

Bonjour,

Voici les « consignes » de votre projet du module « Documents structurés ». Je vous prie de bien vouloir lire attentivement les consignes, afin de mener à bien votre projet.

1/ Votre programme JAVA doit impérativement prendre en argument le répertoire « **projet** » que je vous fournis en pièce jointe dans l'archive « **projet.rar** », contenant 3 fichiers XML, 1 fichier (X)HTML et 2 fichiers TXT :

1. **M457.xml** ;
2. **M674.xml** ;
3. **boitedialog.fxml**;
4. **renault.html** ;
5. **fiches.txt** ;
6. **poeme.txt**.

Vous ne devez pas modifier ces fichiers, sauf s'ils contiennent des **erreurs de syntaxe XML** (erreurs de justesse de forme) que vous devez corriger au préalable avant de commencer à les traiter avec vos codes **JAVA**, qui exploitent l'**API DOM (voir cours 8 et 9)**. La **structure** du répertoire ne doit pas être modifiée ainsi que l'**encodage** des 6 fichiers.

2/ Le but de votre projet consiste à réaliser un programme JAVA, qui permet de transformer ces 6 fichiers dans un autre format.

Les **7 sorties XML** doivent être les suivantes :

1. M674.xml => **sortie1.xml** ;
2. M457.xml => **sortie2.xml** ;
3. renault.html => **renault.xml** ;
4. fiches.txt => **fiches1.xml** ;
5. fiches.txt => **fiches2.xml** ;
6. poeme.txt => **neruda.xml** ;
7. boitedialog.fxml => **javafx.xml**.

Je vous fournis ces 7 sorties attendues en pièce jointe et les deux DTD dans l'archive « **fichiers-de-sortie_et-DTD.rar** ». L'indentation des **7 fichiers XML** que vous allez produire doit être la même que celle des fichiers XML de l'archive « **fichiers-de-sortie_et-DTD.rar** ».

3/ **Trois fichiers (documents) XML obtenus en sortie doivent contenir un lien vers une DTD** :

1. **sortie1.xml => pointe vers la DTD « dom.dtd »** ;
2. **sortie2.xml => pointe vers la DTD « dom.dtd »** ;
3. **neruda.xml => pointe vers la DTD « neruda.dtd »** ;
4. **renault.xml => ne pointe vers aucune DTD** ;
5. **fiches1.xml => ne pointe vers aucune DTD** ;
6. **fiches2.xml => ne pointe vers aucune DTD** ;
7. **javafx.xml => ne pointe vers aucune DTD**.

4/ Le programme JAVA doit être récursif. Il doit donc fonctionner comme le parcours récursif que nous avons vu l'année dernière en « programmation Python » pour lire et parcourir récursivement un répertoire.

Autrement dit, votre sera capable de parcourir n'importe quelle arborescence de répertoires (parcours récursif générique). Le but étant de parcourir chaque niveau de votre répertoire « **projet** » et de traiter tous ses fichiers.

5/ Vous ne devez pas coder une interface graphique. Vous devez juste réaliser un JAR exécutable en ligne (invite) de commandes.

6/ Les méthodes `toString()`, `listFiles()`, `getAbsolutePath()`, `toPath()`, `getPath()`, `getAbsoluteFile()`, `getNodeTypes()`, `getTextContent()`, `setTextContent()`, `newDocument()` **sont interdites dans vos codes JAVA.**

7/ Le package « `java.nio` » **est interdit dans vos codes JAVA.**

8/ Il est possible d'utiliser le langage d'**expressions régulières** dans vos codes JAVA, mais uniquement avec les classes du package de base « `java.lang` ».

9/ **L'interface « `Iterator` »** se trouvant dans le package « `java.util` » **est également interdite dans vos codes JAVA.**

10/ La méthode `getElementsByTagName()`, **ne doit pas recevoir l'étoile * comme argument.**

11/ Les constantes¹ suivantes sont interdites dans vos codes JAVA :

1. `ELEMENT_NODE` ;
2. `ATTRIBUTE_NODE` ;
3. `TEXT_NODE` ;
4. `CDATA_SECTION_NODE` ;
5. `ENTITY_REFERENCE_NODE`,
6. `ENTITY_NODE`;
7. `PROCESSING_INSTRUCTION_NODE` ;
8. `COMMENT_NODE` ;
9. `DOCUMENT_NODE` ;
10. `DOCUMENT_TYPE_NODE` ;
11. `DOCUMENT_FRAGMENT_NODE` ;
12. `NOTATION_NODE`.

12/ **Il est interdit d'importer** le package « `org.w3c.dom.traversal` » et tout ce qu'il contient comme interfaces (`DocumentTraversal`, `NodeFilter`, `NodeIterator` et `TreeWalker`), ainsi que leurs méthodes : `createNodeIterator()`, `createTreeWalker()`, `nextNode()`, `detach()`, `getRoot()`, `previousNode()`, `getWhatToShow()`, `getFilter()`, `getExpandEntityReferences()`.

13/ Vous devez utiliser « **JAXP** » défini dans le package « `javax.xml.parsers` » (voir cours 8 et 9) pour **parser (analyser)** vos documents.

¹ Voir tableau des constantes suivant : <https://www.jmdoudoux.fr/java/dej/chap-dom.htm>

14/ Pour transformer une source en résultat (voir cours 9), vous devez utiliser les classes (« javax.xml.transform.**Transformer** » et « javax.xml.transform.**TransformerFactory** » et les sous-packages (« javax.xml.transform.**stream** » et « javax.xml.transform.**dom**»), qui se trouvent dans le package « javax.xml.**transform** ».

15/ Si le répertoire « **projet** » n'est pas passé en paramètre au JAR, le programme doit impérativement afficher le message d'erreur suivant :

Attention vous avez oublié de spécifier le nom du répertoire à traiter !

16/ Je vous **recommande vivement** de travailler avec JAVA 8 pour éviter tout problème de compatibilité.

17/ Vous devez m'envoyer par mail :

- 1/ un JAR contenant votre programme exécutable en ligne de commandes ;**
- 2/ les résultats de son exécution (fichiers XML obtenus en sortie) ;**
- 3/ les codes sources (classes en .java) de votre programme.**

18/ Le dernier délai d'envoi de votre projet est fixé au vendredi 12 juin à 23h59.

Le non-respect de ces consignes entraînera des pénalités.

Bon travail !