INF1010 Programmation Orientée-Objet

Travail pratique #1 : Allocation dynamique, composition et agrégation

Objectifs :	Permettre à l'étudiant de se familiariser avec les notions de base de la programmation orientée-objet, l'allocation dynamique ainsi que les notions de composition et d'agrégation
Remise du travail :	28 janvier 2013 à 10:00
Références :	Notes de cours sur Moodle & Chapitre 5 du livre Big C++
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 et les commentaires sont obligatoires. Les travaux dirigés s'effectuent obligatoirement en équipe de 2 personnes faisant partie du même groupe. Vous devez obligatoirement suivre le guide codage

Informations préalables

La directive de précompilation « #ifndef »

La directive de précompilation « #ifndef » signifie « if not defined » (si non défini). Comme le type de directive le laisse deviner, cette directive est évaluée avant la phase de compilation du code source. Dans les travaux pratiques, vous l'utiliserez dans les fichiers d'en-têtes (.h), pour éviter la double inclusion. Par exemple, un fichier peut inclure deux fichiers d'entête, soit a.h et b.h. Cependant, il se peut que b.h inclure lui-même le fichier a.h. On se retrouve donc à inclure deux fois le fichier a.h, ce qui entrainerait une erreur de compilation car on ne peut définir deux fois la même classe. La directive #ifndef nous évite donc cette double inclusion. Pour utiliser la directive *« #ifndef »*, il faut respecter la syntaxe suivante :

```
#ifndef NOMCLASSE_H
#define NOMCLASSE_H
// Définir la classe NomClasse ici
#endif
```

La directive de précompilation « #include »

La directive de précompilation pour l'inclusion de fichiers *« #include »* peut prendre 2 formes :

1. #include <nom_fichier>
2. #include"nom_fichier"

Ce qui les différencie est en fait l'emplacement où le fichier spécifié est recherché. Pour la seconde forme, le précompilateur commence tout d'abord par rechercher dans le même répertoire que le fichier compilé. Par la suite, il procède de la même manière que la première forme, c'est-à-dire normalement dans des répertoires prédéfinis par l'environnement de développement intégré.
En résumé, généralement, lorsqu'on inclut un fichier source qui se trouve dans le projet, on utilise la seconde forme. Lorsqu'au contraire on inclut un fichier qui provient d'une bibliothèque externe au projet, on utilise la première forme.

Mise en contexte

Vous voulez classer vos films en Coffrets dans une vidéothèque. Ce classement vous permettra de mieux répertorier tous vos films par leur titre, leur description, leur date d'acquisition etc. L'organisation sera donc faite de la façon suivante : chaque coffret contient plusieurs films que vous pouvez organiser selon un critère prédéfini. Par exemple la saga Twilight(1-5) peut être contenue dans un seul coffret. Votre organisation sera faite de façon à accéder aux informations d'un film, et aussi à celles d'un coffret. C'est dans la classe Videotheque que seront contenus les différents coffrets.

Travail à faire

Vous devez réaliser la définition et l'implémentation des classes Date, Film, Coffret et Videotheque. Certains fichiers d'en-tête (.h) vous seront fournis. Vous devrez implémenter les méthodes dans les fichiers sources (.cpp) correspondants.

Classe Date

Créer une classe « Date » minimaliste représentant une date du calendrier (jour, mois, année), sans se préoccuper ici des valeurs illicites pour les champs (c'est-à-dire que vous n'avez pas à tester si la valeur du jour ne dépasse pas 31 et celle du mois 12 [30/12/2010 et 33/13/2010 seront acceptées]).

- Attributs de type entier non signé jour, mois et année
- Constructeur par défaut
- Constructeur par paramètre
- Destructeur
- Méthode de modification
- Méthode d'accès
- Méthode d'affichage de la date sous la forme: jour/mois/année

Classe Film

Un film est caractérisé par un numéro qui sera considéré comme son identifiant unique, son titre, sa description, sa durée en minutes et sa date d'acquisition. Vous devrez implémenter les méthodes suivantes dans la classe Film :

- Un constructeur par défaut
- Un constructeur par paramètres
- Un destructeur
- Les méthodes d'accès et de modification aux variables membres
- Une méthode qui affiche les informations du film (assurez-vous que l'affichage des informations soit fait de façon intelligible).

Classe Coffret

Un coffret est caractérisé par un identifiant, un titre et sa date de création. Le nombre de film dans le coffret n'est pas limité, car nous utiliserons un tableau dynamique pour représenter le coffret. Par souci de simplification, vous construirez des coffrets d'au plus 6 films. Vous devrez implémenter les méthodes suivantes dans la classe Coffret :

- Un constructeur par défaut
- Un constructeur par paramètres
- Un destructeur

- Les méthodes d'accès et de modification aux variables membres
- Une méthode qui ajoute un film au coffret en respectant la contrainte selon laquelle le même film ne peut paraître deux fois dans le même coffret. Cette contrainte sera respectée en testant l'identifiant et le titre du film.
- Une méthode qui affiche le numéro, le titre, la date de création du coffret ainsi que les informations de chaque film de ce coffret (affichage de façon intelligible)

Classe Videotheque

La vidéothèque permet de stocker les coffrets. Elle est représentée par un tableau dynamique dont la capacité est automatiquement doublée si celui-ci est plein. Concrètement, cela revient à créer un second tableau de capacité double du précédent et toutes les informations contenues dans l'ancien tableau sont copiées dans le nouveau. Votre classe Videotheque devra donc implémenter les méthodes suivantes :

- Un constructeur par défaut
- Un destructeur
- Une méthode qui ajoute un coffret à la vidéothèque (on peut y mettre autant de coffrets que voulu)
- Une méthode qui retourne le nombre de coffret de la vidéothèque
- Une méthode qui affiche les titres et les informations des films de chaque coffret de la vidéothèque (affichage de façon intelligible).

Main.cpp

Suivez les instructions fournies.

Question de cours :

Quelle est la relation qui existe entre la classe Date et la classe Film, ainsi qu'entre la classe Coffret et la classe Videotheque?

Correction

La correction du tp1 sera sur 20 points. Voici les détails de la correction :

- (10 points) Compilation & exécution exactes des différentes méthodes.
- (02 points) Documentation du code source.
- (02 points) Utilisation correcte du mot clé const aux endroits appropriés.
- (02 points) Utilisation adéquate des directives de précompilation.
- (01 points) Utilisation adéquate des fonctions membres dans la fonction principale.
- (01 points) Réponse aux questions de cours.
- (02 points) Allocation et désallocation appropriées de la mémoire.