

1. Mise en œuvre des technologies DOM et SAX avec Java

On considère le fichier `etudiant.xml` suivant

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

1.1 Partie DOM

Ecrire une application `GestionEtudiant` en utilisant le package `org.w3x.dom` du `jdk` de `sun` qui a les fonctionnalités suivantes :

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.

void afficherEtudiants(Document listeEtudiants)

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

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.

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

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

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)

1.2 Partie SAX

Rajouter la validation par rapport à un schéma XSD au chargement du fichier XML
Soit le fichier etudiant.xsd suivant

```
<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>
```

- Ecrire une classe XmlValidHandlerManager qui dérive de la classe DefaultHandler et qui implémente uniquement les méthodes de l'interface ErrorHandler
- Ecrire une classe Validator qui contient la méthode main. Cette méthode prend en paramètre le fichier etudiant.xml et le fichier etudiant.xsd et vérifie que le fichier xml respect le modèle décrit dans le fichier xsd.
- Ecrire une classe MySAXParser **extends** DefaultHandler et qui implémente la méthode startElement(...) déclaré dans l'interface ContentHandler qui liste les éléments rencontrés (voir la sortie ci-dessous)

```
Element rencontré : etudiants
Element rencontré : etudiant
Element rencontré : nom
Element rencontré : prenom
```

- Refaire l'exercice en gérant les espaces de nom dans le fichier etudiant.xml ci-dessous

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