IFT1176 – TP1 – Été 2020

IFT1176 – TRAVAIL PRATIQUE #1 – 19 mai 2020

# Jeux vidéo

Par Michel Reid

**Travail individuel:** Le travail en équipe est interdit.

**Remise :** via Studium, au plus tard : **lundi 15 juin, 22h30, ( -10 points/60 par jour de retard**)

**Conseil:** N'attendez pas la semaine avant la remise pour commencer, vous n'aurez pas le temps.

**But :** Ce TP a pour but de vous faire une révision générale des notions apprises dans un cours de niveau -2- (IFT1170) ainsi que l’utilisation des différentes collections. Ce devoir compte pour 18% de la note globale du cours (sous réserve d’atteindre le seuil).

**Énoncé :** On se propose de gérer une banque de données composée d’un ensemble d’éléments représentant des jeux vidéo.

Un jeu est représenté par :

* le nom du fabricant,
* le titre du jeu,
* la cote (groupe d’âge pour lequel le jeu est recommandé)
* et par la liste des consoles vidéo pour lesquelles le jeu est disponible.

On vous demande d’écrire une classe **Bdd** et d’y définir les services (méthodes) suivants, déclarés dans l’interface **TestInterface (Bdd** doit donc implémenter **TestInterface)** :

* **saveBdd**: permet de sauvegarder la banque de données dans un fichier, dont le nom est passé en paramètre. Le format du fichier est le même que pour la lecture d’un fichier.
* **addBdd**: permet d'ajouter des informations à la banque de données, regroupées dans un fichier, dont le nom est passé en paramètre.
* **loadBdd**: permet de charger la banque de données à partir d'un fichier, dont le nom est passé en paramètre. Si une banque de données est déjà chargée, celle-ci est remplacée. (Vous pouvez faire appel à **addJeu** et/ou **addBdd**)
* **addJeu**: ajoute un jeu (un objet), passée en paramètre, dans la banque de données. Remarques :
  + Il faut tenir compte de la présence du fabricant, ou non, dans la **Map**.
  + Si le jeu est déjà présent pour ce fabricant, il faut s’assurer d’ajouter toutes les consoles, du jeu passé en paramètre, au jeu déjà dans la liste.
* **getJeuxFabricant**: Retourne une **Collection** contenant tous les jeux, se trouvant dans la **Bdd**, correspondant au fabricant passé en paramètre, ou null, s’il n’y est pas.
* **getJeu**: retourne le Jeu dont le titre du jeu et le nom du fabricant sont passés en paramètre, ou null s’il n’y est pas.
* **chercheConsole**: Retourne un **ArrayList** contenant tous les jeux pouvant fonctionner sur une console passée en paramètre. Si la console ne s’y trouve pas, un ArrayList vide est retourné.
* **chercheCote**: affiche à l'écran toutes les informations relatives aux jeux ayant une cote passée en paramètre.

Un programme de test (**Tp1.java**) vous est fourni pour tester les différents services. Lors du développement de votre classe **Bdd**, mettez en commentaire les sections de TP1.java qui ne sont pas encore prises en charge.

Quelques exigences pour l’implémentation :

* Votre devoir doit contenir l’interface **TestInterface** et les 3 classes suivantes :
  + **Bdd** : La classe qui implémente l’interface **TestInterface**, et qui contient la **LinkedHashMap** et les méthodes pour les services de la page 1 du présent énoncé.
  + **Jeu** : La classe qui contient le nom du fabricant, le titre du jeu, la cote et la Collection des consoles ainsi que les méthodes pour traiter ces attributs.
  + **Tp1**: La classe qui contient l’instance de la Bdd et qui teste les différents services. Remarque, il est probable que nous allons remplacer cette classe par notre propre version pour la correction, donc, n’y placez rien qui fait partie de la portion évaluée du programme.
* Chaque fichier remis doit obligatoirement commencer par un commentaire indiquant le nom de l’auteur.
* Pour la banque de données, vous devez utiliser une **Linked****HashMap** dont la clé est le nom du fabricant (Une chaîne de caractères) et la donnée (valeur) est une collection contenant les jeux (objets de la classe Jeu) de ce fabricant.
  + La collection contenant les jeux de ce fabricant doit être triée selon le titre du jeu, et, en cas d’égalité dans le titre du jeu, en ordre croissant du nom du fabricant.
  + Cette collection ne peut contenir des doublons.
* Pour chaque jeu, vous devez utiliser une collection pour conserver l’ensemble des consoles pour lesquelles le jeu est disponible. Cet ensemble doit conserver l’ordre de saisie et ne peut contenir de doublons. Les fichiers **jeux.txt** et **jeuxComplement.txt**, contiennent des données sur des jeux.
  + Les données sur un jeux se retrouvent sur une seule ligne dans le fichier : le nom du fabricant, le titre du jeu, la cote du jeu et la liste des consoles sur lesquelles le jeu est disponible sont séparés par un caractère’ ;’. À l’intérieur de la liste, les différentes consoles sont séparées par une virgule.
  + La sauvegarde dans un fichier doit avoir le même format que le fichier de données.

**Exemple de contenu du fichier :**

EA;Les Sims 4;T;PS4,MAC,XONE,PC

Dans l’exemple précédent, le fabricant est **EA** (Electronic Arts), le titre du jeu est **Les Sims 4**, la cote est **T** (teens) et le jeu est disponible pour les consoles suivantes : **PS4**, **MAC**, **XONE** et **PC**.

**Remise :** Il est important de noter que votre TP sera exécuté par votre auxiliaire sur une machine Windows des laboratoire DESI. Avant la remise, vérifiez que le code que vous avez produit (qui normalement fonctionne correctement chez vous) fonctionne bien sur les ordinateurs de la DESI.

Après avoir fait le ménage (éviter de nous envoyer les .class) dans votre répertoire tp1, compressez-le en format .zip. Pour Windows, placer vous sur le répertoire en question, puis le bouton droit de la souris, puis "envoyer vers", et choisissez "fichier compressé".

Le répertoire compressé doit ensuite être remis via Studium.

**Barème**

Ce TP1 est noté sur 60 points.

- La non remise du code (volontaire ou par erreur) est sanctionnée par la note 0.

***- Aucun travail envoyé par courriel ne sera accepté***

Classe Jeu : 14 points

Classe Bdd : 46 points

- Un programme qui ne compile pas ne peut se mériter plus de 30 points

- Un programme qui compile mais ne fait aucune des choses prévues dans la spécification ne peut mériter plus de 20 points.

-Si un travail est effectué en équipe, les coéquipiers se verront attribués, chacun, une part égale de la note du travail. Par exemple, un travail parfait en équipe de 2 donnera une note de 30 pour chacun des coéquipiers.

- La remise d’un travail qui utilise du code de quelqu’un d’autre :

-Si la source n’est pas citée, plagiat et ou fraude, donc mesures disciplinaires.

-Si la source est citée, peu ou pas de points accordés pour ce code.

- La qualité du code est importante, de plus, celui-ci doit être lisible, bien indenté et commenté.

**Des questions à propos de ce TP?**

L'adresse courriel: [dift1176@iro.umontreal.ca](mailto:dift1176@iro.umontreal.ca)

Pour faciliter le traitement de votre requête, inclure dans le sujet de votre courriel, au moins la chaîne : **ift1176.**

Évitez de nous envoyer votre code source par courriel pour dépannage, sauf si en réponse à une question, nous vous le demandons explicitement, profitez plutôt des périodes de pratiques libres et/ou de travaux pratiques, de Reda Boulos.