

IPO – TP 6 : héritage

<http://www.lri.fr/~blsk/IPO/>

Médias et médiathèque On souhaite développer une application pour une médiathèque qui met à disposition de ses utilisateurs des médias variés (livres, musique, vidéos). Chaque type de média sera naturellement représenté par une classe. Chaque média est décrit par un titre et un auteur. Les utilisateurs peuvent donner leur avis sur la qualité des médias en leur attribuant des notes comprises entre 0 et 5.

On veut pouvoir réaliser les opérations générales suivantes pour tous les médias.

- Représenter le média par une chaîne de caractères (méthode `toString()`) dont le format suit le modèle :

”Titre” par Auteur

- Donner son avis sur le média à l’aide d’une méthode `vote(int note)`, où la note doit être comprise entre 0 et 5.
- Consulter la moyenne des notes reçues par une méthode `moyenneNotes()`. Si le média n’a reçu aucune note, cette méthode doit renvoyer 0.

Remarque : on ne distingue pas dans cette étude les médias physiques des médias numériques, téléchargeables ou consultables en ligne.

1. Écrire le code d’une classe `Media` réalisant ces différents éléments.

Les différents types de médias vont ensuite avoir des caractéristiques spécifiques.

- Un livre possède un nombre de pages.
- Un média audio ou vidéo a une durée.
- Un média audio peut être un album ou un titre isolé.
- Un média vidéo peut posséder des sous-titres.

Pour chaque type de média, on a besoin de constructeurs adaptés qui permettront par exemple d’écrire :

```
Media l = new Livre("Les furtifs", "Alain Damasio", 704);
Media a1 = new Audio("Petit frère", "IAM", "single", 4);
Media a2 = new Audio("Origin of Symmetry", "Muse", 51);
Media v = new Video("The Train Job", "Joss Whedon",
                  new String[]{"en", "de", "fr"}, 45);
```

2. Écrire la classe `Livre`.
3. Écrire la classe `Audio`. On représentera le type (album ou *single*) par une chaîne de caractères. À défaut de précision, l’objet construit devra être un album. On suppose que la durée est donnée en minutes, arrondie à un nombre entier.
4. Écrire la classe `Video`. On représentera l’ensemble des sous-titres disponibles par une collection de chaînes de caractères.
5. Avez-vous trouvé quelque part une redondance que vous pourriez éliminer, quitte à créer une classe supplémentaire ?

On veut également pouvoir modéliser les séries. Une série est un média contenant une liste de médias préalablement définis. Remarquez qu’il est donc possible de noter indépendamment une série elle-même et chacun de ses épisodes.

```
Media s = new Serie("Firefly", "Joss Whedon", new Media[]{v});
v.vote(5);
s.vote(5);
```

6. Écrire la classe `Serie`, de sorte à ce qu’on puisse facilement ajouter de nouveaux éléments à une série déjà définie.
7. Écrire une méthode `moyenneAgregée()` qui calcule une note moyenne pour une série en combinant la moyenne de ses notes globales et les moyennes de ses épisodes.

Gestion des recherches simples Dans une classe `Mediatheque`, la base de données de la médiathèque est une `ArrayList` de `Media`.

8. Écrire une classe `Mediatheque` possédant ceci, ainsi qu'une méthode `add(Media m)` qui permet d'ajouter un média à la collection.

On va maintenant réaliser des méthodes de recherche dans la médiathèque en fonction de critères variés. Pour les recherches basées sur des chaînes de caractères, on ne tiendra pas compte de la casse (majuscule/minuscule).

9. Trouvez dans la bibliothèque standard Java une méthode de comparaison de chaînes de caractères adaptée.
10. Écrire une méthode `searchByTitle(String titre)` qui renvoie un `ArrayList<Media>` contenant l'ensemble des médias ayant le titre passé en paramètre.
11. Écrire une méthode `searchByAuthor(String auteur)` qui renvoie un `ArrayList<Media>` contenant l'ensemble des médias ayant l'auteur passé en paramètre.
12. Écrire une méthode `filtre(String critere, String valeur)` qui renvoie un `ArrayList<Media>` contenant l'ensemble des médias vérifiant le critère passé en paramètre. Ce critère peut porter :
 - sur l'auteur;
 - sur le titre;
 - sur le type de média (on pourra utiliser le test `nomObjet instanceof nomClasse` qui renvoie `true` si l'objet `nomObjet` est une instance de la classe `nomClasse` ou de l'une de ses classes filles)

Par exemple : `m.filtre("titre", "Firefly")` renverra l'ensemble des médias dont le titre est Firefly (avec ou sans majuscules) et `m.filtre("media", "audio")` renverra l'ensemble des musiques de la collection.