INF1010

Programmation Orientée-Objet

Travail pratique #4
Polymorphisme et héritage multiple

Objectifs : Permettre à l'étudiant de se familiariser avec le concept de polymorphisme et d'héritage multiple

Remise du travail : vendredi, 4 mars 2016, 23h

Références : Notes de cours sur Moodle & Chapitre 8 et 19 du livre Big C++ 2e éd.

Documents à remettre : Les fichiers .cpp et .h complétés réunis sous la forme d'une archive au format .zip.

Directives : Directives de remise des Travaux pratiques sur Moodle

Les en-têtes (fichiers, fonctions) et les commentaires sont obligatoires.

Les travaux dirigés s'effectuent obligatoirement en équipe de deux personnes faisant partie du même groupe.

Veuillez suivre le guide de codage

Travail à réaliser

Dans ce TP nous poursuivrons l'implémentation de notre bibliothèque en y incorporant de nouveaux éléments et en améliorant la structure utilisée pour l'héritage.

Lors du dernier TP, il était possible de remarquer certains problèmes qui nous empêchaient d'avoir accès aux méthodes des classes dérivées, qui auraient pu être résolues par le polymorphisme. L'ajout du polymorphisme va permettre de régler ces problèmes, puisque les pointeurs des classes de base permettent d'appeler les méthodes de la classe dérivée, lorsque celles-ci sont implémentées.

Un autre sujet du TP est l'héritage multiple, qui permet à une même classe de dériver de 2 classes de base. Ainsi, il sera possible d'utiliser des livres numériques, qui sont à la fois des livres et des objets numériques.

Grâce à la subvention reçue, on vous demandera de moderniser la bibliothèque et d'offrir des solutions plus adaptée à la technologie mobile.

ATTENTION : C'est à vous de choisir la portée de vos attributs (private, protected ou public) sauf si explicitement mentionné.

ATTENTION : C'est à vous de choisir les méthodes qui sont virtuelles ou virtuelles pures, sauf si c'est explicitement mentionné.

Classe inchangées

Les classes suivantes restent inchangées par rapport au TP3. Toutefois, **vous devez ajouter les *virtual*** selon votre jugement.

- Abonne
- Etudiant
- EtudiantBaccalaureat
- Professeur
- Livre
- DVD

Classe *ObjetEmpruntable*

La classe *ObjetEmpruntable* sert à représenter les objets qui peuvent être empruntés par un abonné.

Les attributs et méthodes liés au TP précédent restent inchangés, sauf si spécifié. N'oubliez pas d'ajouter les *virtual* si nécessaire.

ObjetEmpruntable implémente les nouvelles méthodes suivantes :

- Une méthode *obtenirNomClasse()* qui permet de retourner le nom de la classe d'un pointeur, qu'il soit de type *ObjetEmpruntable* ou de ses classes dérivés. **Cette méthode est fournie.**
- Une méthode *void afficherObjetEmpruntable(std::ostream& o)*.
    - Cette méthode doit déterminer de quelle sous-classe fait partie l'abonné.
    - Une fois déterminé, la méthode doit faire un *dynamic_cast*<> pour obtenir un pointeur de la classe appropriée, dans le but d'appeler l'opérateur << de la sous-classe appropriée. Ex *dynamic_cast* :

```
Classe_Derivé* = dynamic_cast<Classe_Derive*> (Classe_Base*);
```

    - Puisque *this* est un pointeur constant, alors assurez-vous d'utiliser le mot-clé *const* lors de la conversion du type de pointeur.

Classe *Abonne*

La classe *Abonne* sert à représenter les abonnés d'une bibliothèque.

Les attributs et méthodes liés au TP précédent restent inchangés, sauf si spécifié. N'oubliez pas d'ajouter les *virtual* si nécessaire.

Abonne implémente les nouvelles méthodes suivantes :

- Une méthode *obtenirNomClasse()* qui permet de retourner le nom de la classe d'un pointeur, qu'il soit de type *Abonne* ou de ses classes dérivés. **Cette méthode est fournie.**

Classe *Emprunt*

La classe *Emprunt* sert à représenter l'emprunt d'un objet par un abonné.

Les attributs et méthodes liés au TP précédent restent inchangés, sauf si spécifié. N'oubliez pas d'ajouter les *virtual* si nécessaire.

Cette classe va contenir les nouveaux attributs suivants :

- 1 seul pointeur de type *ObjetEmpruntable*, qui remplace les pointeurs *DVD* et *Livre*.

Emprunt implémente les méthodes suivantes :

- Un constructeur par paramètre qui prend un matricule (string), un *ObjetEmpruntable* et une date (*entier positif*). Ce constructeur va **remplacer** les constructeurs par paramètre des TP précédents.
- Les méthodes d'accès et de modification du nouvel attribut.
- Un nouveau destructeur (si requis).

- Un opérateur <<, qui affiche le matricule de l'abonné, affiche le nom de la classe de l'objet, puis appelle l'opérateur << de l'objet empruntable approprié ; tel que présenté dans l'exemple à la fin du document.

Classe *ObjetNumerique*

La classe *ObjetNumerique* sert à représenter un objet numérique détenu dans la banque de données virtuelles de la bibliothèque.

Cette classe est une classe abstraite pure, elle va servir uniquement d'interface pour permettre l'héritage multiple sans causer de problèmes. En d'autres termes, la classe ne contiendra **aucun attribut**, et **uniquement des méthodes virtuelles pures**. Cette classe n'a donc pas besoin de fichier *« .cpp »*.

ObjetNumerique implémente les méthodes suivantes :

- Bien que la classe ne contienne aucun attribut, il est tout de même demandé de faire des méthodes d'accès et de modifications. Ceci va forcer les classes dérivées à implémenter les attributs nécessaires. Ainsi, on demande de faire les méthodes d'accès et de modification virtuelles pures pour :
    - La taille du document numérique (entier positif), en octets.
    - Le lien internet (*string*) vers l'objet numérique.
- Un destructeur.

Classe *LivreNumerique*

La classe *LivreNumerique* est un objet qui dérive à la fois de *Livre* et de *ObjetNumerique*.

LivreNumerique contient les attributs suivants :

- La taille du document numérique (entier positif), en octets.
- Le lien internet (*string*) vers l'objet numérique.
- Le format du document, de type *FORMAT_DOCUMENT*, tel que fourni dans les documents de bases. Cet attribut permet de représenter si le format est en « .pdf », « .epub », « .docx »…

LivreNumerique implémente les méthodes suivantes :

- Un constructeur par défaut.
- Un constructeur par paramètre pour tous les attributs de cette classe et de ses 2 classes mères.
- Un destructeur.
- Des méthodes d'accès et de modification aux attributs.
- Une méthode *obtenirFormatStr()* qui permet de retourner un string associé au format du document. Par exemple, si le format est *FORMAT_PDF*, alors la valeur retournée sera *« pdf »*.

- Une redéfinition de la méthode *Livre::recherche()* pour que la méthode recherche aussi si le mot-clé est **identique** au format du livre.
- Un opérateur << qui affiche que c'est un objet numérique, affiche la taille du document, puis affiche le lien internet. Ensuite, il appelle l'opérateur<< de la classe *Livre*; tel que présenté dans l'exemple à la fin du document.

Classe *Bibliotheque*

La classe *Bibliotheque* est celle qui fait le lien entre toutes les classes précédentes.

Cette classe est partiellement écrite, il ne reste qu'à compléter les méthodes demandées.

Bibliotheque contient les attributs suivants :

- Vecteur de pointeurs *d'abonnés*.
- Vecteur de pointeurs de *ObjetEmpruntable* (qui remplace les vecteurs de *DVD* et de *Livre*)
- Vecteur de pointeurs d'*Emprunt*.

Bibliotheque implémente les nouvelles méthodes suivantes :

Indication : Certaines méthodes demandées sont très semblables à celles présentes dans le TP3, et une partie importante du code pourrait être fournie.

- Un constructeur par défaut.
- Un destructeur.
- Une méthode *trouverObjetEmpruntable()* qui prend une cote (string) en paramètre, recherche l'objet empruntable et retourne un pointeur de type *ObjetEmpruntable*. Si aucun objet empruntable n'est retrouvé, un pointeur nul est retourné sinon.
- Une méthode *obtenirClasseObjet*() qui prend une cote (string) en paramètre, et renvoie le nom de la classe correspondant à l'objet ayant cette cote (« *Livre* », « *DVD* », « *LivreNumerique* »).
- La méthode *ajouterObjetEmpruntable ()* qui permet d'ajouter l'*ObjetEmpruntable* reçu en paramètre seulement s'il n'est pas déjà dans le vecteur. Si l'ajout a été fait, la méthode renvoie true, false sinon.
- La méthode *retirerObjetEmpruntable()* qui permet de retirer l'objet empruntable en utilisant la cote reçue en paramètre. Il est retiré seulement s'il n'est pas emprunté. La méthode retourne donc un booléen *true* s'il est retiré, sinon *false*.
- La méthode *rechercherObjetEmpruntable()* qui prend en paramètre une chaîne de caractères (string),
    - Cette méthode va rechercher les différents éléments empruntables de la bibliothèque qui contiennent l'information désirée en **utilisant la méthode appropriée de la classe**. Pour chaque élément trouvé les informations seront affichées à l'aide de la méthode *afficherObjetEmpruntable()* de la classe *ObjetEmpruntable*. La méthode prendra la variable ***cout*** comme paramètre d'entrée.
    - N'oubliez pas de parcourir tous les éléments qui peuvent être empruntés

    - Sirien n'a été trouvé, affichez un message.
- La méthode *emprunter()* qui prend en paramètres le matricule d'un abonné, la cote d'un objet empruntable et la date de retour associée en fonction du statut de l'abonné. Cette méthode doit retourner une valeur booléenne indiquant si l'emprunt a été fait ou non.
    - Elle doit s'assurer que le nombre d'emprunts ne dépasse pas la limite par type d'abonné.
    - Elle doit diminuer le nombre d'objets associés disponibles.
- La méthode *infoAbonne()* qui prend en paramètre un matricule d'abonné et affiche les informations qui le concerne en utilisant l'opérateur << approprié.
    - Cette méthode doit déterminer de quelle sous-classe fait partie l'abonné.
    - Une fois déterminé, la méthode doit faire un *dynamic_cast*<> pour obtenir un pointeur de la classe appropriée, dans le but d'appeler l'opérateur <<.
- L'opérateur += qui est défini 2 fois, avec un paramètre d'entrée différent :
    - Une définition qui prend un pointeur de la classe *Abonne*. Son comportement est similaire à *ajouterAbonne()*.
    - Une définition qui prend un pointeur de la classe *ObjetEmpruntable*. Son comportement est similaire à *ajouterObjetEmpruntable()*.
- L'opérateur -= qui est défini 2 fois, avec un paramètre d'entrée différent
    - Une définition prend en paramètre un pointeur *Abonne*. Son comportement est similaire à *retirerAbonne()*.
    - Une définition prend en paramètre un pointeur *ObjetEmpruntable*. Son comportement est similaire à *retirerObjetEmpruntable()*.

Main.cpp

Le programme principal contient des directives à suivre pour tester les différentes méthodes que vous devez implémenter dans les différentes classes. Vous n'avez pas à compléter le main, assurez-vous juste du bon fonctionnement des méthodes appelées.

Le résultat final devrait être similaire à ce qui suit :

```
_____________________________________________
CREATION DES ABONNES ET DES OBJETS EMPRUNTABLES


_____________________________________________
AJOUT DES DIFFERENTS ELEMENTS A LA BIBLIOTHEQUE


_____________________________________________
CREATION DES OBJETS NUMERIQUES

Objet Numerique!  Taille : 1270 oct.  vvv.3Mousquetaire.com  Format : epub
Information sur le livre :
NUM392. Les 3 mousquetaires. 1844. 11 ans et plus. Auteur : A. Dumas; Genre : Roman

Objet Numerique!  Taille : 1270 oct.  vvv.3Mousquetaire.com  Format : epub
Information sur le livre :
NUM393. Les 3 mousquetaires 2. 1846. 11 ans et plus. Auteur : A. Dumas; Genre : Roman
_____________________________________________
TESTS D'EMPRUNTS

1630236 a reussi 2 emprunt(s) !
1630236 a rate 5 emprunt(s) !
1782965 a reussi 4 emprunt(s) !
1782965 a rate 3 emprunt(s) !
1865487 a reussi 5 emprunt(s) !
1865487 a rate 2 emprunt(s) !
p878546 a reussi 3 emprunt(s) !
p878546 a rate 4 emprunt(s) !
_____________________________________________
INFO ABONNE AVANT RETOUR

Albus, Dumbledore. 78 ans. #p878546
LISTE DE LIVRES :
1 - Usager #p878546. class Livre. Information sur le livre :
BD302. Harry Potter et le prisionier d'Azkaban. 1999. 3 ans et plus. Auteur : JK ROWLING; Genre : Science-Fiction
Retour prevu : 160204

2 - Usager #p878546. class Dvd. Information sur le Dvd :
D7203. Avenger. 2012. 3 ans et plus. Realisateur : Nick Fury; Acteurs : Iron Man; Thor; Hulk; Captain America; Black Window;
Retour prevu : 160204

3 - Usager #p878546. class LivreNumerique. Objet Numerique!  Taille : 1270 oct.  vvv.3Mousquetaire.com  Format : epub
Information sur le livre :
NUM392. Les 3 mousquetaires. 1844. 11 ans et plus. Auteur : A. Dumas; Genre : Roman
Retour prevu : 160204

Limite d'emprunts : 6
LISTE DES ECOLES : Poudlard; UdM; Polytechnique;
_____________________________________________
TESTS RETOUR LIVRE

BD302 remis par 1630236
Echec remise
Echec remise
Echec remise
D7203 remis par p878546
Echec remise
Echec remise
```

```
_____________________________________________
INFO ABONNE APRES RETOUR

Albus, Dumbledore. 78 ans. #p878546
LISTE DE LIVRES :
1 - Usager #p878546. class Livre. Information sur le livre :
BD302. Harry Potter et le prisionier d'Azkaban. 1999. 3 ans et plus. Auteur : JK ROWLING; Genre : Science-Fiction
Retour prevu : 160204

2 - Usager #p878546. class LivreNumerique. Objet Numerique!  Taille : 1270 oct.  vvv.3Mousquetaire.com  Format : epub
Information sur le livre :
NUM392. Les 3 mousquetaires. 1844. 11 ans et plus. Auteur : A. Dumas; Genre : Roman
Retour prevu : 160204

Limite d'emprunts : 6
LISTE DES ECOLES : Poudlard; UdM; Polytechnique;
_____________________________________________
ENTREZ UN MOT A RECHERCHER (1er test)

les
Recherche pour le mot : les
Information sur le livre :
QA203. Calcul a plusieurs variables partie 1. 2011. 3 ans et plus. Auteur : Cay HORTSTMANN; Genre : Informatique
Information sur le livre :
QA204. Calcul a plusieurs variables partie 2. 2011. 3 ans et plus. Auteur : Cay HORTSTMANN; Genre : Informatique
Objet Numerique!  Taille : 1270 oct.  vvv.3Mousquetaire.com  Format : epub
Information sur le livre :
NUM392. Les 3 mousquetaires. 1844. 11 ans et plus. Auteur : A. Dumas; Genre : Roman
Objet Numerique!  Taille : 1270 oct.  vvv.3Mousquetaire.com  Format : epub
Information sur le livre :
NUM393. Les 3 mousquetaires 2. 1846. 11 ans et plus. Auteur : A. Dumas; Genre : Roman
_____________________________________________
ENTREZ UN MOT A RECHERCHER (2e test)

epub
Recherche pour le mot : epub
Objet Numerique!  Taille : 1270 oct.  vvv.3Mousquetaire.com  Format : epub
Information sur le livre :
NUM392. Les 3 mousquetaires. 1844. 11 ans et plus. Auteur : A. Dumas; Genre : Roman
Objet Numerique!  Taille : 1270 oct.  vvv.3Mousquetaire.com  Format : epub
Information sur le livre :
NUM393. Les 3 mousquetaires 2. 1846. 11 ans et plus. Auteur : A. Dumas; Genre : Roman
Press any key to continue . . .
```

Question

1. Dans ce TP, quelles classes avez-vous définies comme abstraites ? Expliquez pourquoi.
2. Pour les classes *ObjetEmpruntable* et *ObjetNumerique*, quelles méthodes avez-vous choisies comme virtuelles ? Expliquez pourquoi.
3. Pourquoi est-ce que nous devons utiliser des *dynamic_cast* pour l'opérateur<< ?

La correction du TP2 se fera sur 20 points.

Voici les détails de la correction :

- (03 points) Compilation du programme;
- (03 points) Exécution du programme;
- (04 points) Comportement exact des méthodes du programme;
- (04 points) Utilisation adéquate du polymorphisme;
- (02 points) Gestion correcte de la mémoire;
- (01 point) Documentation du code;
- (01 point) Utilisation correcte du mot-clé const et this;
- (02 points) Réponses aux questions.