

On considère le fichier XML suivant :

```
<etudiants>
  <etudiant nom="Durand" prenom="Emile" />
  <etudiant nom="Dupont" prenom="Micheline" />
</etudiants>
```

1. Ecrire une application GestionEtudiant en utilisant le package org.w3c.dom du jdk de sun qui a les fonctionnalités suivantes :

a. Document chargerXML(String fileName)

On utilisera la classe DocumentBuilderFactory pour fabriquer un parser DocumentBuilder qui vous permettra de charger le document XML. On peut optimiser le parser en réglant les propriétés de la fabrique.

b. void afficherEtudiants(Document listeEtudiants)

Parcours la liste des étudiants et affiche le nom et prénom de chacun.

c. void modifierEtudiant(Element etudiant, String newNom,
String newPrenom)

Modifie un étudiant dont on connaît la référence objet dans l'arbre XML avec un nouveau nom et un nouveau prénom.

d. void ajouterEtudiant(Document listeEtudiants, String newNom,
String newPrenom)

Ajoute un nouvel étudiant en créant un nouveau noeud dans l'arbre XML.

e. void sauverXML(Document listeEtudiants, String fileName)

Sauve l'arbre XML en mémoire vive dans un fichier XML. De la même façon que pour le chargement, on utilisera un TransformerFactory pour fabriquer un Transformer capable de réaliser la sauvegarde. On fixera les propriétés suivantes - définies dans la classe OutputKeys - sur le Transformer :

- ✓ l'encodage
- ✓ l'indentation
- ✓ la méthode (xml)
- ✓ l'ajout de déclaration d'entête XML

f. un main qui charge un fichier donné en 1^{er} argument de l'appli, affiche la liste des étudiants, modifie le 1^{er} étudiant en Dupont Maurice, ajoute l'étudiant Dumoulin Valentine et sauve l'arbre résultat dans le fichier de nom passé en 2^{ème} argument de l'appli.

2. Refaire l'exercice 1 en gérant les espaces de noms :

```
<exo:etudiants xmlns:exo="http://ing2.eisti.fr/BDD-XML-II">
  <exo:etudiant nom="Durand" prenom="Emile" />
  <exo:etudiant nom="Dupont" prenom="Micheline" />
</exo:etudiants>
```

3. Rajouter la validation par rapport à un schéma XSD au chargement du fichier XML :

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="etudiants">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="etudiant" minOccurs="0" maxOccurs="unbounded">
          <xs:complexType>
            <xs:complexContent>
```

```

        <xs:restriction base="xs:anyType">
            <xs:attribute name="nom" type="xs:string"/>
            <xs:attribute name="prenom" type="xs:string"/>
        </xs:restriction>
    </xs:complexContent>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

Il faut faire les réglages suivants sur la fabrique de parser :

- ✓ prise en compte des espaces de nom : non (avec la version de l'exercice 1)
- ✓ validation : oui
- ✓ ajouter l'attribut `JAXP_SCHEMA_LANGUAGE` avec la valeur `W3C_XML_SCHEMA`
- ✓ ajouter l'attribut `JAXP_SCHEMA_SOURCE` avec comme valeur le fichier XSD (de type `File`)

avec les constantes :

- ✓ `JAXP_SCHEMA_LANGUAGE` : "http://java.sun.com/xml/jaxp/properties/schemaLanguage"
- ✓ `W3C_XML_SCHEMA` : "http://www.w3.org/2001/XMLSchema"
- ✓ `JAXP_SCHEMA_SOURCE` : "http://java.sun.com/xml/jaxp/properties/schemaSource"

Pour pouvoir suivre le déroulement de la validation on associe un error handler ; on pourra utiliser la classe `DefaultHandler` ou définir une spécialisation de celle-ci pour s'approprier le traitement d'erreurs.

NB : pour pouvoir utiliser la version avec espace de nom, il faut changer le paramètre dédié du parser et ajouter au schéma l'espace de nom en attribut de la balise principale :

```
targetNamespace="http://ing2.eisti.fr/BDD-XML-II" elementFormDefault="qualified"
```

4. Refaire la feuille de TD avec JDom.