

# Compito di Tecnologie WEB e di Internet T-B (Prof. Grandi) del 31 gennaio 2011

Dato il seguente frammento XML contenuto all'interno di un file esami.xml:

```
<esame id="235">
  <studente matricola="773482365">Alienati Prima</studente>
  <materia>Impianti Inconsci</materia><data>2011-01-10</data>
  <voto>30</voto><lode>NO</lode>
</esame>
<esame id="236">
  <studente matricola="7734845453">de Coccio Ugo</studente>
  <materia>Filosofia dello Sport</materia><data>2011-01-16</data>
  <esito>respinto</esito>
</esame>
<esame id="237">
  <studente matricola="7734863330">Allievi Vetusto</studente>
  <materia>Gastronomia Archeologica</materia><data>2011-01-25</data>
  <voto>27</voto>
</esame>
```

## ESERCIZIO 1

Si fornisca il codice **XMLSchema** che definisce un **tipo** per l'**element esame** compatibile con tale frammento XML.

## ESERCIZIO 2

Con riferimento al file esami.xml di cui sopra, si descriva l'effetto del seguente codice **XSLT**:

```
<xsl:template match="/">
<html><body>
  <h2>Esami</h2>
  <table border="1">
    <tr> <th>Materia</th> <th>Studente</th> <th>Voto</th> </tr>
    <xsl:for-each select="//esame [voto]">
      <tr>
        <td><xsl:value-of select="materia"/></td>
        <td><xsl:value-of select="studente"/></td>
        <td><xsl:value-of select="voto"/>
          <xsl:if test="lode='SI'">L</xsl:if></td>
      </tr>
    </xsl:for-each>
  </table>
</body></html>
</xsl:template>
```

## ESERCIZIO 3

Si scriva una interrogazione **XQuery** che svolga il medesimo compito del foglio di stile dell'Es. 2, ma limitatamente ai soli esami sostenuti dopo il 15 gennaio 2011.

## ESERCIZIO 4 (facoltativo)

Si illustri brevemente che cosa sono le **Ontologie** e quale ruolo giocano nel **Semantic Web**.

## SOLUZIONE ESERCIZIO 1

Usando definizioni locali, una possibile soluzione è:

```
<xs:complexType name="tipo_esame">
  <xs:sequence>
    <xs:element name="studente">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="matricola" type="xs:string" use="required"/>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="materia" type="xs:string"/>
    <xs:element name="data" type="xs:date"/>
    <xs:choice>
      <xs:element name="esito" type="xs:string"/>
      <xs:sequence>
        <xs:element name="voto" type="xs:string"/>
        <xs:element name="lode" type="xs:string" minOccurs="0"/>
      </xs:sequence>
    </xs:choice>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
</xs:complexType>
```

Volendo specificare ulteriori vincoli sui tipi di dato, tale soluzione per esempio diventa:

```
<xs:complexType name="tipo_esame">
  <xs:sequence>
    <xs:element name="studente">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="matricola" use="required">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:pattern value="[0-9]{10}"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:element name="materia" type="xs:string"/>
    <xs:element name="data" type="xs:date"/>
    <xs:choice>
      <xs:element name="esito">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="ritirato"/>
            <xs:enumeration value="respinto"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
      <xs:sequence>
        <xs:element name="voto">
          <xs:simpleType>
```

```

        <xs:restriction base="xs:integer">
            <xs:minInclusive value="18"/>
            <xs:maxInclusive value="30"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="lode" minOccurs="0">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:pattern value="SI|NO"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:choice>
</xs:sequence>
<xs:attribute name="id" type="xs:string" use="required"/>
</xs:complexType>

```

## SOLUZIONE ESERCIZIO 2

Il foglio di stile seleziona materia, studente e voto di tutti gli esami superati (il pattern dell'istruzione for-each richiede che l'element esame cui si applica abbia voto come figlio) e produce un risultato in forma di tabella all'interno di una pagina HTML. Se è presente lode con valore SI, viene aggiunta una "L" alla stringa che codifica il voto, es:

```

<html><body>
<h2>Esami</h2>
<table border="1">
  <tr> <th>Materia</th> <th>Studente</th> <th>Voto</th> </tr>
  <tr> <td>Analisi</td> <td>Mario Rossi</td> <td>27</td> </tr>
  <tr> <td>Fisica</td> <td>Anna Violi</td> <td>30L</td> </tr>
  <tr> <td>Storia</td> <td>Paolo Neri</td> <td>30</td> </tr>
  ...
</table>
</body></html>

```

## SOLUZIONE ESERCIZIO 3

```

<html><body>
<h2>Esami</h2>
<table border="1">
<tr> <th>Materia</th> <th>Studente</th> <th>Voto</th> </tr>
{
for $es in doc("esami.xml")//esame[voto]
  where $es/data > "2010-05-31"
  return
    <tr>
      <td>{ $es/materia/text() }</td>
      <td>{ $es/studente/text() }</td>
      <td>{ $es/voto/text(), if ($es/lode/text() = "SI") then "L" else () }
    </td>
    </tr>
}
</table>
</body></html>

```