

TP 3

Collections

Avant de commencer les exercices ci-dessous, assurez-vous d'avoir fait ceux présents dans les diapos du CM.

Comme toujours, le but des tp est de mettre en pratique les notions que vous avez vues en cours. N'hésitez pas à essayer certaines choses quitte à devoir recommencer, c'est en se trompant que l'on apprend.

Exercice 1 : Chargement des données

Téléchargez sur l'ENT l'archive data.zip et décompressez-la.

Vous obtenez différents fichiers que vous devez alors « parser » afin de pouvoir représenter aux mieux les informations à l'aide des classes adéquates que vous devez créer. Le but de cet exercice est de trouver la représentation permettant d'éviter la duplication d'objets. Par exemple, deux utilisateurs peuvent consulter la même musique et il n'est pas très judicieux de créer une liste (ArrayList par exemple) de musiques consultées pour chaque utilisateur. En effet, en procédant de la sorte, deux objets Musique différents seront créés pour chaque utilisateur. Prenez le temps de réfléchir à votre architecture. Un fichier peut donner lieu à la création de plusieurs classes et il est possible que certaines informations soient à recouper entre les différents fichiers.

Pensez à redéfinir les méthodes hashCode et Equals si nécessaire.

Exercice 2 : Exploitation des données

En utilisant les classes précédentes, répondez aux questions suivantes (en utilisant le plus possible les méthodes des collections que vous utilisez) :

1. Quel est le nombre :
 - a. D'utilisateurs
 - b. De musiques
 - c. D'artistes
2. Combien en moyenne, un utilisateur consulte-t-il :
 - a. D'artistes uniques
 - b. De musiques uniques
 - c. De musiques en tout
3. Pour un utilisateur, créez les méthodes permettant de connaître :
 - a. La musique la plus écoutée
 - b. L'artiste le plus écouté
 - c. Le top 10 des musiques
 - d. Le top 10 des artistes
4. Pour une consultation, créez la méthode permettant :
 - a. D'afficher la date et l'heure de la consultation
 - b. De redéfinir la méthode toString en retournant les informations principales, utilisez pour cela les méthodes toString d'un artiste et d'une musique que vous redéfinirez.

5. Dans la classe utilisateur, créez une méthode permettant de générer dans un **fichier** la liste des musiques qu'il a écoutées. Une même musique ne doit pas apparaître plusieurs fois, par contre on doit savoir combien de fois elle a été écoutée par cet utilisateur.
 - a. Réalisez cette méthode
 - b. Surchargez cette méthode et pour lui passer en paramètre un String pouvant prendre la valeur "croissant" ou "décroissant" qui triera alors cette liste de musique selon l'ordre alphabétique au niveau du nom de la musique. (Utilisez un comparator).

Exercice 3

Une fois que les données sont chargées, sauvegardez-les en utilisant la sérialisation. Vérifiez que vous arrivez bien à recharger l'ensemble de vos données.

Exercice 4 Have fun

Ajoutez des méthodes supplémentaires vous permettant d'explorer les données

Par exemple, rechercher une musique sur YouTube à partir de son titre et son artiste en utilisant votre navigateur par défaut. Vous ne savez pas comment faire ?

lancer votre navigateur à partir d'une url :

<https://www.codeproject.com/Questions/398241/how-to-open-url-in-java>

<https://stackoverflow.com/questions/5226212/how-to-open-the-default-webbrowser-using-java>

Reste à trouver maintenant comment faire pour effectuer une recherche sur youtube avec les paramètres que vous désirez (ne recherchez pas tout de suite la solution sur un forum).