

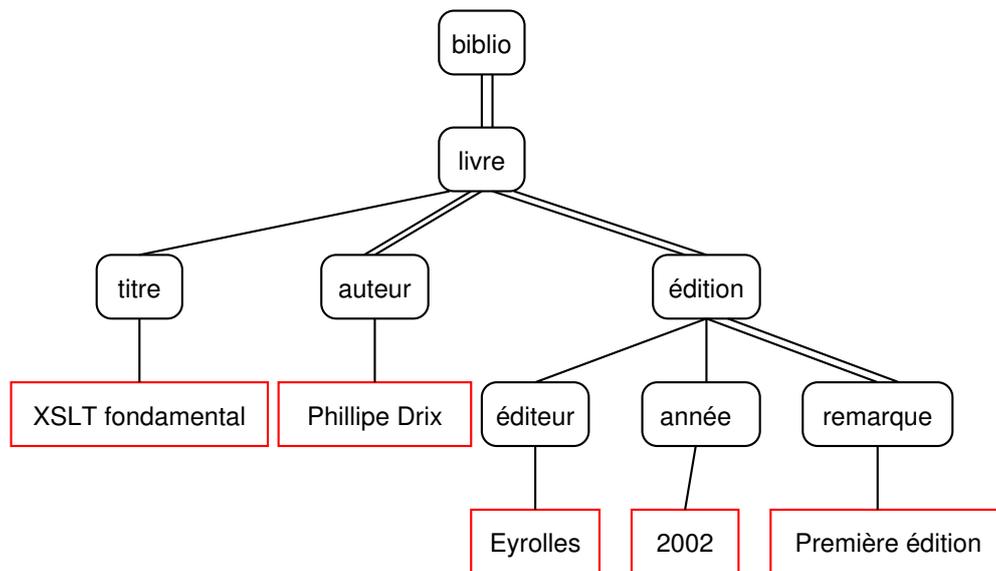
La consultation de documents (supports de cours, livres, etc.) est autorisée. L'interaction avec les autres candidats n'est pas autorisée!

L'examen est constitué d'un seul problème formant une suite de questions. Il est impératif de traiter les deux premières sections avant de passer à la suite. Par contre, les trois dernières sections sont relativement indépendantes les unes des autres.

1 Cadre général et représentation des données

1.1 Énoncé

On se propose dans ce sujet de réaliser une application de gestion de bibliothèque dans laquelle toutes les informations seront représentées en XML. Voici un exemple d'un document XML comportant les éléments principaux de l'application, sous forme d'un arbre :



Dans cette représentation les doubles lignes indiquent que le nœud inférieur peut apparaître plus d'une fois comme fils du nœud supérieur. Par exemple, l'élément `livre` (et donc ses descendants) peut apparaître plus d'une fois comme fils de la racine `biblio`.

Voici le sens à accorder aux éléments proposés dans l'arbre :

- chaque livre de la bibliothèque est représenté par un élément `livre` ;
- le titre du livre est donné par le contenu texte de l'élément `titre` ;
- l'auteur du livre est donné par le contenu texte de l'élément `auteur` (on autorise plusieurs auteurs) ;
- les informations sur les éditions du livre sont regroupées dans des éléments `édition` (on autorise plusieurs éditions) qui proposent l'année d'édition (contenu texte de l'élément `année`), l'éditeur (contenu texte de l'élément `éditeur`) et des remarques sur l'édition ;
- les remarques (contenu texte de l'élément `remarque`) sont en nombre arbitraire (éventuellement aucune) et servent à donner des précisions sur l'édition (par exemple indiquer que l'édition est épuisée, que c'est la troisième, etc.).

1.2 Questions

1. Traduire l'arbre donné en exemple en un fichier XML bien formé.
2. Écrire une DTD pour le modèle de bibliothèque.
3. Écrire un schéma RELAX NG pour le modèle.

2 Gestion des emprunts

2.1 Énoncé

Le format de bibliothèque permet seulement de décrire les livres possédés par la bibliothèque, sans tenir compte ni des quantités, ni des emprunteurs. La représentation des quantités possédés ne pose pas trop de problème. Il suffit en effet d'ajouter à chaque édition le nombre d'exemplaires possédés par la bibliothèque. On peut simplifier les traitements en indiquant aussi le nombre d'exemplaires actuellement empruntés.

La situation est plus complexe pour les emprunteurs. On veut en effet conserver (dans le même fichier) une liste d'emprunteurs, avec pour chacun emprunteur les éléments suivants :

- le nom et le prénom de l'emprunteur
- la liste des livres empruntés
- pour chaque livre emprunté, la date de départ du livre de la bibliothèque

Bien entendu, on veut éviter au maximum la redondance, ce qui implique l'utilisation de références croisées entre les emprunteurs et les livres.

2.2 Questions

1. Proposer (sous forme d'arbre ou de texte) un exemple de sous-arbre `édition` contenant les informations additionnelles suivantes :
 - le nombre d'exemplaires possédés par la bibliothèque
 - le nombre d'exemplaires actuellement empruntés
 - une clé unique pour l'identification de l'éditionIl est vivement conseillé d'utiliser des attributs pour stocker ces informations.
2. Donner les modifications induites par cette nouvelle structure dans la DTD proposée à la section 1.2.
3. Donner un exemple de sous-arbre `emprunteur` (format texte ou dessin) contenant les informations nécessaires à la description d'un emprunteur et de ses emprunts (selon le cahier des charges proposés dans la section précédente). Il est vivement conseillé de ne pas utiliser d'attribut pour représenter les informations, sauf si cela est vraiment indispensable.
4. Donner les modifications induites par cette nouvelle structure dans la DTD de la question 2, en tenant compte des indications suivantes :
 - un fichier XML n'a qu'une seule racine ;
 - une description de bibliothèque doit contenir une liste de livres et une liste d'emprunteurs qu'on aura intérêt à séparer au maximum.

3 Traitements par SAX

3.1 Remarque

On cherche maintenant à produire des programmes de traitement du format de document obtenu dans la section précédente. Dans la présente section, vous devez seulement écrire un `ContentHandler` (en Java, bien sûr), sans vous préoccuper du code nécessaire au chargement du fichier XML et à l'utilisation de votre `ContentHandler` sur ce fichier. De plus, vous pouvez vous passer des `imports`.

3.2 Questions

1. Écrire un `ContentHandler` qui calcule (et affiche dans la méthode `endDocument`) les informations suivantes :
 - le nombre total de livres possédés par la bibliothèque ;

- le nombre total de livres empruntés (selon les deux techniques, c'est-à-dire à partir des informations contenues dans les éditions et avec les informations contenues dans les emprunteurs) ;
- le nombre total d'emprunteurs.

Dessiner l'automate utilisé pour l'analyse du fichier XML.

2. Écrire un `ContentHandler` qui affiche la liste des livres possédés par la bibliothèque en indiquant pour chaque livre les informations suivantes :
 - titre
 - liste des auteurs
 - nombre d'éditions
 - nombre total d'exemplaires

4 Traitements par DOM

4.1 Remarque

Comme dans la section précédente, vous devez écrire seulement ce qui est spécifique à la résolution de la question. Je vous conseille donc d'écrire pour chaque question une ou plusieurs méthodes (et éventuellement vos propres classes) avec une méthode principale de la forme :

```
1 public static void process(Document doc) {  
2     // votre traitement ici  
3 }
```

4.2 Questions

1. Écrire une fonction DOM qui affiche la liste des emprunteurs en indiquant pour chacun d'eux les informations suivantes :
 - le nom et le prénom de l'emprunteur
 - la liste des livres empruntés, avec pour chaque livre :
 - la date d'emprunt
 - le titre
 - la liste des auteurs
 - l'édition empruntée (éditeur, date et liste des remarques)
2. Écrire une fonction DOM qui affiche la liste des livres en indiquant pour chacun d'eux les informations suivantes :
 - le titre et la liste des auteurs
 - le nombre d'exemplaires disponibles (c'est-à-dire non empruntés)
 - la liste des emprunteurs, avec pour chaque emprunteur :
 - la date d'emprunt
 - le nom et le prénom de l'emprunteur

5 XSLT

Écrire un programme XSLT qui transforme un fichier de bibliothèque en une page HTML donnant les informations demandées par les deux questions DOM.