

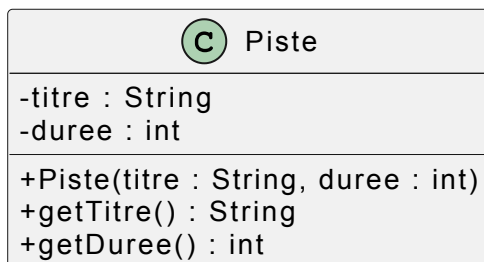
Développement orienté objet

TD4 : Albums de musique

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

Les pistes audio

La classe **Piste** permet de définir les données sur les pistes audio. Un objet **Piste** est défini par son titre et sa durée en nombre de secondes. La classe **Piste** est définie par le diagramme :



Question 1

Donnez le code complet de la classe **Piste** conforme au diagramme ci-dessus.

Question 2

Ajoutez à cette classe une méthode **égale** sachant que deux pistes sont égales si elles ont le même titre et la même durée.

Question 3

Ajoutez à cette classe une méthode **toString** qui retourne une chaîne de caractères contenant le titre et la durée de la piste.

Les albums

La classe **Album** permet de représenter les albums de musique. Un album de musique est défini par son titre, représenté par une chaîne de caractères et les pistes audio qui le composent. Le titre 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 à une à l'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* (et il ne l'est pas dans le cas contraire).

Question 4

Définissez la classe `Album` en TypeScript. La classe devra contenir les attributs et le constructeur, sachant qu'à sa création aucune piste n'a encore été ajoutée à l'album.

Question 5

Donnez le code de la méthode `estFini()` de la classe `Album` dont le résultat est `true` si et seulement si l'album est terminé.

Question 6

La méthode `contient` de la classe `Album` prend en paramètre un objet `Piste`. Son résultat est `true` si et seulement si l'album contient la piste passée en paramètre. Donnez le code de cette méthode.

Question 7

La méthode `ajouterPiste` 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 jamais le vérifier par le code que l'on n'ajoute jamais deux pistes égales dans un même album.

Si l'album est déjà terminé, la méthode `ajouterPiste` ne fait rien.

Donnez le code de cette méthode.

Question 8

Définissez une méthode `duréeTotale` qui retourne la durée totale d'un album terminé. Cette méthode retourne 0 si l'album n'est pas terminé.

Question 9

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

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

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

Question 10

Donnez le diagramme de classes de la classe `Album`.

Question 11

Donnez le code d'un programme principal qui :

1. crée deux pistes `p1` et `p2`, respectivement de titre `"t1"` et `"t2"` et de durée 230 et 180 secondes.
2. initialise une référence `d` sur un objet `Album` de titre `"td"` 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.

