

Interrogation écrite

Exercice 1 : Lecture sax d'un fichier xml

On considère le gestionnaire **sax** défini par la classe **InterroHandler** suivante :

```
1 import org.xml.sax.* ;
2 import org.xml.sax.helpers.* ;
3 public class InterroHandler extends DefaultHandler {
4     private int n ;
5     private boolean b ;
6
7     public void startDocument() {this.n = 0 ; this.b = false ;}
8     public void endDocument() {System.out.println(n) ;}
9     public void startElement(String namespaceURI , String localName ,
10        String rawName , Attributes attributs) {
11         this.b = true ;
12     }
13     public void endElement(String namespaceURI , String localName ,
14        String rawName) {
15         if (b) n++ ; b = false ;
16     }
17     public static void main(String [] args) {
18         try {
19             XMLReader saxReader = XMLReaderFactory.createXMLReader() ;
20             saxReader.setContentHandler(new InterroHandler()) ;
21             saxReader.parse(args[0]) ;
22         } catch (Exception t) {
23             t.printStackTrace() ;
24         }
25     }
26 }
```

On considère aussi le fichier **a.xml** suivant :

```
<?xml version="1.0" encoding="UTF-8"?>
<a> <b/> <c> <d/> <e> <f/> </e> </c> <g/> <h> <i /> </h> </a>
```

Question 1 : (1,5 pt) Donner l'affichage obtenu par l'invocation `java InterroHandler a.xml`.

Exercice 2 : Requêtes XPath

On considère la DTD suivante :

```
<!ELEMENT liste ( livre )*>
<!ELEMENT livre ( titre , auteur , parution )*>
<!ELEMENT titre ( #PCDATA )>
  <!-- ATTLIST titre genre CDATA #REQUIRED -->
<!ELEMENT auteur ( #PCDATA )>
<!ELEMENT parution ( #PCDATA )>
```

dont un exemple d’instance XML est donné en annexe, figure 1.

Question 1 : Donnez des requêtes XPATH permettant d’extraire de tout fichier satisfaisant la DTD lib.dtd :

1. tous les nœuds **titre** présents dans le document ; (0,5 pt)

```
// titre
```

2. tous les nœuds **auteur** présents dans le document ; (0,5 pt)

```
// auteur
```

3. tous les nœuds **titre** correspondant à des livres du genre **jeu** ; (1 pt)

```
// titre [ @genre="jeu " ]
```

4. tous les nœuds attributs **genre** ; (1 pt)

```
// @genre
```

5. le nœud **titre** du premier livre qui est du genre **photo** ; (2 pts)

```
// livre [ titre / @genre="photo" ] [ 1 ] / titre
```

6. tous les nœuds descendants du deuxième livre ; (1 pt)

```
// livre [ 2 ] / descendant::node ( )
```

7. tous les nœuds **titre** des frères suivant le premier livre. (2 pts)

```
// livre [ 1 ] / following-sibling::*/ titre
```

Exercice 3 : XML-Schema

On considère le fichier **compagnie.xml** donné en annexe, figure 2.

Question 1 : (2 pts) Ce fichier satisfait un schéma de nom **compagnie.xsd**. Donner le (un) contenu (cohérent) de la balise ouvrante **<xsd:schema >** du schéma **compagnie.xsd**.

```
<xsd:schema
  xsi:schemaLocation=" http://www.w3.org/2001/XMLSchema http://www.w3 →
    ↪ .org/2001/XMLSchema.xsd "
  targetNamespace=" http://exo-schema "
  elementFormDefault=" qualified "
  xmlns:xsd=" http://www.w3.org/2001/XMLSchema "
  xmlns:xsi=" http://www.w3.org/2001/XMLSchema-instance "
  xmlns=" http://exo-schema ">
```

Question 2 : (1 pt) Donner la définition du type de l'élément **fonction** du schéma **compagnie.xsd**, sachant les seules valeurs possibles de ce type sont **Professeur** et **Maitre-de-Conferences**.

```
<xsd:simpleType name="type-fonction">
  <xsd:restriction base="xsd:token">
    <xsd:enumeration value="Professeur"/>
    <xsd:enumeration value="Maitre-de-Conferences"/>
  </xsd:restriction>
</xsd:simpleType>
```

Question 3 : (1 pt) Donner la définition du type de l'élément **mail** du schéma **compagnie.xsd**, sachant que les seules valeurs autorisées pour cet élément sont de la forme **string.string@string.string**.

```
<xsd:simpleType name="type-mail">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value=".[.].+@.[.].+"/>
  </xsd:restriction>
</xsd:simpleType>
```

Question 4 : (1 pt) Donner la définition du type de l'attribut **numen** du schéma **compagnie.xsd**, sachant qu'un **numen** est une valeur entière comprise entre 10000 et 99999.

```
<xsd:simpleType name="type-numen">
  <xsd:restriction base="xsd:nonNegativeInteger">
    <xsd:minInclusive value="10000"/>
    <xsd:maxInclusive value="99999"/>
  </xsd:restriction>
</xsd:simpleType>
```

Question 5 : (1,5 pt) Donner la définition de l'élément **personne** du schéma **compagnie.xsd**, sachant que l'attribut **numen** est obligatoire.

```
<xsd:element name="personne">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="nom" type="xsd:string"/>
      <xsd:element name="prenom" type="xsd:string"/>
      <xsd:element name="mail" type="type-mail"/>
      <xsd:element name="fonction" type="type-fonction"/>
    </xsd:sequence>
    <xsd:attribute name="numen" type="type-numen" use="required"/>
  </xsd:complexType>
</xsd:element>
```

Question 6 : (2 pts) On souhaite définir comme clé primaire, pour toute personne, la valeur de son attribut `numen`. En précisant dans quelle définition d’élément du fichier `compagnie.xsd` faire apparaître cette définition de clé, donner les instructions `xml-schema` correspondantes.

```
<!-- dans l'element personnel -->
  <xsd:key name="clef-personne">
    <xsd:selector xpath="personne"/>
    <xsd:field xpath="@numen"/>
  </xsd:key>
```

Question 7 : (2 pts) On souhaite ajouter comme contrainte que le responsable d’un service doit être un membre du personnel. En précisant dans quelle définition d’élément du fichier `compagnie.xsd` faire apparaître cette contrainte, donner les instructions `xml-schema` correspondantes.

```
<!-- dans l'element compagnie -->
  <xsd:keyref name="contrainte-responsable" refer="clef-personne">
    <xsd:selector xpath="services/service"/>
    <xsd:field xpath="responsable"/>
  </xsd:keyref>
```

Annexe

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE liste SYSTEM "lib.dtd">
<liste>
  <livre>
    <titre genre="jeu">Le Texas Hold 'Em Poker online</titre>
    <auteur>Mark Stohan</auteur>
    <auteur>Robert Bluman</auteur>
    <parution>2006</parution>
  </livre>
  <livre>
    <titre genre="jeu">Sudoku Manga</titre>
    <auteur>Sudoku factory</auteur>
    <parution>2007</parution>
  </livre>
  <livre>
    <titre genre="jeu">Kakoku</titre>
    <auteur>Hizi Kagochi</auteur>
    <parution>2005</parution>
  </livre>
  <livre>
    <titre genre="photo">Manuel de la photo</titre>
    <auteur>Jackie Contiboeuf</auteur>
    <auteur>Alain Mocney</auteur>
    <parution>2006</parution>
  </livre>
</liste>
```

FIGURE 1 – Une instance XML correspondant à la DTD de l'exercice 2.

```
<?xml version="1.0" encoding="utf-8"?>

<compagnie xmlns="http://exo-schema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://exo-schema http://www.fil.univ-
    ↳ lille1.fr/compagnie.xsd">
  <nom>UFR IEEA</nom>
  <adresse>
    <rue>Cite scientifique</rue>
    <CP>59655</CP>
    <ville>Villeneuve d'Ascq cedex</ville>
    <pays>France</pays>
  </adresse>
  <personnel>
    <personne numen="59678">
      <nom>Routier</nom>
      <prenom>Jean-Christophe</prenom>
      <mail>jean-christophe.routier@univ-lille1.fr</mail>
      <fonction>Professeur</fonction>
    </personne>
    <personne numen="59864">
      <nom>Roos</nom>
      <prenom>Yves</prenom>
      <mail>yves.roos@univ-lille1.fr</mail>
      <fonction>Maitre-de-Conferences</fonction>
    </personne>
  </personnel>
  <services>
    <service>
      <nom>Informatique</nom>
      <batiment>M3</batiment>
      <responsable>59678</responsable>
    </service>
  </services>
</compagnie>
```

FIGURE 2 – Le fichier `compagnie.xml` de l'exercice 3.