

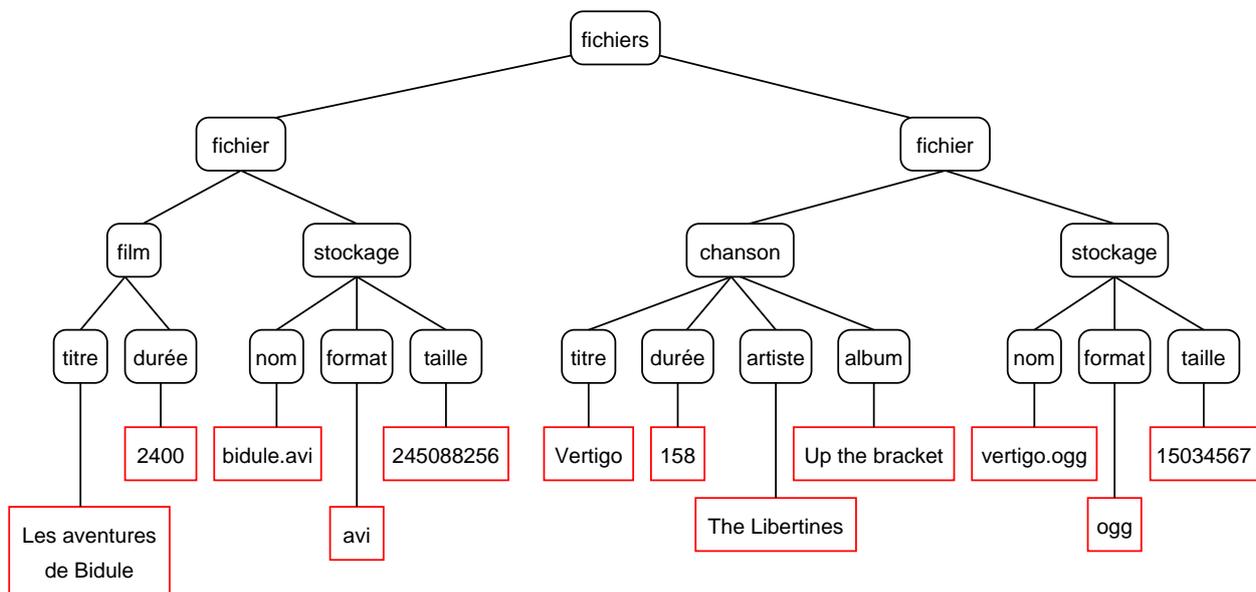
La consultation de documents (supports de cours, livres, etc.) est autorisée. L'interaction avec les autres candidats n'est pas autorisée!

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 partage de fichiers dans laquelle toutes les informations sont représentées sous forme de fichiers XML. Dans un premier temps, on se focalise sur le stockage d'une liste de fichiers à télécharger, dont voici un exemple représenté sous forme d'arbre :



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

- chaque fichier à télécharger est représenté par un élément **fichier**. La liste de fichiers peut éventuellement être vide ;
- on considère deux catégories de fichiers, les films et les chansons :
 - un film est décrit par un élément **film** précisé par deux sous-éléments obligatoires, **titre** (le titre du film) et **durée** (la durée en secondes du film) ;
 - une chanson est décrite par un élément **chanson** précisé par quatre sous-éléments obligatoires, **titre** (le titre de la chanson), **durée** (la durée de la chanson en secondes), **artiste** (l'interprète de la chanson) et **album** (le titre de l'album dont est issue la chanson) ;
- la description informatique de chaque fichier est donnée par le sous-élément obligatoire **stockage**, lui-même précisé par les trois sous-éléments suivants :
 - **nom** donne le nom du fichier ;
 - **format** précise le format du fichier. On considère pour les films les formats **avi** et **mpg**, et pour les chansons les formats **mp3** et **ogg**. Le format ne correspond pas nécessairement à la fin du nom du fichier ;
 - **taille** indique la taille en octets du fichier.

1.2 Questions

1. Traduire l'arbre donné en exemple en un fichier XML bien formé.

2. Écrire une DTD pour le modèle (en respectant au maximum les contraintes indiquées dans la description précédente).
3. Écrire un schéma RELAX NG pour le modèle (même remarque que pour la DTD). Préciser les contraintes que le schéma permet d'introduire en plus par rapport à la DTD (sans utiliser de *datatype library*). Quelles contraintes pourrait-on ajouter en utilisant une *datatype library* ?

2 Sources

2.1 Énoncé

Le format étudié dans la section précédente permet simplement de conserver une liste de fichiers. Pour réaliser un partage, il faut savoir depuis quelles sources on peut télécharger un fichier. Pour ce faire, on veut ajouter au format une liste de sources comportant les informations suivantes pour chaque source :

- adresse ip de la source (4 octets, à représenter séparément les uns des autres) ;
- nom de la source (une chaîne de caractères libre) ;
- liste des fichiers proposés par la source.

Pour éviter la redondance dans le fichier XML, on utilisera des références croisées (ID et IDREF).

2.2 Questions

1. Indiquer les modifications à apporter à la DTD de la section précédente pour ajouter un attribut obligatoire de type ID permettant l'identification de chaque fichier.
2. Même question pour le schéma RELAX NG.
3. On suppose que la liste des fichiers proposés par une source est représentée par des références croisées (des IDREFs). Donner un exemple de liste de sources (avec au moins 2 sources), sous forme d'un arbre (on se contentera de la partie sources du fichier).
4. Donner les modifications à apporter à la DTD de la section précédente pour qu'un document XML contienne une liste de fichiers à télécharger et une liste de sources pour ces fichiers, avec le mécanisme de références croisées étudiés dans les questions précédentes.

3 Traitement par SAX

3.1 Remarque

On cherche maintenant à produire des programmes de traitement du format de document obtenu dans la section précédente (il s'agit donc du format complet, contenant à la fois les fichiers et les sources). 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 fichiers dans le document, avec la répartition entre les films et les chansons ;
 - le nombre total de sources ;
 - le nombre moyen de fichiers partagés par source.

2. Écrire un `ContentHandler` qui calcule (et affiche dans la méthode `endDocument`) les informations suivantes :
 - le volume total en octets des fichiers décrits dans le document (dessiner l’automate associé à la construction de cette information) ;
 - le nombre de fichiers pour chaque format considéré (`ogg`, `avi`, etc.) ;
 - la liste des fichiers dont le nom de termine pas par le format de stockage précédé d’un point (par exemple un fichier au format `avi` de nom `toto.mpeg`).

4 Traitement 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 calcule et affiche le nombre de sources pour chaque fichier, sous forme d’une liste comportant le titre du fichier et le nombre de sources. La fonction affichera de plus le nombre moyen de sources par fichier.
2. Écrire une fonction DOM qui affiche la liste des sources en indiquant pour chacune d’elles les informations suivantes :
 - le nom de la source et son adresse ip ;
 - la liste des titres des fichiers partagés, en commençant par les films ;
 - le volume total des fichiers partagés ;
 - la durée totale (en seconde) des fichiers partagés.
3. Il n’est pas possible d’indiquer dans une DTD (ou un schéma RELAX NG), que les doublons sont interdits dans la liste des fichiers partagés par une source (une source peut donc faire référence plusieurs fois au même fichier). Écrire une fonction DOM qui affiche la liste des sources contenant de tels doublons.

5 XSLT

1. Écrire un programme XSLT qui à partir d’un document au format retenu dans la section 2.2 produit un nouveau fichier XML contenant la liste des sources et pour chaque source la liste des descriptions complètes de fichiers qu’elle partage. L’idée est donc de supprimer la structure de références croisées (on supprime donc la liste globale des fichiers).
2. Écrire un programme XSLT qui réalise la transformation “complémentaire” de celle proposée dans la section précédente : il s’agit de produire un nouveau fichier contenant la liste des fichiers, avec comme information additionnelle pour chaque fichier, la liste des descriptions complètes des sources de ce fichier. Comme dans la question précédente, on supprime la structure de références croisées (et donc la liste globale de sources).
3. Écrire un programme XSLT qui transforme un document au format retenu dans la section 2.2 en une page HTML en conservant la structure de références croisées sous forme de liens internes.