



SUJET :

JRefrain

DESCRIPTION DU TP

Introduction

L'entreprise « Soleil » vous charge de son projet le plus important, l'application « JRefrain » qui est une bibliothèque multimédia dont le nom n'a pas été tiré d'une application déjà existante. Dans le but d'obtenir une application compatible sur toutes les plateformes, vous devrez utiliser le langage de programmation JAVA afin de réaliser ce projet.

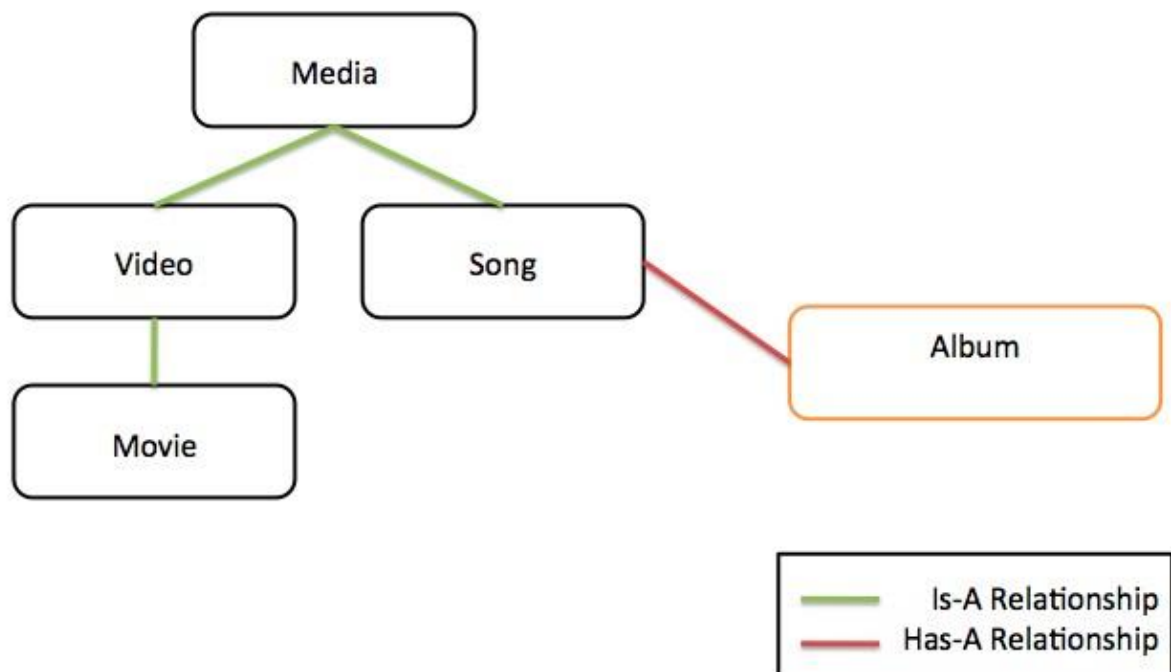
Vous serez en charge de la partie Modèle de l'application, on vous demande de réaliser une médiathèque qui aura pour but de gérer :

- la musique
- les vidéos

Ce logiciel étant très important pour la société, vous allez être guidé pour la réalisation et suivi par votre dévoué formateur.

1. Les classes

On vous demande pour le moment d'utiliser des tableaux pour stocker toutes les informations, des listes pourront être utilisées par la suite du projet mais pour le moment on vérifiera la faisabilité du projet. Voici un schéma de tous les objets que contiendra l'application avec leurs liaisons :



Voici le détail concernant chaque classe :

i. Media :

L'application gèrera un type nommé **Media**. Ce dernier aura pour caractéristiques, un *nom*, une *taille*, un *type*. Par contre, il ne sera pas possible de créer des objets de type média (cette classe nous sert uniquement à rassembler les caractéristiques et fonctions communes aux différents types de média).



Attention : La phrase soulignée implique une notion très importante en JAVA. J'espère que cette phrase n'est pas trop **abstraite** pour vous.

ATTRIBUTS		
Nom	Valeur Par Défaut	Valeur constante
name	« noname »	-
size	0	-
type	-	« notype »

Pour le champ « type » il est conseillé de faire une énumération avec les valeurs qui seront utilisées dans chaque classe. L'énumération sera tout simplement « Type ».



Tous les objets (sauf l'album) sont des Java Beans (POJO). Attention aux notions que tout cela implique ;)

ii. Song :

Comme son nom l'indique, cette classe représente une chanson. C'est, bien entendu, un Media mais qui sera de type Music, elle aura donc toutes les caractéristiques de la classe Media. Mais aussi les siennes :

ATTRIBUTS		
Nom	Valeur Par Défaut	Valeur constante
duration	0	-
trackNumber	0	-
type	-	« song »

iii. Video :

La vidéo est un media de type générique (il ne sera donc pas possible de créer des objets de ce type, j'espère que vous avez compris ce qu'il faut faire avec ces classes). Video a aussi des attributs qui lui sont propres comme :

ATTRIBUTS		
Nom	Valeur Par Défaut	Valeur constante
duration	0	-
type	-	« video »

iv. Movie :

Le film est une vidéo particulière qui aura pour caractéristiques supplémentaires de donner le nom du réalisateur et la liste des acteurs ayant participé au film.

ATTRIBUTS		
Nom	Valeur Par Défaut	Valeur constante
director	« unknown »	-
type	-	« movie »

v. Album :

Après en avoir fini avec les POJO (enfin il était temps), l'entreprise vous demande de passer à la création de classe de « stockage » afin d'utiliser vos objets (ce serait bête de ne pas les utiliser après avoir pris autant de temps pour les créer).

Le premier d'entre eux sera l'album, il sera capable de stocker les chansons (dans un tableau) ainsi que des informations complémentaires comme son nom, son artiste, le nombre de chansons.

Au niveau des fonctions spécifiques, il sera disponible d'ajouter (impossible d'ajouter des chansons qui ont le même nom) et de supprimer des chansons. Il faudra aussi trier l'album dans un premier temps par rapport au numéro de piste et ensuite par rapport à l'ordre alphabétique.

Concernant leur création, un album pourra être construit de 2 façons différentes :

- Sans paramètres, dans ce cas, le nom sera unknown et pourra contenir 20 chansons
- Avec un nom et un nombre précis de chansons.

ATTRIBUTS	
name	Le Titre de l'album (par défaut « unknown »). N'oubliez pas si la chanson a un nom d'album différent, on ne pourra pas l'ajouter ;)
songsList	Le tableau qui stockera les chansons (par défaut, il aura une taille de 20).
FONCTIONS	
addSong(Song)	Permet d'ajouter une chanson à l'album
removeSong(int)	Permet de supprimer une chanson de l'album grâce à son numéro dans la liste.

2. Les Exceptions

Maintenant que nos objets sont prêts, nous allons passer à une première phase de gestion qui sera celle des exceptions. On vous demande de gérer le cas où :

- i. FullAlbumException : On ajoute une chanson alors que l'album est rempli.
- ii. AlreadyInAlbumException : Lorsque l'on souhaite ajouter une chanson dont le nom existe déjà dans l'album.

3. Les Collections

Pendant que vous développiez l'application, l'entreprise a réfléchi à une amélioration pour faciliter le stockage d'informations dans vos classes. Désormais, vous allez utiliser quelques classes du Framework Collection.

Voici donc les modifications que l'on vous demande d'apporter :

- Créer une nouvelle classe nommée AlbumCollection qui au lieu d'utiliser un tableau pour stocker les chansons, utilisera l'une des classes du Framework Collection qui permettra de répondre à toutes les contraintes de la classe Album.

Servez-vous de la JavaDoc pour comprendre comment fonctionnent les différentes classes utilisées:



<https://docs.oracle.com/en/java/javase/15/index.html>

Pensez à regarder du côté de la fonction d'ajout ;)

4. Égalités

L'application sera amenée à comparer des musiques et des albums entre eux, afin de savoir si ces objets sont égaux ou différents :

Critères d'égalité pour les chansons :

- Deux chansons qui ont le même nom et la même durée sont égales.

Critères d'égalité entre deux albums :

- Deux albums ayant les mêmes chansons mais des noms différents sont égaux.
- Deux albums ayant le même nom et le même nombre de chansons sont égaux.

