

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

1 Application étudiée

On étudie dans ce problème un format XML permettant de décrire des commandes. Voici un exemple de fichier respectant ce format :

```
_____ commande.xml _____
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <bon>
3   <client>
4     <nom>Doe</nom>
5     <prénom>John</prénom>
6     <adresse>
7       <num>5</num>
8       <voie type="rue">de la gare</voie>
9       <code>75045</code>
10      <ville>Paris</ville>
11      <pays>France</pays>
12    </adresse>
13    <téléphone type="fixe">07 44 44 44 44</téléphone>
14  </client>
15  <commande>
16    <article id="t1223">
17      <nom>Un truc</nom>
18      <quantité>1</quantité>
19      <prix>17.34</prix>
20    </article>
21    <article id="t5543">
22      <nom>Un Bidule</nom>
23      <quantité>2</quantité>
24      <prix>7.2</prix>
25    </article>
26  </commande>
27 </bon>
```

L'exemple est insuffisant pour observer toutes les possibilités du format. On tiendra donc compte des remarques suivantes :

- le client peut avoir deux prénoms ;
- la nature de la voie (attribut **type**) est à choisir parmi diverses possibilités classiques (rue, avenue, place, boulevard, etc.) ;
- la nature du téléphone (attribut **type**) est à choisir parmi diverses possibilités classiques (fixe, portable et fax) ;
- le client peut avoir plusieurs numéros de téléphone ;
- la commande est constituée d'un ou plusieurs articles ;
- l'attribut **id** de l'élément **article** est de type ID.

2 DTD et schéma

2.1 Remarque

Vous pouvez représenter les fichiers XML soit sous forme d'arbres, soit sous forme écrite classique, à votre convenance.

2.2 Questions

1. Écrire une DTD pour le format proposé dans la section 1, en tenant compte de l'exemple et des remarques.
2. Écrire un schéma RELAX NG pour le sous-arbre `client` du format considéré.
3. Expliquer brièvement quelles contraintes additionnelles (spécifiques à l'application envisagée) pourraient être introduites grâce à l'utilisation d'un schéma RELAX NG pour le format considéré.

3 SAX

3.1 Remarque

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 le nombre d'articles différents commandés (attention, ici on ne tient pas compte de la quantité commandée pour chaque article, mais simplement des articles eux-mêmes). Pour le fichier d'exemple, on doit donc compter 2 articles commandés.
2. Écrire un `ContentHandler` qui affiche la liste des articles commandés, sous la forme des noms des articles suivis par la quantité commandée pour chaque article. Pour le fichier d'exemple, on doit donc obtenir :

```
1 Un truc (1)
2 Un Bidule (2)
```

3. Écrire un `ContentHandler` qui calcule et affiche le montant total de la commande, en tenant compte cette fois-ci des quantités commandées. Pour le fichier d'exemple on doit donc obtenir un total de 31,74 euros.

4 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 les coordonnées du client correspondant à une commande, de façon formatée pour une utilisation sur une enveloppe, comme par exemple :

```

1  John Doe
2  5, rue de la gare
3  75045 Paris
4  France

```

2. Écrire une fonction DOM qui calcule et affiche le montant total de la commande, comme pour la question 3 de la section 3.2.

5 Références croisées

5.1 Le problème

On souhaite maintenant combiner plusieurs bons de commande dans un seul fichier, avec comme racine l'élément `archives`. Le problème est que si on se contente de recopier les bons sous `archives`, le fichier obtenu risque d'être très redondant : deux clients différents peuvent avoir commandé des articles semblables et on peut avoir plusieurs commandes d'un même client. On décide de faire les modifications suivantes :

- les commandes d'un même client sont regroupées, c'est-à-dire que l'élément `bon` peut avoir plusieurs fils `commande` ;
- on ajoute à l'élément `commande` un attribut `date` qui précise, comme son nom l'indique, la date de la commande ;
- on remplace les sous-éléments `article` par des sous-éléments `item` de la forme suivante :

```

1  <item ref="t1223" quantité="1"/>

```

l'attribut `ref` est un IDREF qui désigne l'article commandé, alors que l'attribut `quantité` donne la quantité commandée.

- on regroupe les descriptions d'articles (sous-arbre `article`) comme fils d'un élément `catalogue` lui-même premier fils direct de `archives` (les sous-éléments `bon` sont fils directs de `archives` et sont situés après `catalogue`). Dans ce regroupement, les sous-éléments `quantité` sont supprimés.

Voici la traduction de l'exemple de la section 1 dans le nouveau format proposé :

```

new-commande.xml
1  <?xml version="1.0" encoding="ISO-8859-1"?>
2  <archives>
3    <catalogue>
4      <article id="t1223">
5        <nom>Un truc</nom>
6        <prix>17.34</prix>
7      </article>
8      <article id="t5543">
9        <nom>Un Bidule</nom>
10       <prix>7.2</prix>
11     </article>
12   </catalogue>
13   <bon>
14     <client>
15       <nom>Doe</nom>

```

```
16 <prénom>John</prénom>
17 <adresse>
18   <num>5</num>
19   <voie type="rue">de la gare</voie>
20   <code>75045</code>
21   <ville>Paris</ville>
22   <pays>France</pays>
23 </adresse>
24 <téléphone type="fixe">07 44 44 44 44</téléphone>
25 </client>
26 <commande date="14/04/2003">
27   <item ref="t1223" quantité="1"/>
28   <item ref="t5543" quantité="2"/>
29 </commande>
30 </bon>
31 </archives>
```

5.2 Questions

1. Donner les modifications à apporter à la DTD pour obtenir le nouveau format (il n'est pas utile de réécrire ce qui n'est pas modifié).
2. Écrire un programme XSLT qui transforme le nouveau format en l'ancien (comme l'ancien format ne permet pas de représenter plusieurs bons de commande, on regroupera les éléments **bon** sous un élément **commandes**).
3. Écrire une fonction DOM qui affiche la liste des clients (le nom et le prénom pour chaque client), avec le montant total des achats de chacun d'eux (en tenant compte, bien entendu, de la quantité commandée pour chaque article). On rappelle qu'un client peut avoir passé plusieurs commandes.