

XML - TP2 : Dom

L3 Miage - 2008/2009

January 14, 2011

1 Création d'un doc xml

Utilisation de l'implémentation ORACLE de DOM sous JDeveloper ¹.

Exemple

Ainsi le programme suivant

```
import java.io.*;
import org.w3c.dom.*;
import javax.xml.parsers.*;

public class creaXML
{
    public static void main(String[] args) throws Exception
    {
        Attr a; Element e, e1, l ;
        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
        dbf.setIgnoringElementContentWhitespace(true);
        DocumentBuilder db = dbf.newDocumentBuilder();
        Document d = db.newDocument();
        Element rac = d.createElement("Mandats");
        rac.setAttribute("etablissement","petite ville");
        rac.setAttributeNode(a= d.createAttribute("exercice"));
        a.setValue("2005");
        (e = d.createElement("Mandat")).setAttribute("numero","101");
        (e1 = ((Element) e.cloneNode(true))).setAttribute("numero","102");
        e.appendChild(l = d.createElement("ligne")).appendChild(d.createElement("libelle"));
        appendChild(d.createTextNode("Cle USB"));
        l.appendChild(d.createElement("montant")).appendChild(d.createTextNode("60"));
        d.appendChild(rac).appendChild(e); rac.appendChild(e1);
        XMLTools.ecrireXML(d, "c:/m14.xml",null);
    }
}
```

fabrique le document XML suivant

¹ajouter pour le projet dans ses librairies Oracle XML Parser V2

```

<Mandats etablisement="petite ville" exercice="2005">
  <Mandat numero="101">
    <ligne>
      <libelle>CléUSB</libelle>
      <montant>60</montant>
    </ligne>
  </Mandat>
  <Mandat numero="102"/>
</Mandats>

```

Question 1 Ecrire un programme JAVA/DOM construisant le document XML (fichier) suivant:

```

<catalogue>
  <article nom="peigne">
    <prix unite="euro">10</prix>
  </article>
  <article nom="brosse">
    <prix unite="dollar">5</prix>
  </article>
  <article nom="shampooing">
    <prix>4</prix>
  </article>
  <article nom="laque">
    <prix unite="yen">9</prix>
  </article>
</catalogue>

```

2 Modification d'un document XML

Exemple

qui utilise l'implémentation de base (API Java) de DOM

```

import java.io.*;
import org.w3c.dom.*;
import javax.xml.parsers.*;
public class ModifXML
{
  public static void main(String[] args) throws Exception
  {
    DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
    dbf.setIgnoringElementContentWhitespace(true);
    DocumentBuilder db = dbf.newDocumentBuilder();
    Document document = db.parse(new File("c:/m14.xml"));
    //ajout d'un attribut date="20/12/2005" au mandat 101
  }
}

```

```

//ajout d'une ligne au mandat 102 comportant un libelle "pile" et un montant "3"
NodeList l = document.getElementsByTagName("Mandat");
for (int i=0;i<l.getLength();i++)
{ Element e = (Element) l.item(i);
  if (e.getAttributeNode("numero").getValue().equals("101"))
    e.setAttribute("date","20/12/2005");
  if (e.getAttributeNode("numero").getValue().equals("102"))
  { Element aj = document.createElement("ligne");
    Text t = document.createTextNode("pile");
    e.appendChild(aj); aj.appendChild(t);
    aj = document.createElement("montant");
    t = document.createTextNode("3");
    e.appendChild(aj);aj.appendChild(t);  }
}
// outils fourni
XMLTools.ecrireXML(document,"c:/m14.xml",null);}
}

```

Question 2 A travers une application JAVA, modifier le document XML issu de la question précédente de sorte que le coût du *shampoing* soit en euros; que *la laque* disparaisse et d'ajouter entre *peigne* et *brosse* la *perruque* à 90 euros.

3 Expression arithmétique XML

On considère un mode de représentation XML d'expressions arithmétiques dans lesquelles apparaissent des constantes nommées. On se limitera en termes d'opérations à : +, -, *, /.

Dans la représentation XML, les constantes nommées et les expressions porteront des identifiants (attribut *nom*).

Question 3 A partir de l'exemple suivant, inférer une DTD possible couvrant tous les documents valides de ce type.

```

<formules>
  <constantes>
    <const valeur="3" nom="X"/>
    <const valeur="78" nom="Y"/>
    <const valeur="45" nom="Z"/>
  </constantes>
  <exps>
    <exp nom="R1">
      <op symbole="+>
        <var nom="X"/>
        <var nom="Z"/>
      </op>
    </exp>
    <exp nom="R2">
      <op symbole="+>
        <op symbole="*>

```

```

        <var nom="X"/>
        <var nom="X"/>
    </op>
    <op symbole="*">
        <var nom="Y"/>
        <var nom="Y"/>
    </op>
</op>
</exp>
</exps>
</formules>

```

Question 4 Quel fragment XML faut-il intégrer à ce document pour y inclure l'expression suivante : $x * (x * (x * (y + z)))$ de nom R3.

3.1 DOM

Dans le cadre d'une application JAVA utilisant l'Api DOM, on se propose d'évaluer les expressions contenues dans un tel document; ainsi pour l'exemple présenté, on obtiendrait comme résultat de l'exécution de cette application : R1=48.0 R2=6093.0 R3=3321.0.

Question 5 Restituer les deux blocs JAVA A COMPLETER dans le code de l'application qui suit:

```

public class ExpMath {
    public static double evaluate(Element e, Element rac){
        double v;
        if (e.getTagName().equals("op"))
        {    // A COMPLETER }

        if (e.getTagName().equals("var"))
        {    // A COMPLETER }
        return v;
    }
    public static void main(String[] args) throws Exception{
        DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
        dbf.setCoalescing(false);
        dbf.setIgnoringElementContentWhitespace(true);
        DocumentBuilder db = dbf.newDocumentBuilder();
        Document document = db.parse(new File("c:/expMath.xml"));
        // Rac est l'élément racine du document
        Element rac = document.getDocumentElement();
        NodeList lesEx = rac.getElementsByTagName("exp");
        for (int i=0; i<lesEx.getLength(); i++) {
            String n = ((Element)lesEx.item(i)).getAttribute("nom");
            Element ex = (Element) lesEx.item(i).getFirstChild();
            System.out.println(n+":="+evaluate(ex, rac));
        }
    }
}

```

4 TestsPsycho.java

4.1 Saisie Assisté de test

Question 6 On considère des tests tels que ceux considérés dans le T.P. précédent avec les spécifications suivantes:

- on ne se fixe pas à priori le nombre d’item de réponses par question
- cependant ce nombre d’item est le même pour chaque question
- le nombre de “caractères” étudiés est au nombre d’item de réponses par question
- les items de réponses d’une question correspondent chacun à un des caractères
- l’ordre des caractères des item de réponses pour une question est aléatoire

Modifier votre DTD en fonction de ces éléments.

Indiquez en quoi, la plupart de ces spécifications n’est pas exprimable dans une DTD.

Afin d’assurer la conformité du document XML correspondant avec ces contraintes, on se propose d’écrire une application de saisie assistée de “test” à intégrer dans le document XML répertoriant les différents tests.

Cette application demandera succesivement à son utilisateur la saisie des informations suivantes:

- Titre du test
- Nombre de Caractères Etudiés
- Identifiant et commentaire associé à chacun des caractères
- Nombre de Question
- Libellé de chaque question
- Libellé et Identifiant du caractère associé à chaque item de réponse à cette question

On s’appuiera sur une instance de document DOM pour effectuer cette mise-à-jour du document XML intégrant les différents tests.

L’interface de saisie seraz rudimentaire (mode texte).

4.2 Application Test

Question 7 Adapter l’application JAVA que vous avez réalisé au premiser semestre de sorte qu’elle prenne comme source de données le document XML ainsi construit, et qu’elle puisse les informations dont elle a besoin dans l’objet DOM correspondant au document XML.

Au terme du test, l’application devra indiquer à l’utilisateur son ”profil psychologique” tel qu’il est révélé par les items de réponses qu’il aura choisi.