Examen intra – INF4375 – Automne 2011
30% de la note finale
21 octobre 2011 13h30 à 16h30

Aucune documentation n'est permise. Répondez dans le cahier fourni à cet effet.

Question #1 – 16%

Considérant le document XML suivant :

```
<?xml version="1.0" encoding="utf-8"?>
<simple xmlns="http://jberger.org/inf4375/"
        xmlns:xtr="http://www.cirpa.ca/"
        xmlns:ccs="http://www.ccs.ca/home/index_e.aspx">
   <ccs:tapis xmlns="http://www.canadianwebhosting.com/">
      <eveil xmlns:ter="http://www.ter-sncf.com/">Data provider</eveil>
   </ccs:tapis>
   <xtr:tinyl xmlns="http://www.mjk.com/"
              xmlns:ssre="http://www.sgbf.ch/index_fr.html">
      <bear>
         <ssre:mjk>Personal Data</ssre:mjk>
         <th7 xmlns="http://th7.org/">
            <terne>Panda</terne>
         </th7>
      </bear>
   </xtr:tinyl>
</simple>
```

Quelle est la valeur du namespace (et non pas le préfixe) de l'élément suivant :

1. simple
2. tapis
3. eveil
4. tinyl
5. bear
6. mjk
7. th7
8. terne

Question #2 – 7%

L'architecture REST est reconnue pour sa légèreté. Pourquoi?

Question #3 – 7%

Quel est le rôle de l'objet XmlHttpRequest dans le paradigme Ajax?

Question #4 – 5%

Décrivez un scénario où il est plus avantageux d'utiliser un parser StAX plutôt qu'un parser DOM.

Question #5 – 10%

À partir du document XML suivant :

```
<?xml version="1.0" encoding="utf-8"?>
<library>
	<documents>
		<document type="book">
			<title>Théorie de la musique</title>
			<author>A. Danhauser</author>
			<year>1996</year>
		</document>
		<document type="book">
			<title>Test-Driven Development by Example</title>
			<author>Kent Beck</author>
			<year>2003</year>
		</document>
		<document type="book">
			<title>Design Patterns</title>
			<author>Erich Gamma</author>
			<author>Richard Helm</author>
			<author>Ralph Johnson</author>
			<author>John Vlissides</author>
			<year>1995</year>
		</document>
		<document type="article">
			<title>XML Schema</title>
			<author>Eric van der Vlist</author>
			<year>2001</year>
		</document>
	</documents>
</library>
```

Évaluez ces expressions XPath :

1. `//document[@type = 'book' and count(author) = 1]/year[. < 2000]/preceding-sibling::title`
2. `//year[preceding-sibling::* = 'Erich Gamma']`
3. `//document[string-length(author) > 12]/@type`
4. `//*[. = 1996]/../*[count(following-sibling::*) = 2]`
5. `/library/documents/child::*/year[parent::*[@type = 'article']]`

Question #6 – 7%

Quelles sont les trois endroits où l'on peut définir un style CSS dans une page HTML et quelle est la priorité accordée à chacun de ces endroits?

Question #7 – 7%

Quelle est la différence entre les axes following et following-sibling avec XPath?

Question #8 – 10%

Écrivez un document XML qui respecte ce schéma :

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="rue">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" name="adresses">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" maxOccurs="unbounded" name="no"
                          type="xs:int" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="nom" type="xs:string" />
      <xs:attribute name="codepostal" type="xs:string" />
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Question #9 – 10%

Appliquez la transformation XSL donnée sur le document XML donné.

Voici le document XML :

```
<?xml version="1.0" encoding="UTF-8"?>
<class>
    <name>Personne</name>
    <attribute>
        <name>Nom</name>
        <type>string</type>
    </attribute>
    <attribute>
        <name>Prenom</name>
        <type>string</type>
    </attribute>
    <attribute>
        <name>Age</name>
        <type>integer</type>
    </attribute>
</class>
```

Voici le XSL :

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
    <xsl:output method="text" indent="no"/>

    <xsl:template match="/class">
        <xsl:text>public class </xsl:text>
        <xsl:value-of select="name"/>
        <xsl:text> {
        </xsl:text>
        <xsl:apply-templates select="attribute"/>
        <xsl:text>
}        </xsl:text>
    </xsl:template>

    <xsl:template match="attribute">
        <xsl:text>private </xsl:text>
        <xsl:choose>
            <xsl:when test="type = 'string'">
                <xsl:text>String</xsl:text>
            </xsl:when>
            <xsl:when test="type = 'integer'">
                <xsl:text>int</xsl:text>
            </xsl:when>
        </xsl:choose>
        <xsl:text> </xsl:text>
        <xsl:value-of select="name"/>
        <xsl:text>;
        </xsl:text>
        <xsl:apply-templates select="name | type"
                             mode="GenerateGettersAndSetters"/>
        <xsl:text>

        </xsl:text>
    </xsl:template>
</xsl:stylesheet>
```

Question #10 – 7%

Décrivez les 3 formes de design que l'on peut utiliser lors de l'écriture d'un schéma XSD.

Question #11 – 7%

Parmi les 8 erreurs courantes en développement d'applications distribuées, identifiez 2 erreurs et expliquez les brièvement.

Question #12 – 7%

Quelle est la différence entre un parser XML de type push et un autre de type pull?