

# TD : La classe Album et Piste

## Questions Avancées

1. **Fusion d'albums** Implémentez une méthode `fusionnerAvec(autre: Album): Album` qui prend un autre album en paramètre et retourne un **nouvel album** contenant toutes les pistes des deux albums sans doublons. Le nouvel album porte le titre "Fusion de `titre1` et `titre2`" et a une capacité égale à la somme des capacités des deux albums.
2. **Suppression de piste** Ajoutez une méthode `supprimerPiste(titre: string): void` à la classe `Album` pour supprimer une piste en fonction de son titre. Si la piste n'existe pas, la méthode ne fait rien.
3. **Extraction d'un sous-album** Implémentez `extraireSousAlbum(n: number): Album` qui crée un **nouvel album** contenant les `n` premières pistes de l'album original. Si `n` est supérieur au nombre total de pistes, l'album extrait contiendra toutes les pistes.
4. **Tri des pistes** Ajoutez `trierPistes(): void` qui trie les pistes par ordre alphabétique de leur titre.
5. **Comparaison d'albums par durée totale** Ajoutez `comparaisonDuree(autre: Album): number` qui retourne 1 si l'album actuel est plus long que `autre`, -1 s'il est plus court, et 0 s'ils ont la même durée.
6. **Album le plus long d'une liste** Écrivez une fonction `trouverAlbumLePlusLong(albums: Album[]): Album | null` qui retourne l'album ayant la plus grande durée totale, ou `null` si la liste est vide.
7. **Fusionner les albums terminés** Écrivez `fusionnerAlbumsTerminees(albums: Album[]): Album | null` qui fusionne tous les albums terminés en un seul album et retourne `null` si aucun album terminé n'existe.
8. **Suppression des doublons d'albums** Écrivez `supprimerDoublonsAlbums(albums: Album[]): Album[]` qui supprime les albums dupliqués (ayant exactement les mêmes pistes).