

# Développement Logiciel, 2013

## TP3 Entrées/sorties

**Préambule** : Récupérez les sources à l'adresse suivante :

`http://www.arlem.info/TP3.zip`

**Très important** : Ce TP est noté. Vous devrez envoyer par e-mail les sources du TP dans une archive à `roza.lemdani@u-psud.fr` avant le **dimanche 10 février 2013 à 23h59**. Il s'agit d'un travail personnel à effectuer en monôme, les binômes sont donc interdits.

**Important** : Pensez à bien indenter votre code (Eclipse se charge de le faire pour vous grâce au raccourci clavier Ctrl+i), à le commenter et à respecter les noms des méthodes qui vous sont imposés. Des points supplémentaires sont attribués aux codes clairs respectant ces conditions.

On se propose de créer un gestionnaire de séries TV. Un gestionnaire de séries TV permet de savoir où on en est dans les épisodes d'une série. En effet, il offre la possibilité d'entrer tous les épisodes d'une série que l'on veut suivre, marquer des épisodes comme vus et afficher la liste des épisodes non vus.

Un **gestionnaire** est composé d'un *ensemble de séries*. Une **série** a un *titre*, un *nombre d'épisodes vus* et contient un *ensemble d'épisodes*. Tous les **épisodes** ont un *titre*, une *date de diffusion* (afin de simplifier, nous allons utiliser la classe String pour représenter les dates) et un état spécifiant si l'épisode a été *vu* ou pas. Il y a deux types d'épisodes : les **épisodes standards** qui ont un numéro de *saison* et un numéro d'*episode* et les **épisodes spéciaux**.

La classe abstraite **Episode** vous a été fournie. Il vous est interdit de la modifier ou d'y ajouter quoi que ce soit.

**Question 1:** Créez les classes **EpisodeStandard**, **EpisodeSpecial**, **Serie** et **Gestionnaire** du package **gestionnaire**, ainsi que leurs constructeurs et leur méthode public **String toString()** telle que :

- l'affichage d'une série est sous la forme : **titre serie (X episodes, Y non vus)**, où X est le nombre d'épisodes de la série et Y le nombre d'épisodes non vus. Par exemple :  
Doctor Who (133 episodes, 127 non vus)
- l'affichage d'un épisode standard est sous la forme : **SxxEyy titre episode (date de diffusion : date)**, où xx est le numéro de saison sur **au moins** deux caractères et yy est le numéro de l'épisode formaté sur **au moins** deux caractères également. Par exemple :  
S05E10 "Vincent and the Doctor" (date de diffusion : 05/Jun/10)
- l'affichage d'un épisode spécial est sous la forme : **(SPECIAL) titre episode (date de diffusion : date)**. Par exemple :  
(SPECIAL) "Serenity" (date de diffusion : 20/Dec/05)
- l'affichage du gestionnaire est sous la forme : **Z. titre serie (X episodes, Y non vus)**, où Z est l'index de parcours du gestionnaire allant de 1 au nombre de séries se trouvant dans le gestionnaire. Par exemple :
  1. Doctor Who (133 episodes, 127 non vus)
  2. Firefly (16 episodes, 16 non vus)

**Question 2:**

1. Ajoutez à la classe Serie les méthodes **private boolean ajouterEpisode (String title, String date, int season, int episode)** qui ajoute un épisode standard à la série et **private boolean ajouterEpisode (String title, String date)** qui ajoute un épisode spécial à la série. Les deux méthodes renvoient vrai si l'ajout a été effectué et faux sinon.
2. Le site <http://epguides.com/> propose des descriptifs de séries au format CSV (comma-separated values). Chaque ligne d'un fichier correspond à des données sur

un épisode suivant un même format :

```
number,season,episode,production code,airdate,title,special?,tvrage
```

Ce qui nous intéresse ici, c'est `season`, `episode`, `airdate`, `title`, `special?`, où `special?` a pour valeur `y` s'il s'agit d'un épisode spécial et `n` s'il s'agit d'un épisode standard. `title` représente le titre de l'épisode et `airdate` sa date de diffusion (pas forcément renseignée, mais ce n'est pas grave). `season` représente le numéro de saison d'un épisode standard et `episode` son numéro d'épisode.

Par exemple, des épisodes standards auront la forme :

```
62,5,10,"",05/Jun/10,"Vincent and the Doctor",n,"http://www.tvrage..."
```

```
63,5,11,"",12/Jun/10,"The Lodger",n,"http://www.tvrage.com/DoctorW..."
```

```
64,5,12,"",19/Jun/10,"The Pandorica Opens (1)",n,"http://www.tvrag..."
```

```
65,5,13,"",26/Jun/10,"The Big Bang (2)",n,"http://www.tvrage.com/D..."
```

Un épisode spécial ressemblera à :

```
,"1",,,20/Dec/05,"Serenity",y,"http://www.tvrage.com/Firefly/episo..."
```

Ajoutez à la classe `Serie` la méthode `private boolean ajouterEpisode (String ligne)` qui prend en paramètre une ligne au format décrit précédemment, la parse et fait appel à une des deux autres méthodes du même nom selon qu'il s'agit d'un épisode spécial ou standard. Vous pouvez utiliser la méthode `split` de la classe `String` qui permet de séparer une chaîne de caractères en plusieurs éléments suivant un motif (une virgule par exemple).

3. Maintenant que vous savez convertir une ligne du fichier CSV en un épisode, vous allez parcourir tout le fichier CSV, grâce à la classe `FileReader` qui permet de lire un fichier, et convertir chaque ligne en un épisode puis ajouter ce dernier à la liste des épisodes de la série. La méthode à créer se trouvera dans la classe `Serie` et aura pour signature `public void fileToShow (String filename)`. Pensez à ignorer la première ligne du fichier qui contient le format des lignes mais pas d'épisode. Deux fichiers CSV vous ont été fournis dans l'archive.
4. Créez la méthode `public boolean ajouterSerie (String title, String filename)` dans la classe `Gestionnaire`, qui ajoute à la liste des séries une série de titre `title` décrite dans le fichier CSV `filename`.

**Question 3:** À ce niveau, vous pouvez ajouter des épisodes à une série et ajouter des séries au gestionnaire. Il est temps de penser à sauvegarder votre gestionnaire pour pouvoir le charger ultérieurement.

1. Créez une classe `EntreesSorties` dans le package `gestionnaire.run`.
2. Ajoutez à la classe `EntreesSorties` la méthode `public static String lireClavier()` qui renvoie la ligne lue sur l'entrée standard. Pour ceci vous pouvez utiliser `System.in` qui est un `InputStream`.

3. Ajoutez à la classe `EntreesSorties` la méthode `public static void sauvegarderFichier (Object contenu)` qui demande à l'utilisateur le chemin du fichier dans lequel enregistrer l'objet passé en paramètre (qui sera en l'occurrence un objet de type `Gestionnaire`).
4. Ajoutez à la classe `EntreesSorties` la méthode `public static Object chargerFichier ()` qui demande à l'utilisateur le chemin du fichier à lire et renvoie l'objet qui y était sauvegardé.

**Question 4:** La classe `Main` et la classe abstraite `Menu` vous ont été fournies.

Le menu consiste en une boucle proposant de :

- 1 : Charger un gestionnaire
- 2 : Afficher les series
- 3 : Ajouter une serie
- 4 : Afficher les episodes non vus
- 5 : Marquer des episodes comme vus
- 6 : Sauvegarder le gestionnaire
- 0 : Terminer

L'utilisateur doit sélectionner le chiffre correspondant à l'action qu'il veut effectuer.

1. Ajoutez à la classe `EntreesSorties` la méthode `public static int lireEntier(int min, int max)` qui demande à l'utilisateur d'entrer un nombre tant qu'il ne sera pas correct et compris entre deux valeurs `min` et `max`.
2. Ajoutez à la classe `EntreesSorties` la méthode `public static ArrayList<Integer> lireEntiers(int min, int max)` qui demande à l'utilisateur d'entrer des nombres séparés par des espaces, tant que chacun d'entre eux ne sera pas correct et compris entre deux valeurs `min` et `max`. Cette méthode renvoie la liste des entiers entrés.
3. Chaque action est associée à une méthode abstraite. Créez la classe `MenuGestionnaire` qui hérite de `Menu`. Implémentez les méthodes correspondant aux actions tel que :
  - `protected void chargerGestionnaire()` : charge un ancien gestionnaire sauvegardé.
  - `protected void afficherSeries()` : affiche la liste des séries du gestionnaire.
  - `protected void ajouterSerie()` : demande à l'utilisateur le nom de la série à ajouter et le chemin du fichier CSV contenant les informations sur les épisodes et effectue l'ajout de la série et des épisodes.
  - `protected void afficherEpisodesNonVus()` : affiche pour chaque série ayant des épisodes non vus, son titre suivi de la liste numérotée des épisodes non vus.
  - `protected void marquerEpisodeVu()` : affiche une liste numérotée des séries ayant des épisodes non vus et demande à l'utilisateur de sélectionner la série dont il veut marquer des épisodes comme vus. La liste numérotée des épisodes non vus de cette série est ensuite affichée et il est demandé à l'utilisateur d'entrer les numéros, un ou plusieurs séparés par un espace, des épisodes à marquer comme vus.
  - `protected void sauvegarderGestionnaire()` : sauvegarde le gestionnaire courant dans un chemin de fichier choisi par l'utilisateur.