiporder>

```

1)

2)

3)

XML-Schema: shiporder.xsd

```

1  <?xml version="1.0" encoding="UTF-8" ?>
2  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
3    xmlns="http://www.shiporder.de/schema"
4    targetNamespace="http://www.shiporder.de/schema"
5    elementFormDefault="qualified">
6    <xs:element name="shiporder">
7      <xs:complexType>
8        <xs:sequence>
9          <xs:element name="order" minOccurs="1" maxOccurs="unbounded">
10             <xs:complexType>
11               <xs:sequence>
12                 <xs:element name="orderperson" type="xs:string"/>
13                 <xs:element name="shipto" type="shiptoType"/>
14                 <xs:element name="item" type="itemType"
15                   minOccurs="1" maxOccurs="unbounded"/>
16               </xs:sequence>
17               <xs:attribute name="orderid" type="orderidType" use="required"/>
18             </xs:complexType>
19           </xs:element>
20         </xs:sequence>
21       </xs:complexType>
22     </xs:element>
23     <xs:complexType name="shiptoType">
24       <xs:sequence>
25         <xs:element name="name" type="xs:string"/>
26         <xs:element name="address" type="xs:string"/>
27         <xs:element name="city" type="xs:string"/>
28         <xs:element name="country" type="xs:string"/>
29       </xs:sequence>
30       <xs:attribute name="kind" type="kindType" use="optional"/>
31     </xs:complexType>
32     <xs:complexType name="itemType">
33       <xs:sequence>
34         <xs:element name="title" type="xs:string"/>
35         <xs:element name="note" type="xs:string" minOccurs="0"/>
36         <xs:element name="quantity" type="xs:positiveInteger"/>
37         <xs:element name="price" type="xs:decimal"/>
38       </xs:sequence>
39     </xs:complexType>
40     <xs:simpleType name="orderidType">
41       <xs:restriction base="xs:integer">
42         <xs:pattern value="[0-9]{6}"/>
43       </xs:restriction>
44     </xs:simpleType>
45     <xs:simpleType name="kindType">
46       <xs:restriction base="xs:string">
47         <xs:enumeration value="present"/>
48         <xs:enumeration value="single shipments"/>
49       </xs:restriction>
50     </xs:simpleType>
51 </xs:schema>

```

4)

5)

6)

8. XPath | Formulieren Sie XPath-Ausdrücke für folgende Bedingungen! (4 Punkte).

employeesByDepartment.xml

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <employeesByDepartment>
3    <department name="Accounts">
4      <employee jobTitle="Auditor">
5        <lastname>Bloom</lastname><firstname>Leo</firstname>
6      </employee>
7      <employee jobTitle="Bookkeeper">
8        <lastname>Cratchit</lastname><firstname>Bob</firstname>
9      </employee>
10     <employee jobTitle="Accountant">
11       <lastname>Pacioli</lastname><firstname>Luca</firstname>
12     </employee>
13   </department>
14   <department name="IT">
15     <employee jobTitle="Developer">
16       <lastname>Ayers</lastname><firstname>Danny</firstname>
17     </employee>
18     <employee jobTitle="Head of Infrastructure">
19       <lastname>Babbage</lastname><firstname>Charles</firstname>
20     </employee>
21     <employee jobTitle="Developer">
22       <lastname>Fawcett</lastname><firstname>Joe</firstname>
23     </employee>
24     <employee jobTitle="Developer">
25       <lastname>Hopper</lastname><firstname>Grace</firstname>
26     </employee>
27   </department>
28   <department name="Management">
29     <employee jobTitle="CEO">
30       <lastname>Bialystock</lastname><firstname>Max</firstname>
31     </employee>
32     <employee jobTitle="VP of Products">
33       <lastname>De Bris</lastname><firstname>Bob</firstname>
34     </employee>
35     <employee jobTitle="PA to the VP of Products">
36       <lastname>Ghia</lastname><firstname>Carmen</firstname>
37     </employee>
38   </department>
39 </employeesByDepartment>
    
```

a) Lassen Sie sich die Vornamen (firstname) aller Mitarbeiter (employee) ausgeben, die in der IT Abteilung (department) als Entwickler (developer) arbeiten! (nur Text/Inhalt)

b) Ermitteln Sie, welche Berufe (jobTitle) es in der Abteilung (department) "Buchhaltung"(Accounts) gibt! (nur Text/Inhalt)

9. XPath | Welche Knoten bzw. Inhalte selektieren folgende XPath-Ausdrücke? (4 Punkte)

a) //department[@name="IT"]/employee[position()=1]/firstname/text()

b) //department[last()]/employee[2]/*

10. DTD und XML-Schema | Bei der Übersetzung folgender DTD in eine XML-Schema Variante sind Fehler entstanden. Ermitteln Sie die Stellen in der Schemadefinition, die nicht den Bedingungen der DTD entsprechen! Begründen Sie ihre Auswahl kurz! (6 Punkte)

playlist.xsd

playlist.dtd

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
3   xmlns="http://www.playlist.de/schema"
4   targetNamespace="http://www.playlist.de/schema"
5   elementFormDefault="qualified">
6   <xs:element name="playlist">
7     <xs:complexType>
8       <xs:sequence>
9         <xs:element name="playlistitem" minOccurs="1" maxOccurs="5">
10          <xs:complexType>
11            <xs:all>
12              <xs:element name="artists" type="xs:string"/>
13              <xs:element name="song" type="xs:string" minOccurs="1"
14                maxOccurs="unbounded" />
15              <xs:element name="album" type="albumType"
16                minOccurs="0" maxOccurs="unbounded"/>
17            </xs:all>
18            <xs:attribute name="version" type="xs:string" use="required"/>
19          </xs:complexType>
20        </xs:element>
21      </xs:sequence>
22    </xs:complexType>
23  </xs:element>
24  <xs:complexType name="albumType">
25    <xs:simpleContent>
26      <xs:extension base="xs:string">
27        <xs:attribute name="source"
28          type="sourceType" use="optional" />
29      </xs:extension>
30    </xs:simpleContent>
31  </xs:complexType>
32  <xs:simpleType name="sourceType">
33    <xs:restriction base="xs:string">
34      <xs:enumeration value="vinyl"/>
35      <xs:enumeration value="cd"/>
36      <xs:enumeration value="tape"/>
37      <xs:enumeration value="dvd"/>
38      <xs:enumeration value="itunes" />
39    </xs:restriction>
40  </xs:simpleType>
41 </xs:schema>

```

```

1 <!DOCTYPE playlist [
2
3 <!ELEMENT playlist (playlistitem+)>
4 <!ELEMENT playlistitem (artist,song+,album?)>
5 <!ELEMENT artist (#PCDATA)>
6 <!ELEMENT song (#PCDATA)>
7 <!ELEMENT album (#PCDATA)>
8
9 <!ATTLIST playlistitem version CDATA #REQUIRED>
10 <!ATTLIST album source (vinyl|cd|tape|dvd) "tape">
11 ]>

```

- 1)
- 2)
- 3)
- 4)
- 5)
- 6)

11. FLWOR | Wofür steht das Akronym FLWOR? In welchem Kontext kommt FLWOR zum Einsatz? Was ist ein Äquivalent zu FLWOR im RDBM-Kontext? (5 Punkte)