

Cours de Java avancé - Projet d'évaluation - Gestion de mp3

1 Gestion des mp3 : principe

Le projet consiste à créer une application permettant de gérer un ensemble de fichiers mp3. Pour réaliser cette application il faut tout d'abord réaliser des composants génériques, puis les assembler pour obtenir l'application.

2 Définition des composants - Partie obligatoire

Les composants sont répartis en deux familles :

- Trois composants "modèle" permettant de décrire la structure des données :
 - un composant `Entite` représentant une collection d'objets de type `Champ`.
 - un composant `Champ` représentant un nombre, une chaîne de caractères, un composant de type `Liste`, ou une référence à un composant de type `Entite`.
 - un composant `Liste` représentant une liste d'objets de type `Entite`.

Ces composants pourront être sauvegardés et lus à partir d'un fichier XML.

- Les composants "vue". On trouvera en particulier :
 - un composant affichant un composant modèle `Liste` dans un `javax.swing.JList`.
 - un composant affichant un composant modèle `Entite` dans un `javax.swing.JTable`.

Le projet doit comporter au moins les composants précédents. Une solution devra être trouvée pour pouvoir éditer les composants `Entite`, `Champ` ou `Liste` à partir des composants `vue`.

3 Utilisation des composants - Partie obligatoire

Le but est d'assembler les composants développés dans la section précédente pour créer une application Java permettant de gérer une collection de fichiers mp3.

Voici la description Xml d'une collection :

```
<?xml version="1.0" encoding="UTF-8"?>
<mp3collection>
  <touslestypes>
    <typemusique><id>1</id><nom>jazz</nom></typemusique>
    <typemusique><id>2</id><nom>rock</nom></typemusique>
    <typemusique><id>3</id><nom>pour la voiture</nom></typemusique>
  </touslestypes>
  <collection>
    <chanson>
      <id>1</id>
      <artiste>Michael Bubl  </artiste>
      <titre>It's time</titre>
      <fichier>c:/Documents and setting/fab/Ma musique/mick.mp3</fichier>
      <listetype>
        <type>1</type>
        <type>2</type>
        <type>3</type>
      </listetype>
    </chanson>
    <chanson>
      <id>2</id>
      <artiste>Madonna</artiste>
```

```

<titre>Confessions on a Dance Floor</titre>
<fichier>c:/Documents and setting/fab/Ma musique/mado.mp3</fichier>
<listetype>
  <type>2</type>
  <type>3</type>
</listetype>
</chanson>
</collection>
</mp3collection>

```

On remarque qu'une même chanson peut avoir plusieurs type d'écoute : ainsi la chanson de Madonna (identifiant 2) a les types d'écoute 2 (rock) et 3 (pour la voiture). Comme dans tous fichiers XML les \ sont remplacés par des /

On y trouve quatre types de composants `Entite` : le type de musique, la collection, la chanson, la liste des types.

Voici deux composants utilisés pour décrire un type de musique : un identifiant (balise `id`), le nom. Une collection possède un champ qui est une liste de chansons. Une chanson possède 5 champs :

- un identifiant de la chanson,
- le nom de l'artiste,
- le titre de la chanson,
- le nom du fichier mp3,
- un objet de type `Entite` qui est une liste de référence à des types de musiques.

à partir de la description Xml précédente les composants doivent pouvoir afficher le contenu de la collection.

4 Pour aller plus loin - Partie optionnelle

- On peut procéder à un assemblage automatique des composants. Par exemple pour l'exemple précédent les composants seront automatiquement créés grâce à la description XML ci-dessous :

```

<?xml version="1.0" encoding="UTF-8"?>
<assemblage>
  <!-- définition d'un type de musique -->
  <objet nom="typemusique">
    <!-- donne le nom du type de musique -->
    <champ nom="id" type="nombre"/>
    <champ nom="nom" type="textuel"/>
  </objet>
  <!-- référence à un type de musique -->
  <objet nom="typemusique_ref">
    <!-- donne le nom du type de musique -->
    <champ nom="id" type="reference" objet="typemusique"/>
  </objet>
  <!-- définition d'une chanson -->
  <objet nom="chanson">
    <!-- donne les attributs de la chanson -->
    <champ nom="id" type="nombre"/>
    <champ nom="artiste" type="textuel"/>
    <champ nom="titre" type="textuel"/>
    <champ nom="fichier" type="textuel"/>
    <liste nom="listetype" objet="typemusique_ref"/>
  </objet>
  <!-- définition d'une collection de mp3-->
  <objet nom="mp3collection">
    <liste nom="touslestypes" objet="typemusique"/>
    <liste nom="collection" objet="chanson"/>
  </objet>

```

```
</objet>  
</assemblage>
```

- Les chansons peuvent comporter d'autres champs comme l'appartenance à un album, à une playliste.
- On s'est seulement attaché aux parties modèle et vue. Le projet peut également inclure un aspect contrôleur, en définissant par exemple la notion d'administrateur (qui possède tous les droits pour modifier les données) et d'utilisateur pour lesquels seulement une lecture est possible.
- On peut également créer en plus des composants vue générant du HTML et non des objets JList ou JTable. On peut en plus les intégrer à des pages JSP pour faire un site web dynamique.