

Énoncé :

1. Création de la Classe Article :

- Définissez une classe `Article` avec les attributs suivants :
 - Référence (chaîne de caractères)
 - Désignation (chaîne de caractères)
 - PrixHT (nombre décimal)
 - TauxTVA (nombre décimal) - cet attribut est commun à tous les articles.
- Tous les attributs doivent être privés et uniquement accessibles via des propriétés en lecture et écriture.

2. Constructeurs :

- Implémentez un constructeur par défaut qui initialise les attributs à des valeurs par défaut.
- Ajoutez un constructeur qui initialise tous les attributs avec des valeurs spécifiées.

3. Méthodes :

- `CalculerPrixTTC()`: Méthode qui calcule et retourne le prix TTC de l'article ($\text{PrixHT} + \text{PrixHT} * \text{TauxTVA} / 100$).
- `AfficherArticle()`: Méthode qui affiche les informations de l'article.

4. Programme Principal (Main) :

- Dans la méthode `Main`, créez une liste d'au moins 5 objets `Article` en utilisant les différents constructeurs.
- Parcourez la liste et affichez le prix TTC de chaque article.
- Notez que le TauxTVA est partagé entre tous les articles. Ajustez votre classe et le programme principal pour refléter cette caractéristique.

5. Exigences Supplémentaires :

- Assurez-vous que le TauxTVA est un attribut statique (partagé) au niveau de la classe `Article`.
- Testez votre code pour vérifier que la modification de TauxTVA dans un objet `Article` affecte tous les autres objets.

Énoncé :

1. Création de la Classe Livre :

- Définissez une classe `Livre` avec les attributs suivants :
 - ISBN (chaîne de caractères)
 - Titre (chaîne de caractères)
 - Auteur (chaîne de caractères)
 - Prix (nombre décimal)
- Tous les attributs doivent être privés et uniquement accessibles via des propriétés en lecture et écriture.

2. Constructeur :

- Implémentez un constructeur sans arguments qui initialise les attributs à des valeurs par défaut.

3. Méthodes :

- `SaisirLivre()`: Méthode pour saisir les informations d'un livre depuis la console.
- `AfficherLivre()`: Méthode qui affiche les informations du livre.

4. Programme Principal (Main) :

- Dans la méthode `Main`, créez une liste pour stocker plusieurs objets `Livre`.
- Permettez à l'utilisateur de saisir les informations pour plusieurs livres et ajoutez-les à la liste.
- Implémentez une fonctionnalité de recherche par ISBN pour trouver un livre dans la liste.
- Ajoutez une option permettant à l'utilisateur de modifier les informations d'un livre trouvé par sa recherche.

5. Exigences Supplémentaires :

- Assurez-vous que les méthodes de saisie et d'affichage sont bien encapsulées dans la classe `Livre`.
- Testez votre code pour s'assurer qu'il gère correctement la saisie, l'affichage, la recherche et la modification des livres.

Énoncé :

1. Création de la Classe Client :

- Définissez une classe `Client` avec les attributs suivants :
 - ID (entier)
 - Nom (chaîne de caractères)
 - Adresse Email (chaîne de caractères)
- Les attributs doivent être privés et accessibles via des propriétés en lecture et écriture.

2. Fonctions CRUD :

- **Create (Créer) :** Une méthode pour ajouter un nouveau client à la liste.
- **Read (Lire) :** Une méthode pour afficher les informations de tous les clients.
- **Update (Mettre à jour) :** Une méthode pour modifier les informations d'un client spécifique dans la liste.
- **Delete (Supprimer) :** Une méthode pour supprimer un client de la liste.

3. Programme Principal (Main) :

- Créez une liste pour stocker des objets de type `Client`.
- Implémentez un menu dans la méthode `Main` qui permet à l'utilisateur de choisir parmi les opérations CRUD.
- Assurez-vous que l'utilisateur puisse ajouter, voir, modifier et supprimer des clients via ce menu.

4. Exigences Supplémentaires :

- Assurez-vous que les ID des clients sont uniques.
- Implémentez des validations pour éviter les erreurs, comme la suppression d'un client inexistant.