

Exercice : Gestion d'une bibliothèque de médias

Vous travaillez sur un projet de gestion de bibliothèque de médias (livres, DVD, CD) en utilisant C#. Le projet doit gérer différentes collections de médias, permettre l'emprunt et le retour d'articles, ainsi que la recherche d'articles spécifiques.

Partie 1 : Classes, Héritage et Polymorphisme (15 minutes)

1.1. Créez une classe de base appelée "Media" avec les propriétés suivantes :

- Titre (string)
- Numéro de référence (int)
- Nombre d'exemplaires disponibles (int)

1.2. Déclarez trois sous-classes : "Livre", "DVD" et "CD" qui héritent de la classe "Media". Ajoutez des propriétés spécifiques à chaque type de média (par exemple, "Auteur" pour les livres, "Durée" pour les DVD, "Artiste" pour les CD).

1.3. Implémentez une méthode polymorphique "AfficherInfos" dans chaque sous-classe qui affiche les informations spécifiques au type de média.

Partie 2 : Surcharge des opérateurs (15 minutes)

2.1. Créer une classe Tools

2.2 Surchargez l'opérateur "+" pour permettre l'ajout de médias à la bibliothèque. Assurez-vous de mettre à jour le nombre d'exemplaires disponibles correctement.

2.3. Surchargez l'opérateur "-" pour permettre le retrait de médias de la bibliothèque. Gérez les exceptions si le nombre d'exemplaires disponibles est insuffisant.

Partie 3 : Collections et Indexeurs (30 minutes)

3.1. Créez une classe appelée "Library" qui contient une collection de médias (utilisez List<T>).

3.2. Implémentez un indexeur dans la classe "Library" pour permettre l'accès aux médias par leur numéro de référence.

3.3 Implémenter les méthodes suivantes :

1. **Ajouter un Média** : Ajoute un média à la bibliothèque.
2. **Retirer un Média** : Retire un média de la bibliothèque.
3. **Emprunter un Média** : Une méthode pour emprunter un média de la bibliothèque. Cette méthode devrait vérifier si le média est disponible, mettre à jour le nombre d'exemplaires disponibles et enregistrer les détails de l'emprunt.
4. **Retourner un Média** : Une méthode pour retourner un média emprunté. Cette méthode devrait augmenter le nombre d'exemplaires disponibles et mettre à jour les enregistrements d'emprunt.

5. **Rechercher un Média par Titre ou Auteur** : Une méthode qui permet aux utilisateurs de rechercher des médias par titre, auteur, ou d'autres critères de recherche.
6. **Lister les Médias Empruntés par un Utilisateur** : Une méthode pour afficher les médias empruntés par un utilisateur spécifique. Cette méthode nécessite de tenir un suivi des emprunts. (Classe Emprunt, modification des fonctions Emprunter / Retourner Media)
7. **Afficher les Statistiques de la Bibliothèque** : Une méthode qui affiche des statistiques telles que le nombre total de médias dans la bibliothèque, le nombre d'exemplaires empruntés, le nombre d'exemplaires disponibles, etc.

Partie 4 : Sérialisation JSON (10 minutes)

4.1 Sauvegarder et Charger la Bibliothèque depuis un Fichier : Des méthodes pour sauvegarder l'état de la bibliothèque dans un fichier et pour charger la bibliothèque à partir d'un fichier, de sorte que les données puissent être persistantes entre les sessions en utilisant la sérialisation JSON.

Partie 5 : Programme principal (20 minutes)

- 5.1. Dans un programme principal, créez une instance de "Library" et ajoutez plusieurs médias à la bibliothèque en utilisant l'opérateur "+".
- 5.2. Effectuez des opérations d'emprunt et de retour de médias en utilisant des blocs try-catch pour gérer les exceptions potentielles.
- 5.3. Affichez les informations de chaque média dans la bibliothèque en utilisant la méthode "AfficherInfos".
- 5.4 Sauvegarder, puis charger la bibliothèque et afficher les informations sauvegardées.

Partie 6 : Gestion des exceptions (20 minutes)

- 4.1. Ajoutez des mécanismes de gestion des exceptions pour les opérations d'emprunt et de retour de médias dans la classe "Library". Par exemple, si un média n'est pas disponible pour l'emprunt, lancez une exception personnalisée.
- 4.2. Utilisez des blocs try-catch pour gérer les exceptions possibles lors de l'utilisation de la classe "Library" dans un programme principal. Affichez des messages d'erreur informatifs en cas d'exception.

Assurez-vous de bien commenter votre code et de suivre les meilleures pratiques de programmation en C#.