EXAM IN "SEMI-STRUCTURED DATA" **184.705**			22. 06. 2016
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 DTD schema file **test.dtd**:

```
<!ELEMENT A ((A | B), C?, B)>

<!ELEMENT B (#PCDATA | A | C)*>

<!ELEMENT C EMPTY>

<!ATTLIST A id ID #IMPLIED>

<!ATTLIST C letter (a|b|c|d) #IMPLIED>
```

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.dtd**, assuming that the root element is A.

1. `<A><B/><B/></A>` valid ⊗ invalid ◯
2. `<A/>` valid ◯ invalid ⊗
3. `<A id="a"><B><C/></B><B>test</B></A>` valid ⊗ invalid ◯
4. `<A id="a"><A id="b"><B/><B/></A><C/><B/></A>` valid ⊗ invalid ◯
5. `<A id="b"><B>test<A><B/><B/></A>test</B><B/></A>` valid ⊗ invalid ◯
6. `<A id="a"><B/><B><A id="b"><B/><B/><B/></A></B></A>` valid ◯ invalid ⊗
7. `<A id="a"><B/><B><A id="a"><B/><B/></A></B></A>` valid ◯ invalid ⊗
8. `<A id="a"><B><C letter="a"/></B><C letter="d"/><C letter="c"/></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 are true or false.

1.	Structured data can be represented as a graph.	true ⊗ false ◯
2.	The "X" in XML stands for eXchangeable.	true ◯ false ⊗
3.	An XML document is not a database.	true ⊗ false ◯
4.	An XML document must be well-formed.	true ⊗ false ◯
5.	DTDs are not XML documents.	true ⊗ false ◯
6.	Validating errors cannot be ignored.	true ◯ false ⊗
7.	DTDs are more expressive than XML schemas.	true ◯ false ⊗
8.	Tree-based parsers use a constant amount of memory.	true ◯ false ⊗
9.	XPath is a query language.	true ⊗ false ◯
10.	XPath is more powerful than XSLT.	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 euro.xml, which can be found on the last page of this exam.

Exercise 3: (10)

Complete the following XML Schema document **euro.xsd** such that the **euro.xml** document is valid. Consider the following specification:

- You only have to complete the type `qualifyingType`, the rest of the schema is already given.
- The element `qualifying` has at least one and at most 6 `group` elements.
- Every element `group` has exactly four `team` elements, followed by zero or an unbounded number of `news` elements.
- The content of a `news` element is mixed. The `news` element is allowed to have zero or an unbounded number of `player` or `team` elements in any order.
- All attributes are required and of type `xs:string`.
- It is **not** required to define keys and key references.

File **euro.xsd**:

```
 <!-- More space on the following page! -->

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="euro">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="teams" type="teamsType"/>
        <xs:element name="player" maxOccurs="unbounded" type="playerType"/>
        <xs:element name="qualifying" type="qualifyingType"/>
      </xs:sequence>
      <xs:attribute name="year" type="xs:nonNegativeInteger" use="required"/>
    </xs:complexType>
  </xs:element>

  <xs:complexType name="teamsType">
    <xs:sequence>
      <xs:element name="team" maxOccurs="unbounded">
        <xs:complexType>
          <xs:attribute name="shortname" type="xs:string" use="required"/>
          <xs:attribute name="name" type="xs:string" use="required"/>
          <xs:attribute name="champion">
            <xs:simpleType>
              <xs:restriction base="xs:string">
                <xs:enumeration value="yes"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
```

```
  <xs:complexType name="playerType">
    <xs:attribute name="id" type="xs:integer" use="required"/>
    <xs:attribute name="name" type="xs:string" use="required"/>
    <xs:attribute name="team" type="xs:string" use="required"/>
    <xs:attribute name="no" type="xs:integer" use="required"/>
  </xs:complexType>

  <xs:complexType name="qualifyingType">

    <xs:sequence>
      <xs:element name="group" maxOccurs="6">
        <xs:complexType>
          <xs:sequence>

            <xs:element name="team" type="xs:string" minOccurs="4" maxOccurs="4"/>
            <xs:element name="news" minOccurs="0">
              <xs:complexType mixed="true">
                <xs:choice maxOccurs="unbounded">

                  <xs:element name="team">
                    <xs:complexType>
                      <xs:attribute name="short" type="xs:string" use="required"/>
                    </xs:complexType>
                  </xs:element>
                  <xs:element name="player">
                    <xs:complexType>
                      <xs:attribute name="id" type="xs:string" use="required"/>
                    </xs:complexType>
                  </xs:element>

                </xs:choice>
              </xs:complexType>
            </xs:element>

          </xs:sequence>
          <xs:attribute name="name" type="xs:string" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>

  </xs:complexType>
</xs:schema>
```

Exercise 4: (9)

Write an XPath expression for the following queries. These expressions will be evaluated over documents that are valid with respect to **euro.xsd**.

1. Select all player elements with number (no) 11.

```
//player[@no='11']
```

2. Select the shortname of all countries that have not been a champion.

```
//teams/team[not(@champion)]/@shortname
```

3. Select the groups with only non-champion teams.

```
//group[count(team[.=//teams/team[not(@champion)]/@shortname])=4]
```

Exercise 5: (10)

Consider the following XQuery **squads.xq**:

```
<squads>
{for $team in //teams/team
 where $team/@shortname = ("FRA","WAL","SVK")
 return
    <nation name="{$team/@name}">
    {for $player in //player
     where $player/@team = $team/@shortname
     return
        <player no="{$player/@no}">
            {$player/@name}
        </player>}
    </nation>}
</squads>
```

Write the output of **squads.xq** evaluated over **euro.xml**.
Whitespaces do not have to be formatted correctly.

```
<squads>
   <nation name="France">
      <player no="20" name="Kingsley Coman"/>
      <player no="7" name="Antoine Griezmann"/>
   </nation>
   <nation name="Slovakia"/>
   <nation name="Wales">
      <player no="11" name="Gareth Bale"/>
   </nation>
</squads>
```

Exercise 6: (10)

Create an XSLT stylesheet **squads.xsl** that, after applied to **euro.xml**, outputs the same XML document as the XQuery **squads.xq** on the previous page.

It is **not** allowed to use `xsl:for-each` and `xsl:if`.

Write the stylesheet **squads.xsl** below.

File **groups.xsl**:

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

<xsl:template match="/">
    <squads>
        <xsl:apply-templates select="//teams/team[@shortname=('FRA','WAL','SVK')]" />
    </squads>
</xsl:template>

<xsl:template match="team">
    <nation name="{@name}">
        <xsl:variable name="team" select="."/>
        <xsl:apply-templates select="//player[@team=$team/@shortname]"/>
    </nation>
</xsl:template>

<xsl:template match="player">
    <player trikotnr="{@no}">
        <xsl:value-of select="@name"/>
    </player>
</xsl:template>

</xsl:stylesheet>
```

Exercise 7: (9)

Consider the method `run` given below. Write the output of this method `run`, assuming the variable `doc` contains the DOM representation of the XML document **euro.xml**.

```
public void run() throws Exception {
    NodeList list = doc.getElementsByTagName("news");
    for (int i = 0; i < list.getLength(); i++) {
        Node n = list.item(i).getFirstChild();
        String s = "";
        do {
            switch (n.getNodeType()) {
                case Node.TEXT_NODE:
                    s += n.getNodeValue();
                    break;
                case Node.ELEMENT_NODE:
                    if (n.getNodeName().equals("team")) {
                        XPathExpression xpe = xPath.compile("//teams/team[@shortname = \""
                                + n.getAttributes().getNamedItem("short").getNodeValue() + "\"]");
                        Node name = (Node) xpe.evaluate(doc, XPathConstants.NODE);
                        s += name.getAttributes().getNamedItem("name").getNodeValue();
                    } else {
                        XPathExpression xpe = xPath.compile("//player[@id = \""
                                + n.getAttributes().getNamedItem("id").getNodeValue() + "\"]");
                        Node name = (Node) xpe.evaluate(doc, XPathConstants.NODE);
                        s += name.getAttributes().getNamedItem("name").getNodeValue();
                    }
            }
            n = n.getNextSibling();
        } while (n != null);
        System.out.println(s);
    }
}
```

```
The last match in Group A ended with a draw between France
and Switzerland. Yann Sommer was selected man of
the match.
```

Total points: 75

You can remove this sheet!

File **euro.xml**:

```
<euro year="2016">
    <teams>
        <team shortname="SUI" name="Switzerland" champion="yes" />
        <team shortname="FRA" name="France" champion="yes" />
        <team shortname="ROU" name="Romania" />
        <team shortname="ALB" name="Albania" />
        <team shortname="ENG" name="England" />
        <team shortname="SVK" name="Slovakia" />
        <team shortname="WAL" name="Wales" />
        <team shortname="RUS" name="Russia" />
        <!-- .... -->
    </teams>
    <player id="0210" name="Gareth Bale" team="WAL" no="11"/>
    <player id="0230" name="Jamie Vardy" team="ENG" no="11"/>
    <player id="0160" name="Daniel Sturridge" team="ENG" no="15"/>
    <player id="0301" name="Kingsley Coman" no="20" team="FRA"/>
    <player id="0312" name="Antoine Griezmann" no="7" team="FRA"/>
    <player id="0314" name="Yann Sommer" no="1" team="CHF"/>
    <!-- .... -->
    <qualifying>
        <group name="A">
            <team>FRA</team><team>SUI</team><team>ROU</team><team>ALB</team>
            <news>
                The last match in Group A ended with a draw between <team short="FRA" />
                and <team short="CHF" \>. <player id="0314" /> was selected man of
                the match.
            </news>
        </group>
        <group name="B">
            <team>ENG</team><team>SVK</team><team>WAL</team><team>RUS</team>
        </group>
        <!-- .... -->
    </qualifying>
</euro>
```