EXAM IN "SEMI-STRUCTURED DATA" **184.705**			23. 10. 2015
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**:

```
<?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:choice>
            <xsd:element name="B" type="BType"/>
            <xsd:element name="C" type="xsd:string"/>
        </xsd:choice>
    </xsd:complexType>

    <xsd:complexType name="BType" mixed="true">
        <xsd:sequence>
            <xsd:element name="C" type="xsd:string" maxOccurs="2"/>
            <xsd:element name="B" type="BType" minOccurs="0" maxOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>

</xsd:schema>
```

Consider additionally the following eight different XML files. All of the following files are well-formed. In this exercise you have to decide, which of the following are valid according to **test.xsd**.

1. `<A>SSD</A>` valid ◯ invalid ⊗
2. `<A><C>SSD</C></A>` valid ⊗ invalid ◯
3. `<A><B>SSD</B></A>` valid ◯ invalid ⊗
4. `<A><B><C>SSD</C></B></A>` valid ⊗ invalid ◯
5. `<A><B>SSD<C>SSD</C></B></A>` valid ⊗ invalid ◯
6. `<A><B><C>SSD</C></B><C>SSD</C></A>` valid ◯ invalid ⊗
7. `<A><B><C>SSD</C><C>SSD</C></B></A>` valid ⊗ invalid ◯
8. `<A><B><C><C>SSD</C></C><B><C>SSD</C></B></B></A>` 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 is true or false.

1. Semi-structured data is a flexible format for exchanging data.	true ⊗	false ◯
2. The "M" in XML stands for model.	true ◯	false ⊗
3. XML can be used for storing digitized data such as photos.	true ◯	false ⊗
4. An element in an XML document cannot be empty.	true ◯	false ⊗
5. Namespaces can be used for disambiguating elements and attributes.	true ⊗	false ◯
6. Validating errors cannot be ignored.	true ◯	false ⊗
7. DTDs are more powerful than XML schemas.	true ◯	false ⊗
8. Event-based parsers use a constant amount of memory.	true ⊗	false ◯
9. XPath is a schema language.	true ◯	false ⊗
10. XSLT documents are XML documents.	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 bibliography.xml, which can be found on the last page of this exam.

Exercise 3: (10)

Create a DTD document **bibliography.dtd** such that the **bibliography.xml** document is valid. Consider the following specification:

- The root element of the document is called `bibliography`. It contains an unbounded number of `conference` and `journal` elements in an arbitrary order.
- The `conference` and the `journal` element contain exactly one `title` element and an unbounded number of `article` elements.
- The `article` element stores scientific articles. It has an attribute `id`, which is required and identifies this element. Furthermore, the `article` element contains exactly one `title` element, an unbounded number of `author` elements and an unbounded number of `ref` elements.
- The `author` element contains either a `fullname` element or a `firstname` and `lastname` element in an arbitrary order.
- The `ref` element is empty and has a required attribute `to`, which refers to an article id.

File **bibliography.dtd**:

```
<!ELEMENT bibliography (conference|journal)*>

<!ELEMENT conference (title,article*)>

<!ELEMENT journal (title,article*)>

<!ELEMENT article (title,author*,ref*)>
<!ATTLIST article id ID #REQUIRED>

<!ELEMENT author (fullname|((firstname,lastname)|(lastname,firstname)))>

<!ELEMENT ref EMPTY>
<!ATTLIST ref to IDREF #REQUIRED>

<!ELEMENT title (#PCDATA)>
<!ELEMENT fullname (#PCDATA)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT lastname (#PCDATA)>
```

Exercise 4: (10)

Consider the following XPath expressions and evaluate them over the **bibliography.xml** document.

- If the expression selects several nodes, separate the output with whitespaces.
- If the XPath expression selects no nodes, write "No output!".

Write the output of the following expressions:

```
count(//title)
```

```
7
```

```
//article[count(descendant::*)<6]/@id
```

```
a b d
```

```
count(//article[@id = //ref/@to]/author)
```

```
4
```

```
//*[count(./article) >= 2]/title/text()
```

```
AAAI '13
```

```
//article[@id = "d"]/ancestor::*/following::*/article/@id
```

```
No output!
```

Exercise 5: (6)

Consider the following XQuery statement **bibliography.xq**:

```
for $a in //article
for $r in //article[ref/@to = $a/@id]
order by $r/title descending, $a/title ascending
return <article>{$r/title/text()} - {$a/title/text()}</article>
```

Write the output of **bibliography.xq** evaluated over **bibliography.xml** here.
Whitespaces don't have to be formatted correctly.

```
<article>Paper 4 - Paper 3</article>
<article>Paper 3 - Paper 1</article>
<article>Paper 3 - Paper 2</article>
<article>Paper 1 - Paper 2</article>
```

Exercise 6: (10)

Create an XSLT stylesheet **bibliography.xsl** that, after applied to **bibliography.xml**, outputs the following XML document, which gives some statistics about the articles stored in **bibliography.xml**:

```
<?xml version="1.0" encoding="UTF-8"?>
<stats>
  <a>
    <authors>1</authors>
    <references>1</references>
  </a>
  <b>
    <authors>1</authors>
    <references>0</references>
  </b>
  <c>
    <authors>2</authors>
    <references>2</references>
  </c>
  <d>
    <authors>1</authors>
    <references>1</references>
  </d>
</stats>
```

This means:

- The root element is `stats`
- For each article with id `x`, create an `x` element. This `x` element has one `authors` and one `references` element as children.
- The content of the `authors` element is the number of author elements in the article element with id `x`.
- The content of the `references` element is the number of ref elements in the article element with id `x`.

Write the stylesheet here **bibliography.xsl**.

File **pruefung.xsl**:

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

   <xsl:template match="bibliography">
      <stats><xsl:apply-templates select="*" /></stats>
   </xsl:template>

   <xsl:template match="conference">
      <xsl:apply-templates select="article" />
   </xsl:template>

   <xsl:template match="journal">
      <xsl:apply-templates select="article" />
   </xsl:template>

   <xsl:template match="article">
      <xsl:element name="{@id}">
         <authors><xsl:value-of select="count(author)" /></authors>
         <references><xsl:value-of select="count(ref)" /></references>
      </xsl:element>
   </xsl:template>

</xsl:stylesheet>
```

Exercise 7: (12)

Complete the following SAX handler that, after applied to the **bibliography.xml** document, outputs a list of article titles together with the number of citations. The output should be as follows:

```
Paper 1: 1 Citations.
Paper 2: 2 Citations.
Paper 3: 1 Citations.
Paper 4: 0 Citations.
```

For example, the article with the name "Paper 2" is referenced by article "Paper 1" and "Paper 3", i.e. by **2** articles.

The format of the output is not important. The variable `xPath` can be used to evaluate XPath expressions. The variable `bibDocument` contains the XML document on which this SAX handler is run on. You don't need to handle any exceptions.

```
public class PrintCitations extends DefaultHandler {

   static XPath xPath = XPathFactory.newInstance().newXPath();;
   private Document bibDocument;
   private String eleText;

   //Additional variables
   private int citations;
   private boolean inArticle;

   public void characters(char[] text, int start, int length) throws SAXException {
     eleText = new String(text, start, length);
   }

   public PrintCitations(Document bibDocument) {
      this.bibDocument = bibDocument;
   }

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

      if ("article".equals(localName)) {
         inArticle = true;
         try {
            XPathExpression xpathExpr = xPath.compile("//article/ref[@to = '"+
                                                    atts.getValue("id")+"']");
            NodeList articleList = (NodeList)xpathExpr.evaluate(bibDocument,
                                                              XPathConstants.NODESET);
            citations = articleList.getLength();
         } catch (XPathExpressionException e) {
            e.printStackTrace();
         }
      }
   }

   public void endElement(String namespaceURI, String localName, String qName) throws SAXException {
      if ("article".equals(localName)) inArticle = false;
      if ("title".equals(localName) && inArticle) {
         System.out.println(eleText + ": " + citations + " Citations.");
      }
   }
}
```

Total points: 75

You can remove this sheet!

File **bibliography.xml**:

```
<bibliography>
  <conference>
    <title>AAAI '13</title>
    <article id="a">
      <title>Paper 1</title>
      <author>
        <fullname>Jo Mi</fullname>
      </author>
      <ref to="b"/>
    </article>
    <article id="b">
      <title>Paper 2</title>
      <author>
        <firstname>John</firstname>
        <lastname>Meyer</lastname>
      </author>
    </article>
  </conference>
  <journal>
    <title>Jornal of AI</title>
    <article id="c">
      <title>Paper 3</title>
      <author>
        <lastname>Meyer</lastname>
        <firstname>John</firstname>
      </author>
      <author>
        <fullname>Jo Mi</fullname>
      </author>
      <ref to="a"/>
      <ref to="b"/>
    </article>
  </journal>
  <conference>
    <title>AAAI '14</title>
    <article id="d">
      <title>Paper 4</title>
      <author>
        <fullname>Jo Mi</fullname>
      </author>
      <ref to="c"/>
    </article>
  </conference>
</bibliography>
```