Concepts objet et Java

2017-18

Projet noté

L'objet de ce projet noté est de développer une application simple de consultation et d'analyse d'un programme de télévision en langage Java. Il s'agit d'un projet à réaliser individuellement et à rendre selon les modalités indiquées à la fin de ce document.

I Description générale

Un programme de télévision donne pour un certain nombre de jours consécutifs la programmation des émissions pour un certain nombre de chaînes [1]. Chaque chaîne rend disponible sa programmation plusieurs jours à l'avance, ce qui permet notamment d'effectuer certains contrôles ainsi que de produire certaines statistiques.

Dans ce sujet, on s'intéressera au programme des chaînes de la TNT française dans un scénario qui pourrait concerner le Conseil Supérieur de l'Audiovisuel (CSA). Un seul fichier contiendra des informations sous forme semi-structurée pour l'ensemble des chaînes pour un certain nombre de jours sur lesquels porteront les analyses. Les informations pertinentes seront chargées à partir de ce fichier pour créer des objets pertinents qui seront organisés dans des collections appropriées permettant d'effectuer les analyses demandées. L'interface sera en mode texte (ou optionnellement en mode graphique).

II Description détaillée

a) Données en entrée

Les données sont disponibles sur le site : `http://xmltv.fr` [2]. Elles sont fournies dans un langage XML [3]. Les deux types d'éléments importants pour ce sujet concernent les **chaînes** (`channel`) [4] et les **émissions** (`programme`), illustrés ci-dessous :

```
<channel id="C111.api.telerama.fr">
  <display-name>Arte</display-name>
```

1. On parlera ici de *programme* pour dénoter l'ensemble des *émissions* programmées pour les chaînes considérées. On fera plus loin référence à l'utilisation du terme anglais *programme* pour faire référence à une *émission*.
2. La génération du fichier est rendue possible par l'utilisation des API Télérama : `https://api.telerama.fr`
3. Voir le schéma sous forme de DTD dans les fichiers fournis ; la validation du fichier XML selon cette grammaire DTD n'est toutefois par attendue ici.
4. La seule information utile ici est la correspondance unique entre l'attribut `id` d'un élément `channel` et la valeur texte de l'élément fils `display-name`.

```
    <icon src="http://television.telerama.fr/sites/tr_master/files/sheet_media/tv/500x500/
    11.png" />
  </channel>


  <programme start="20180510003000 +0200" stop="20180510023000 +0200" showview=""
  channel="C444.api.telerama.fr">
    <title>Empire State</title>
    <desc lang="fr">Dans les années 80, dans le quartier du Queens à New York. Chris
     Potamitis travaille au sein d'une compagnie de transport de fonds. Après avoir
      mentionné les failles de la société à Eddie, son ami d'enfance, le jeune homme
       se retrouve bien malgré lui au coeur d'un braquage hors normes...</desc>
    <credits>
      <director>Dito Montiel</director>
      <actor>Liam Hemsworth (Chris Potamitis)</actor>
      <actor>Michael Angarano (Eddie)</actor>
      <actor>Dwayne Johnson (James Ransome)</actor>
      <actor>Paul Ben-Victor (Tommy)</actor>
      <actor>Jerry Ferrara (Jimmy)</actor>
      <actor>Greg Vrotsos (Mike Dimitriu)</actor>
      <actor>Michael Rispoli (Tony)</actor>
      <actor>Emma Roberts (Nancy Michaelides)</actor>
    </credits>
    <category lang="fr">téléfilm policier</category>
    <length units="hours">2</length>
    <icon src="http://television.telerama.fr/sites/tr_master/files/sheet_media/media/
    169_EMI_679785.jpg" />
    <country>américain</country>
    <video>
      <aspect>16:9</aspect>
    </video>
    <audio>
      <stereo>stereo</stereo>
    </audio>
    <previously-shown />
    <rating system="CSA">
      <value>-10</value>
      <icon src="http://upload.wikimedia.org/wikipedia/commons/thumb/b/bf/Moins10.svg/
      200px-Moins10.svg.png" />
    </rating>
  </programme>
```

Plusieurs approches sont possibles pour analyser des fichiers XML, et Java propose en standard plusieurs implémentations de parseurs depuis sa version 6.0 [5]. L'analyse du fichier XML pourra se faire avec une approche efficace pour la lecture et peu gourmande en mémoire, StAX (Streaming Api for XML) avec sa variante de type curseur (voir des détails et un exemple sur : `https://www.jmdoudoux.fr/java/dej/chap-stax.htm#stax-4`).

5. De nombreuses références existent pour les traitements XML en Java ; voir par exemple : `https://www.jmdoudoux.fr/java/dej/chap-xml.htm`

b) Modélisation

Des classes spécifiques représenteront chaque type d'émission [6] organisées dans une hiérarchie commune adaptée permettant d'organiser les émissions de différents types dans de mêmes collections. Une émission pouvant être rediffusée, il faudra distinguer les *émissions* elles-mêmes des *émissions programmées* (i.e. pour une chaîne et un horaire donné) et donc ne disposer que d'une seule copie en mémoire des premières en cas de diffusions multiples dans la période. Chaque type d'émission disposera d'un affichage court adapté (pour être utilisé dans une liste dont les entrées sont sélectionnables) ainsi que d'un affichage long adapté (la fiche de l'émission).

c) Fonctionnalités de l'application

Une fois les données chargées, l'application doit permettre de :

1. Consulter la liste des chaînes.
2. Consulter la liste des jours disposant de programmes télé.
3. Consulter la programmation d'une chaîne pour un jour donné.
4. Consulter la fiche d'une émission (avec un affichage adapté à son type).
5. Consulter la liste des émissions qui seront en cours de diffusion à un moment donné.
6. Consulter la liste des films concernant un réalisateur ou un acteur.

L'application doit également permettre d'effectuer les analyses suivantes :

1. La liste des acteurs ordonnée par leur nombre d'apparitions dans des films diffusés.
2. Le nombre d'émissions de chaque type (ex. film, documentaire, sport, cf. `category`) par jour et sur la période.
3. Le nombre d'émissions par catégorie CSA (ex. `<rating system="CSA"><value>Tout public </value></rating>`) par chaîne.
4. La liste des chaînes par ancienneté moyenne décroissante des films diffusés (cf. éléments `date`) [7].

Finalement, l'application devra implémenter l'une des fonctionnalités ci-dessous :

1. Production d'un rapport au format XHTML donnant une synthèse des analyses permises par l'application.
2. Fonctionnement dans une interface graphique simple.
3. Recherche par mots-clés sur le texte de description des émissions (éléments `desc`).

III Rendu attendu

Ce travail est à réaliser et à rendre **individuellement**. Le programme sera écrit en langage Java en mettant en œuvre les principes étudiés lors du semestre. Une attention particulière devra être portée au modèle objet proposé, à l'encapsulation des données, à la gestion des erreurs, et à la documentation du

6. Il ne sera pas nécessaire de distinguer par sous-type (ex. valeurs *téléfilm biographique*, *téléfilm catastrophe*, *téléfilm d'action* pour l'élément `category`), ce qui pourra nécessiter de *parser* ce champ texte pour ne reconnaître que la partie utile (ex. *téléfilm*).

7. Il est possible qu'une chaîne ne diffuse aucun film ; on pourra en outre considérer une valeur minimale de films diffusés pour inclure une chaîne à cette analyse.

code au moyen d'annotations JavaDoc [8].

Le rendu sera livré sous une archive JAR exécutable au nom de l'étudiant qui contiendra les répertoires suivants :

— `src` : l'ensemble des sources du projet.
— `bin` : l'ensemble des classes compilées du projet.
— `doc` : l'arborescence JavaDoc du projet.

Chaque archive sera à envoyer par email pour le lundi **4 juin 2018** [9] à son chargé de TP respectif (Lydia Ould Ouali (`ouldouali@limsi.fr`) ou Quentin Bouillaguet (`quentin.bouillaguet@cea.fr`)) avec comme titre d'email :

```
[Polytech > Et3] rendu projet Java PRENOM_ETUDIANT NOM_ETUDIANT
```

8. En particulier, l'ensemble des choix particuliers faits devront être clairement décrits à l'emplacement approprié, et une documentation principale devra se trouver à la racine du projet.

9. **Tout jour de retard entraînera une perte de 3 points sur la note finale ; il est nécessaire d'avoir obtenu un accusé de réception par email afin de pouvoir considérer son travail comme rendu.**