

PRÜFUNG IN "SEMI-STRUKTURIERTE DATEN" 184.705			27. 06. 2017
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 folgende XML Schema Datei **test.xsd**:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="A">
    <xsd:complexType mixed="false">
      <xsd:sequence>
        <xsd:element name="A" type="xsd:boolean" maxOccurs="2"/>
        <xsd:element name="B" type="B" maxOccurs="2"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:element name="B">
    <xsd:complexType mixed="false">
      <xsd:sequence>
        <xsd:element name="A" type="B" minOccurs="0"/>
      </xsd:sequence>
    </xsd:complexType>
  </xsd:element>

  <xsd:complexType name="B" mixed="true">
    <xsd:choice>
      <xsd:element name="size" type="xsd:integer"/>
    </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><A>1<size>-42</size> valid invalid
- 2. valid invalid
- 3. <A><A>1<size>-42</size><A>1 valid invalid
- 4. <size>-42</size> valid invalid
- 5. <A><size>large</size> valid invalid
- 6. <A><A>1<A>1<A>1<size>-42</size> valid invalid
- 7. <A><A>1<A>1 valid invalid
- 8. <A><size>1</size><A><size>1</size> valid invalid

(Für jede korrekte Antwort 1.5 Punkte, für jede falsche Antwort -1.5 Punkte, unbeantwortete Fragen 0 Punkt, Insgesamt nicht weniger als 0 Punkte)

Aufgabe 2:

(15)

Entscheiden Sie, ob die folgenden Aussagen wahr oder falsch sind.

1. Das übliche Datenmodell für semi-strukturierte Daten sind vollständige Binärbäume. wahr falsch
2. XML ist eine Einschränkung von HTML. wahr falsch
3. Sowohl XML Schemas als auch DTDs sind in XML Syntax geschrieben. wahr falsch
4. XPath ist Teil des XQuery W3C Standards. wahr falsch
5. XPath ist Teil des XSLT W3C Standards. wahr falsch
6. XML wird häufig als Netzwerk Protokol verwendet. wahr falsch
7. Nicht jedes wohlgeformte XML Dokument ist auch valide. wahr falsch
8. DOM ist eine Baum-basierte API um XML Dokumente zu manipulieren. wahr falsch
9. Namespace URIs sind immer gültige XML Namen. wahr falsch
10. Attribute ohne Prefix sind im Default Namespace. wahr falsch

(Für jede korrekte Antwort 1.5 Punkte, **für jede falsche Antwort -1.5 Punkte**, unbeantwortete Fragen 0 Punkt, Insgesamt nicht weniger als 0 Punkte)

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

Aufgabe 3:

(12)

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

- Ein `game` Element enthält genau ein `name` Element, beliebig viele `developer` Elemente, genau ein `year` Element, zumindest ein `platform` Element, beliebig viele `genre` Elemente, und kann ein `dlcs` Element enthalten (in dieser Reihenfolge).
- Jedes `game` Element hat ein Attribut `key` das einen eindeutig Wert hat und kann ein `rating` Attribut haben, das eine Zahl zwischen 0 und 10 speichert.
- Ein `dlcs` Element enthält ein oder mehrere `dlc` Elemente.
- Die `series` Elemente enthalten genau ein `name` Element, gefolgt von mindestens einem `genre` Element, und genau einem `games` Element, welches selbst ein oder mehrere `reference` Elemente enthält.
- Alle `reference` Elemente haben ein Attribut `ref`, das auf den `key` eines `game` Elements verweist.
- Wenn nicht angegeben treffen Sie plausible Annahmen über die Typen von Elementen.

File `games.dtd`:

```
<!ELEMENT gamescollection (game | series)*>
<!ELEMENT game (name,developer*,year,platform+,genre*,dlcs?)>
<!ATTLIST game key ID #REQUIRED>
<!ATTLIST game rating (0|1|2|3|4|5|6|7|8|9|10) #IMPLIED>
<!ELEMENT name (#PCDATA)>
<!ELEMENT developer (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT platform (#PCDATA)>
<!ELEMENT genre (#PCDATA)>
<!ELEMENT dlcs (dlc+)>
<!ELEMENT dlc (#PCDATA)>

<!ELEMENT series (name,genre+,games)>
<!ELEMENT games (reference+)>
<!ELEMENT reference EMPTY>
<!ATTLIST reference ref IDREF #REQUIRED>
```

Aufgabe 4:

(10)

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

- Falls der angegebene XPath Ausdruck keine Knoten selektiert, notieren Sie im entsprechenden Feld "leere Ausgabe".
- Falls als Ergebnis eine Zahl selektiert wird (count, sum, ...), geben Sie diese Zahl an.

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

```
sum(//@rating)
```

32

```
//game[@key=//series//reference/@ref] [last()]/name/text()
```

Die Schicksalsklinge

```
//game[@rating] [last()]/platform
```

```
<platform>DOS</platform>  
<platform>Amiga</platform>
```

```
//game[last()][@rating]/platform
```

empty output

```
//game[not(@key=//@ref)]/year
```

```
<year>1994</year>  
<year>2013</year>
```

Aufgabe 5:

(6)

Betrachten Sie folgende-XQuery Abfrage **xquery.xq**:

```
<developers>
{
  for $d in distinct-values(//developer)
  for $g in //game[developer = $d]
  where $g/year > 1993 and $g/year <= 2013
  order by $d ascending, $g/year descending
  return
    <dev name="{ $d }">{ $g/name/text() }</dev>
}
</developers>
```

Geben Sie nun die Ausgabe von **xquery.xq** angewandt auf **gamescollection.xml** an.

Sie müssen sich nicht um Whitespaces kümmern.

```
<developers>
  <dev name="MG">UFO: Enemy Unknown</dev>
  <dev name="MP">UFO: Enemy Unknown</dev>
  <dev name="ND">The Last of Us</dev>
  <dev name="ND">Uncharted: Drake's Fortune</dev>
</developers>
```

Aufgabe 6:

(10)

Erstellen Sie ein XSLT-Stylesheet **xslt.xml**, das angewandt auf Dokumente der Gestalt **gamescollection.xml** folgende Ausgabe liefert:

- Ausgegeben werden soll ein HTML Dokument.
- Es sollen alle **PS4** Spiele eines Dokuments der Gestalt **gamescollection.xml** ausgegeben werden.
- Falls ein Spiel ein Rating größer als **7** besitzt soll der Text: “**Rating:** Excellent Game!” ausgegeben werden.
- Geben Sie zusätzlich jede Serie aus, die dieses Spiel referenziert.

Betrachten Sie dazu folgende Ausgabe, die ihr XSLT-Stylesheet **xslt.xml** angewandt auf **gamescollection.xml** (siehe Anhang) produzieren soll:

```
<html><head><title>My PS4 Games</title></head>
<body>
<h1>My PS4 Games</h1>
  <h2>Uncharted: Drake's Fortune</h2>
    <b>Serie:</b>Uncharted<br />
    <b>Serie:</b>Drake's Anthology<br />
  <h2>Uncharted 4: A Thief's End</h2>
    <b>Rating:</b> Excellent Game!<br />
    <b>Serie:</b>Uncharted<br />
</body>
</html>
```

Vervollständigen Sie hier das XSLT-Stylesheet **textref.xml**. Sie brauchen sich nicht um Whitespaces etc. zu kümmern.

File **xslt.xml**:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html"/>

  <xsl:template match="gamescollection">
    <html><head><title>My PS4 Games</title></head>
    <body>
    <h1>My PS4 Games</h1>
    <xsl:apply-templates select="game[platform/text() = 'PS4']" />
    </body>
  </html>
  </xsl:template>

  <xsl:template match="game">
    <xsl:variable name="gkey" select="@key" />
    <h2><xsl:value-of select="name" /></h2>
    <xsl:if test="@rating > 7"><b>Rating:</b> Excellent Game!<br /></xsl:if>
    <xsl:apply-templates select="//series[games/reference/@ref = $gkey]" />
  </xsl:template>

  <xsl:template match="series">
    <b>Serie:</b> <xsl:value-of select="name" /><br />
  </xsl:template>
</xsl:stylesheet>
```

Aufgabe 7:

(10)

Vervollständigen Sie die Methode `insDLC`, die zwei Parameter `key` und `dlcName` besitzt und mittels **DOM** folgende Änderungen am, in der Variable `doc` gespeicherten, Dokument durchführt:

- Im Spiel mit dem Key `key` soll der DLC `dlcName` eingefügt werden.
- Falls das Element `dlcs` im Spiel noch nicht existiert, soll dieses auch erzeugt werden.

Sie müssen sich nicht um die Fehlerbehandlung kümmern. Sie können annehmen, dass das Spiel mit dem Key `key` im Dokument existiert.

```
private static XPath xPath = XPathFactory.newInstance().newXPath();
Document doc;

public void insDLC (String key, String dlcName) throws Exception {
    XPathExpression xpe = xPath.compile("//game[@key = '" + key + "']");

    NodeList list = (NodeList) xpe.evaluate(doc, XPathConstants.NODESET);

    Node game = list.item(0);

    xpe = xPath.compile("dlcs");
    Node dlcs = (Node)xpe.evaluate(game, XPathConstants.NODE);
    if (dlcs == null) {
        dlcs = doc.createElement("dlcs");
        game.appendChild(dlcs);
    }

    Node dlc = doc.createElement("dlc");
    dlc.setTextContent(dlcName);

    dlcs.appendChild(dlc);
}
```

Total points: 75

Sie können diese Seite abtrennen!

File gamescollection.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<gamescollection>
  <game rating="7" key="Uncharted1">
    <name>Uncharted: Drake's Fortune</name>
    <developer>ND</developer>
    <year>2007</year>
    <platform>PS3</platform>
    <platform>PS4</platform>
  </game>
  <game rating="8" key="Uncharted4">
    <name>Uncharted 4: A Thief's End</name>
    <developer>ND</developer>
    <year>2016</year>
    <platform>PS4</platform>
  </game>
  <series>
    <name>Uncharted</name>
    <genre>Action-adventure</genre>
    <games>
      <reference ref="Uncharted1"/>
      <reference ref="Uncharted4"/>
    </games>
  </series>
  <game rating="9" key="XCOM">
    <name>UFO: Enemy Unknown</name>
    <developer>MP</developer>
    <developer>MG</developer>
    <year>1994</year>
    <platform>DOS</platform>
    <platform>PS1</platform>
  </game>
  <game rating="8" key="DSA1">
    <name>Die Schicksalsklinge</name>
    <developer>Attic </developer>
    <year>1992</year>
    <platform>DOS</platform>
    <platform>Amiga</platform>
  </game>
  <game key="LastofUs">
    <name>The Last of Us</name>
    <developer>ND</developer>
    <year>2013</year>
    <platform>PS3</platform>
    <genre>survival horror</genre>
    <genre>Action-adventure</genre>
    <dlcs><dlc>The Last of Us: Left Behind</dlc></dlcs>
  </game>
  <series>
    <name>Die Nordland-Trilogie</name>
    <genre>RPG</genre>
    <games><reference ref="DSA1"/></games>
  </series>
  <series>
    <name>Drake's Anthology</name>
    <genre>Action-adventure</genre>
    <games><reference ref="Uncharted1"/></games>
  </series>
</gamescollection>
```