

XML et la programmation Java

Objectif

L'objectif de ce TP est de présenter les bases de la programmation Java autour des documents XML : l'analyse de documents avec sax, la modélisation en mémoire avec DOM et l'évaluation de chemin XPath et de feuilles XSLT avec Xalan.

Outils utilisés :

- Le langage Java
- Le parseur XML : Xerces
- Processeur XSLT : Xalan (<http://xml.apache.org/xalan-j/index.html>)

1 L'utilisation de Sax

Dans un premier temps, compiler l'exemple `MySAXApp.java` (qui provient du tutoriel <http://www.saxproject.org/quickstart.html>). Regarder le code source et exécuter ce programme sur le document `animaux.xml`. Expliquer le résultat.

Attention, un parser sax est fourni en standard avec le jdk (xerces). Si vous voulez contrôler la version, il faut compiler et exécuter de l'une des deux façons suivantes :

- `java -Xbootclasspath/p:/usr/share/java/xalan.jar:/usr/share/java/xerces.jar TestAnimal`
- `java -Djava.endorsed.dirs="/usr/local/xalan-j_2_6_0/bin" TestAnimal`

1.1 Un premier exemple

Pour illustrer sax nous allons analyser un document XML qui décrit des animaux (des chiens et des chats) qui sont dans des enclos. Les animaux ont un nom et peuvent être malade. Un exemple de répartition est donné dans le document XML `animaux.xml` (la dtd est donnée dans `animaux.dtd`).

Pour représenter cette organisation, une paquetage java est fournit : `C11-TP5-20060207.jar`. Un exemple d'utilisation est donné dans le programme `TestAnimal.java`. Vous pouvez le compiler puis l'exécuter de la façon suivante :

- `javac -classpath C11-TP5-20060207.jar TestAnimal.java`
- `java -classpath C11-TP5-20060207.jar:. TestAnimal`

2 L'analyse d'un document

En vous inspirant du tutoriel <http://www.saxproject.org/quickstart.html> écrire un programme java qui analyse un document XML conforme à la DTD `animaux.dtd` et qui construit les enclos en mémoire puis affiche le résultat.

3 L'utilisation du modèle DOM

Nous utiliserons l'implantation de Apache : <http://xerces.apache.org/xerces2-j>, les exemples sont disponibles ici : [/usr/local/share/java/dom/xerces-2.7.1/samples](http://usr/local/share/java/dom/xerces-2.7.1/samples).

3.1 La construction en mémoire d'un document XML

L'API DOM offre les constructeur de base pour créer un document XML en mémoire. Examiner, compiler et exécuter le programme Java `ConstructionDom.java`.

En vous inspirant de cet exemple et du programme `TestAnimal.java`, écrire une méthode qui prend en paramètre un `Enclos` (la méthode `getAnimaux()` retourne un objet de type `List<Animal>`) et qui crée un document DOM qui le représentent en mémoire.

3.2 La modélisation en mémoire d'un document XML

Un document DOM peut aussi être créé à partir de l'analyse d'un document XML (avec un parseur sax). Le programme java `DOMFromXML.java` illustre cette utilisation.

Modifier le programme java exemple pour qu'il valide le document XML lors de la création du document DOM. Tester ce programme en modifiant le document XML `animaux.xml` pour qu'il ne soit d'abord plus bien-formé, puis qu'il ne soit plus valide.

3.3 Le parcours d'un document

L'API DOM propose un ensemble de méthode associée à un noeud pour accéder à d'autres noeuds. Le document peut donc être parcouru à partir de la racine.

En plus de cela, trois méthode d'accès et parcours de plus haut niveau sont proposées : `Iterator`, `NameNodeFilter` et `TreeWalker` (cf. `/usr/local/share/java/dom/xerces-2.7.1/samples/dom/traversal`).

En utilisant ces méthodes :

- Compter les nombres d'enclos
- Compter le nombre de chats
- Afficher le document sous la forme d'un arbre

4 Pour aller plus loin

Il est aussi possible d'évaluer des expressions XPath (avec DOM3) et éventuellement d'appliquer une feuille de style XSLT sur une instance DOM pour en créer une autre.

4.1 L'évaluation d'une expression XPath

Le programme `XPathTest.java` vous montre comment créer une expressions XPath et l'évaluer.

Ecrire un programme Java qui répond aux questions du TP sur XPath en affichant les résultats.

4.2 L'application d'une feuille XSLT

Le programme `TestXSLT.java` vous montrer comment appliquer une feuille de style XSLT depuis Java.

Vous pouvez donc maintenant écrire un programme Java qui à partir du document XML qui décrit les notes :

- Charge ce fichier en mémoire sous forme d'un document DOM
- Modifie le document en mémoire (calcul des moyennes, de la meilleure note de chaque matière, ...)
- Applique une feuille de style au document pour produire le SVG ou le XHTML.