

Albums de musique

On cherche à représenter des *albums* de musique caractérisés par les pistes audio (*audio tracks*) qui les composent.

Les pistes audio.

La classe `Track` appartient au paquetage `music.data`. Elle permet de définir les données sur les pistes audio. Un objet `Track` est défini par son titre (*title*) et sa durée (*duration*) en nombre de secondes.

La classe `Track` est définie par le diagramme :

music::data::Track
- title : String - duration : int
+ Track(title : String, duration : int) + getDuration() : int + getTitle() : String

- Q 1.** Donnez le code complet de la classe `Track` conforme au diagramme ci-dessus.
- Q 2.** Ajoutez à cette classe une méthode `equals` sachant que deux pistes sont considérées égales si elles ont même titre et même durée.
- Proposez des tests unitaires pour cette méthode.
- Q 3.** Ajoutez à cette classe une méthode `toString` dont le résultat reprendra le titre et la durée de la piste.

Genres musicaux.

Les albums sont classés par catégorie de genres musicaux. On suppose qu'il n'existe que 4 genres musicaux *rock*, *rap*, *pop*, *classique* (par exemple).

Le type `Category` du paquetage `music.data` permet de représenter les genres musicaux.

- Q 4.** Proposez la définition qui vous semble la mieux adaptée pour le type `Category`.

Les albums.

La classe `Album` appartient au paquetage `music`. Elle permet de représenter les albums de musique.

Un album de musique est défini par son titre, représenté par une chaîne de caractères, son genre musical de type `Category` et les pistes audio qui le composent. Le titre, la catégorie et le nombre de pistes composant un album sont fournis à la création de cet album. On utilise un **tableau** pour mémoriser les pistes d'un album.

Les pistes sont ajoutées une par une à un album. Un attribut est utilisé pour compter les pistes ajoutées à l'album. Lorsque toutes les pistes ont été ajoutées, l'album est dit *terminé* (*finished*) (et il ne l'est pas dans le cas contraire).

- Q 5.** Donnez le code de déclaration (entête) de la classe `music.Album`, de ses attributs et d'un constructeur, sachant qu'à sa création aucune piste n'a encore été ajoutée à l'album.
- Q 6.** Donnez le code de la méthode `isFinished()` de la classe `Album` dont le résultat est *vrai* si et seulement si l'album est terminé.
- Q 7.** La méthode `contains` de la classe `Album` prend en paramètre un objet `Track`. Son résultat est *true* si et seulement si la piste donnée en paramètre est égale à l'une des pistes de l'album.
- Q 7.1.** Proposez une ou des méthodes de test pour cette méthode.
- Q 7.2.** Donnez le code de cette méthode.

- Q 8.** La méthode `addTrack` de la classe `Album` permet d'ajouter une piste à un album. Les pistes sont toujours ajoutées les unes après les autres en commençant par la première (c'est-à-dire : « *le tableau est rempli dans l'ordre* »).

On supposera sans le vérifier par le code que l'on ajoute jamais deux pistes égales dans un même album.

Si l'album est déjà terminé au moment de l'ajout, cette méthode **déclenche** une exception `AlbumFinishedException` (dont la classe est supposée définie dans le paquetage `music.data`).

Q 8.1. Donnez la documentation, au format javadoc, de cette méthode.

Q 8.2. Ecrivez une ou plusieurs méthodes de test qui permettent de vérifier si une implémentation de la méthode `addTrack` vérifie le cahier des charges.

Q 8.3. Donnez le code de la méthode `addTrack` de `Album`.

Q 9. Définissez une méthode `totalDuration` qui retourne la durée totale d'un album terminé, cette méthode retourne -1 si l'album n'est pas terminé.

Q 10. On dit que deux albums sont égaux s'ils contiennent les mêmes pistes, mais celles-ci peuvent être dans des ordres différents dans les deux albums.

Donnez le code d'une méthode `equals` de la classe `Album` conforme à cette spécification.

Rappel : on suppose qu'un album ne contient jamais deux pistes égales.

Q 11. Donnez le code d'une méthode `main` qui :

1. crée deux *pistes* `p1` et `p2`, respectivement de titre "`t1`" et "`t2`" et de durée 230 et 145.
2. initialise une référence `d` avec un objet `Album` de titre "`td`", dont le genre musical est *rap* et composé de 2 pistes,
3. ajoute à l'album, dans l'ordre, les pistes `p1` et `p2` puis affiche la durée totale de l'album.