EXAM IN "SEMI-STRUCTURED DATA" **184.705**			27. 06. 2017
Study Code	Student Id	Family Name	First Name

Working time: 100 minutes.
Exercises have to be solved on this exam sheet; Additional slips of paper will not be graded.
First, please fill in your name, study code and student number. Please, prepare your student id.

Exercise 1: (12)

Consider the following XML schema file **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>
```

Furthermore, consider the eight different XML files, which are listed below.

You may assume that each of the following XML files is well-formed. The point is to determine the validity according to **test.xsd**.

Check which of the following XML files are valid according to **test.xsd**.

1. `<A><A>1</A><B><size>-42</size></B></A>` valid ◯ invalid ◯
2. `<B/>` valid ◯ invalid ◯
3. `<A><A>1</A><B><size>-42</size></B><A>1</A></A>` valid ◯ invalid ◯
4. `<B><B><size>-42</size></B></B>` valid ◯ invalid ◯
5. `<B><A><size>large</size></A></B>` valid ◯ invalid ◯
6. `<A><A>1</A><A>1</A><A>1</A><B><size>-42</size></B></A>` valid ◯ invalid ◯
7. `<A><A>1</A><A>1</A></A>` valid ◯ invalid ◯
8. `<B><A><size>1</size></A><A><size>1</size></A></B>` valid ◯ invalid ◯

(For every correct answer 1.5 points, **for every incorrect answer -1.5 points**, for every unanswered question 0 points, you can have at least 0 points on this exercise)

Exercise 2: (15)

Decide which of the following statements are true or false.

1. The usual data model for semi-structured data are complete binary trees.	true ◯	false ◯
2. XML is a restriction of HTML.	true ◯	false ◯
3. Both XML Schemas and DTDs are written in XML syntax.	true ◯	false ◯
4. XPath is part of the XQuery W3C standard.	true ◯	false ◯
5. XPath is part of the XSLT W3C standard.	true ◯	false ◯
6. XML is often used as Network protocol.	true ◯	false ◯
7. Not all well-formed XML documents are valid.	true ◯	false ◯
8. DOM is a tree-based API for manipulating XML documents.	true ◯	false ◯
9. Namespace URIs are always valid XML names.	true ◯	false ◯
10. Unprefixed attributes are in the default namespace.	true ◯	false ◯

(For every correct answer 1.5 points, **for every incorrect answer -1.5 points**, for every unanswered question 0 points, you can have at least 0 points on this exercise)

The following Exercises 3 – 7 are referring to the XML document gamescollection.xml, which can be found on the last page of this exam.

Exercise 3: (12)

Complete the DTD **games.dtd**, so that XML documents structured like **gamescollection.xml** (see attachment) are valid according to this DTD. Consider the following points when creating the DTD:

- A `game` element contains exactly one `name` element, any number of `developer` elements, exactly one `year` element, at least one `platform` element, any number of `genre` elements and might contain one `dlcs` element (in that order).
- Each `game` element has an attribute `key` with a unique attribute value and might have an `rating` attribute storing a number between 0 and 10.
- The `dlcs` element contains one or more `dlc` elements.
- The `series` elements contain exactly one `name` element, followed by at least one `genre` element, and exactly one `games` element, which itself contains one or more `reference` elements.
- All `reference` elements have an attribute `ref` which refers to the `key` of a `game` element.
- If not specified make reasonable assumptions on the types of elements.

File **games.dtd**:

```
<!ELEMENT gamescollection (game | series)*>
```

Exercise 4: (10)

Consider the following XPath queries applied to the document **gamescollection.xml** (see attachement).

- If the given XPath expression selects the empty node set, write as output "empty output"
- If a number is selected as the result (count,sum,...), write as output this number.

Now give the outputs of the respective XPath queries:

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

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

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

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

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

Exercise 5: (6)

Consider the following XQuery expression **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>
```

Now give the output of **xquery.xq** applied to **gamescollection.xml**.

You do not need to consider whitespace issues.

Exercise 6: (10)

Create an XSLT-Stylesheet **xslt.xsl**, which returns, applied to documents of the form **gamescollection.xml** the following output:

- The output is an HTML document.
- Output all **PS4** games of a document of the form **gamescollection.xml**.
- If a game has a rating greater then **7**, also output the text: "**Rating:** Excellent Game!".
- Additionally, for every game list all series, which reference this game.

Consider the following output that your XSLT-Stylesheet **xslt.xsl** shall return applied to **gamescollection.xml**:

```
<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>
```

Now write the XSLT-Stylesheet **xslt.xsl**. You do not need to consider whitespace issues.

File **xslt.xsl**:

```
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

</xsl:stylesheet>
```

Exercise 7: (10)

Complete the method `insDLC`, which has two parameters `key` and `dlcName`. The method should use **DOM** to apply the following changes to the document stored in the variable `doc`:

- The dlc `dlcName` should be added to the game with the key `key`.
- If the element `dlcs` does not exist for this game, then such an element should be created as well.

You do not need to be concerned with error handling in this task. You can assume that a game with the key `key` exists in the document.

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

public void insDLC (String key, String dlcName) throws Exception {

}
```

Total points: 75

You may separate this page!

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>
```