

Documents autorisés : transparents de cours seulement

Vous ne vous préoccupez pas dans cet examen de la gestion des exceptions.

Lecture de noms avec un analyseur SAX

Un logiciel d'apprentissage de vocabulaire stocke ses données dans des documents XML. Ces documents contiennent des mots de vocabulaire dans différents langages, en particulier des verbes sous la forme :

```
<verbe>
<niveau>4</niveau>
<mot lang="en">to eat</mot>
<mot lang="fr">manger</mot>
...
</verbe>
```

On veut récupérer les traductions françaises de tous les verbes listés dans un tel document, c'est-à-dire les contenus des éléments `<mot lang="fr">` des éléments `<verbe>`, sachant que les traductions d'un verbe sont dans un ordre quelconque.

Écrivez la classe `VerbesFrancaisHandler`, *gestionnaire de contenu SAX* qui après avoir été utilisé par un analyseur SAX aura rangé dans un tableau de chaînes de caractères (`ArrayList<String>`) toutes les traductions françaises des verbes du document lu (tableau renvoyé par la méthode `getVerbes()`). Vous ferez attention à ce qu'un objet de type `VerbesFrancaisHandler` puisse être utilisé plusieurs fois.

Changement de format dans le DOM

Vous supposerez dans cette partie que les DOM ne contiennent pas de nœuds texte blanc d'indentation.

Des logiciels d'apprentissage de vocabulaire stockent leurs données dans des documents XML. Les mots de vocabulaire sont contenus dans des éléments `<nom>`, `<verbe>`, `<adjectifs>`... Dans un premier logiciel, le niveau d'apprentissage est stocké dans un attribut `niveau` de ces éléments, alors que dans un deuxième logiciel le niveau est stocké dans un élément `<niveau>` premier enfant de ces éléments. On veut convertir les mots du format 1 (attribut `niveau`) au format 2 (élément `<niveau>`), soit par exemple pour un verbe :

```
Format 1                               Format 2
<verbe niveau="4">                    <verbe>
<mot lang="fr">manger</mot>           <niveau>4</niveau>
...                                     <mot lang="fr">manger</mot>
</verbe>                               ...
                                         </verbe>
```

On veut faire cette transformation directement dans le DOM représentant de tels documents XML.

Écrivez la classe `ModificateurNiveau` dont la méthode `changerNiveau(elem)` change les attributs `niveau` des éléments de l'arbre dont l'élément `elem` est la racine en éléments `<niveau>` (on suppose qu'un élément avec un attribut `niveau` ne contient pas lui-même d'éléments avec un attribut `niveau`), et dont la méthode `changerNiveau(doc)` change de la même façon les attributs `niveau` des éléments du document `doc`.

Un autre logiciel stocke les mots dans un fichier XML de la façon suivante : l'élément `<mots>` contient l'élément `<appris>` qui contient les mots déjà appris au format 2 puis l'élément `<encours>` qui contient les mots `<nom>`, `<verbe>`... en cours d'apprentissage au format 1 :

```
<mots>
<appris>
<!-- mots déjà appris au format 2 -->
...
</appris>
<encours>
<!-- mots en cours d'apprentissage au format 1 -->
...
</encours>
</mots>
```

Quand un mot en cours d'apprentissage devient appris, il faut le changer d'endroit et de format dans le document XML. Soit la classe `DOMVocabulaire` qui contient déjà la méthode `estMotFR(motfr, elem)` qui renvoie vrai si l'élément `elem` contient le mot français `motfr`.

Écrivez la méthode `motDevientAppris(motfr, doc)` (de cette même classe) qui transforme dans le document `doc` le mot français `motfr` en cours d'apprentissage en mot appris : il faut localiser l'élément contenant le mot français parmi les mots en cours d'apprentissage et s'il existe le transformer au format 2 et le déplacer parmi les mots déjà appris.