

Série N°5 TD/TP
Module : Programmation Orientée Objet en C++
Filière : Génie Informatique
Niveau : GI1

Un libraire souhaite gérer sa base de données d'ouvrages à l'aide d'un programme.

La classe Livre

La classe **Livre** dotée des attributs suivants : **titre** (**string**), **auteur** (**string**), nombre de pages (**int**) ainsi qu'un attribut indiquant si l'ouvrage est un **bestseller** ou pas.

Dotez votre classe :

- d'une méthode **calculer_prix** permettant de calculer le prix d'un livre. Ce prix est calculé comme étant le nombre de pages multiplié par 0.3 auquel on ajoute 50 francs si c'est un bestseller.
- d'un constructeur prenant comme arguments des paramètres permettant d'initialiser tous les attributs de la classe.
- d'un destructeur virtuel vide
- d'une méthode **afficher** affichant la valeur des attributs d'un livre. Le prix du livre ne sera pas affiché par cette méthode. Inspirez-vous de l'exemple de l'exécution donné ci-dessous pour réaliser vos affichages.

Les attributs de votre classe seront privés. Les définitions des méthodes de votre classe se feront à l'extérieur du corps de la classe.

1. Testez votre classe au moyen d'un petit **main** de sorte à produire quelque chose comme :

Titre : P00 en langageC++ Auteur : F. Mohammed Nombre de pages :822 Bestseller :oui Prix: 296.6

Titre: Lefou Auteur: Gogol Nombre de pages:252 Bestseller: non Prix: 75.6

Exercice 1 Des sous-classes de Livres(Héritage)

Continuez en modifiant les droits d'accès des attributs privés de la classe **Livre** de sorte à ce qu'ils soient accessibles directement dans les sous-classes de **Livre**.

Codez ensuite les sous-classes suivantes de la classe **Livre** :

1. **Roman** : caractérisé par un attribut indiquant s'il s'agit d'une **biographie** ou pas. Cet attribut sera protégé.
2. **BeauLivre** (ouvrages illustrés à la présentation soignée) : pas d'attributs spécifiques

La classe **Roman** possède elle aussi une sous-classe, **Policier**, modélisant les romans policiers. Cette sous-classe n'a pas d'attributs spécifiques. Vous la coderez également.

Les sous-classes ne doivent pas restreindre les droits d'accès des attributs et méthodes hérités des classes **Livre** et **Roman**.

La méthode **calculer_prix** codée dans la classe **Livre** donne le prix *standard* d'un livre.

Dotez vos sous-classes des méthodes publiques suivantes :

1. **Classe Roman** :
 - un constructeur prenant en arguments les paramètres permettant d'initialiser l'ensemble des attributs de la classe **Roman**.
 - un destructeur virtuel vide
 - une méthode **afficher** permettant d'afficher la valeur de l'ensemble des attributs d'un roman. Le prix du roman ne sera pas affiché par cette méthode. Inspirez-vous de l'exemple d'exécution ci-dessous pour réaliser vos affichages.
2. **Classe BeauLivre** :
 - un constructeur. Ce dernier devra prendre en arguments les paramètres permettant d'initialiser l'ensemble des attributs hérités de **Livre**.

- un destructeur virtuel vide
- une méthode `calculer_prix` calculant le prix d'un **BeauLivre** comme le prix standard augmenté de 30 francs. Par exemple un **BeauLivre** de 100 pages (prix standard 30 francs) et qui est un best-seller (prix standard 30+50 = 80 francs) se vendra 110 francs (= 80 + 30 francs).

3. Classe **Policier** :

- un constructeur. Ce dernier devra prendre en arguments les paramètres permettant d'initialiser l'ensemble des attributs hérités de la classe **Roman**.
- une méthode `calculer_prix` calculant le prix d'un roman **Policier** comme le prix standard auquel on enlèvera de 10 francs. Si le prix ainsi obtenu est négatif, le prix de l'ouvrage sera de 1 franc.
- un destructeur virtuel vide

Le calcul du prix dans les méthodes `calculer_prix` redéfinie dans les sous-classes **Policier** et **BeauLivre** doit faire appel à la méthode `calculer_prix` (prix standard) héritée de la classe **Livre**.

La méthode `afficher` de la classe **Roman** doit également faire appel à celle de la classe **Livre** afin d'éviter la duplication inutile du code commun.

1. Testez votre programme au moyen de main de sorte à produire quelque chose comme :

Titre : Lechien des Baskerville Auteur : A.C.Doyle Nombre de pages : 221 Bestseller : non Ce roman n'est pas une biographie Prix : 56.3	Bestseller : non Ce roman n'est pas une biographie Prix : 84.9
Titre : Le Parrain Auteur : A.Cuso Nombre de pages : 367 Bestseller : oui Ce roman n'est pas une biographie Prix : 150.1	Titre : Mémoires de géronimo Auteur : S.M. barrett Nombre de pages : 173 Bestseller : non Ce roman est une biographie Prix : 51.9
Titre : Lebaronperché Auteur : I.Calvino Nombre de pages : 283	Titre : Fleuves d'europe Auteur : C. Osborne Nombre de pages : 150 Bestseller : non Prix : 75

Exercice 2

Une collection de Livres(Polymorphisme)

Collection de livres

Codez maintenant une classe **Librairie** ayant un attribut privé nommé `livres`. Cet attribut est une collection de **Livres** codée comme un tableau dynamique.

Prototypez et définissez ensuite les méthodes suivantes de la classe **Librairie** :

- `void ajouter_livre(Livre*)` qui ajoute un (pointeur sur) un **Livre** la collection
- `void afficher()` const faisant appel à la méthode `afficher()` (de la classe **Livre** ou de ses classes dérivées) sur chaque élément de la librairie
- `void vider_stock()` permettant de supprimer tous les livres stockés dans la collection, tout en libérant leur espace mémoire.

19.3.2 Affichage polymorphique

Modifiez la fonction `afficher` associée à la classe **Livre** de sorte à ce qu'elle affiche aussi le prix du livre. Le calcul du prix à afficher se fera au moyen de la fonction `calculer_prix()`.

Dans la classe **Librairie**, la méthode `afficher` de la classe **Livre** est invoquée de façon polymorphique sur les éléments d'une collection de livres.

Modifiez les déclarations des méthodes `afficher` relatives aux livres de sorte à ce qu'elle puissent faire l'objet d'une résolution dynamique des liens. Faites-en de même avec les méthodes `calculer_prix` (car `calculer_prix` est invoquée par `afficher`).

En principe, votre fichier `librairie.cc` contient maintenant tous les outils nécessaires pour traiter les objets de la collection hétérogène de façon polymorphique.

1. Tester le code au moyen de la fonction `main` de façon à obtenir :

Titre : Le chien des Baskerville
Auteur : A.C.Doyle
Nombre de pages : 221
Bestseller : non
Prix: 56.3
Ce roman n'est pas une biographie

Policier

Titre : Le Parrain
Auteur : A.Cuso
Nombre de pages : 367
Bestseller : oui
Prix: 150.1
Ce roman n'est pas une biographie

Policier

Titre : Le baron perché
Auteur : I. Calvino
Nombre de pages : 283
Bestseller : non
Prix: 84.9
Ce roman n'est pas une biographie

Roman

Titre : Mémoires de géronimo
Auteur : S.M. barrett
Nombre de pages : 173
Bestseller : non
Prix: 51.9
Ce roman est une biographie

Roman

Titre : Fleuves d'europe
Auteur : C. Osborne
Nombre de pages : 150
Bestseller : non
Prix: 75

BeauLivre

à partir du code suivant à compléter :

```
int main(){  
    Librairie l;  
    ...  
}
```

```
return 0;  
}
```