

Rapport Projet INF224

Haj Salah Nada

Question 4:

Comme pour la fonction d'affichage, la fonction pour jouer l'objet ne modifie pas l'objet et elle doit être déclarée dans les classes Photo et Video et dans la classe de base afin de permettre un appel polymorphique sur la hiérarchie de classes. Cependant, contrairement à la fonction d'affichage, elle ne peut pas avoir d'implémentation au niveau de la classe de base (car a priori chaque type d'objet nécessite un utilitaire différent pour être joué). Comment appelle-t-on ce type de méthode et comment faut-il les déclarer ?

- Le type de ce méthode: une méthode abstraite ou virtuelle pure
- Declaration: déclarée dans le .h comme "virtual" et avec l'indication "= 0".

Si vous avez fait correctement ce qui précède, il ne sera plus possible d'instancier des objets de la classe de base. Pourquoi ?

- On ne peut pas instancier des objets car la classe de base déclare une fonction virtuelle pure donc c'est une classe abstraite, non instanciable.

Question 5:

Quelle est la propriété caractéristique de l'orienté objet qui permet de faire cela ?

- C'est le polymorphisme, Il permet de définir plusieurs comportement au même objet.

Qu'est-il spécifiquement nécessaire de faire dans le cas du C++? Quel est le type des éléments du tableau : est-ce que ce tableau contient les objets ou des pointeurs vers ces objets ? Pourquoi ? Comparer à Java.

- Pour utiliser le polymorphisme en C++ on doit utiliser des pointeurs (ou des références) et le mot-clé "virtual".

- Il faut allouer un tableau ou vector de pointeurs vers des objets Multimedia: Multimedia* m, car si la taille en mémoire des objets est inconnue ou bien non constante, ça permet la résolution dynamique,

En Java, il suffit de déclarer un tableau d'objets car java travaille en interne avec des références.

Question 6:

Que faut-il faire pour que l'objet Film ait plein contrôle sur ses données et que son tableau de durées des chapitres ne puisse pas être modifié (ou pire, détruit) à son insu ?

- On peut mettre les attributs de Film à private ou protected,
- Pour l'accesseur, il faut qu'il retourne un pointeur de type const int*.
- Pour setteron, il ne faut pas passer affecter le tableau tel qu'il est mais en faire une copie à l'objet et l'affecter à chaptersDurations.

Attention, le même problème se pose si un accesseur retourne directement ce tableau sans prendre les précautions nécessaires: la encore le contenu du tableau n'est pas récopié et l'appelant peut le modifier à sa guise. Quelle est la solution très simple que propose C/C++ pour éviter ce problème ?

-Pour l'accesseur, il faut qu'il retourne un pointeur de type const int*.

Question 7:

Contrairement à Java ou C#, C/C++ ne gère pas la mémoire dynamique automatiquement (*) : comme il n'y a pas de ramasse miettes, tout ce qui a été créé avec new doit être détruit avec delete sinon on aura des fuites mémoires. Parmi les classes précédemment écrites quelles sont celles qu'il faut modifier afin qu'il n'y ait pas de fuite mémoire quand on détruit leurs instances ?

-La mémoire qui n'est pas gérée automatiquement par c++ est la mémoire dynamique(on utilise new).La classe Film a besoin d'un destructeur qui libérera la mémoire allouée avant la destruction de l'objet.

De même, la copie d'objets peut poser problème dans certains cas. Pourquoi et que faudrait-il faire ?

- ça pose un problème lorsque l'objet possède un pointeur comme attribut. En effet , lorsqu'on veut faire la copie des valeurs d'une instance à une autre, le pointeur est copié mais le pointé devient partagé par les deux instances). Dans ce cas, il faudrait empêcher la copie d'objets sinon la gérer en redéfinissant le constructeur par recopie et l'opérateur =.

Question8:

On rappelle aussi que la liste d'objets doit en fait être une liste de pointeurs d'objets. Pourquoi ? Comparer à Java.

- On utilise des pointeurs dans la liste afin d'utiliser le polymorphisme sur les objets du groupe.
- Dans java, le polymorphisme fonctionne correctement juste avec une liste d'objets car les références sont gérées en interne.

Question 10:

Les méthodes précédentes permettent d'assurer la cohérence de la base de données ar quand on crée un objet on l'ajoute à la table adéquate. Par contre, ce ne sera pas le cas si on crée un objet directement avec new(il n'appartiendra à aucune table). Comment peut-on l'interdire, afin que seule la classe servant à manipuler les objets puisse en créer de nouveaux ?

- En c++, on peut utiliser du mot-clé friend.

Question 11 :

Votre méthode processRequest() devra pouvoir accéder aux données de la classe créée à la question précédente. Sachant que cette méthode peut appartenir à n'importe quelle classe, quelle est la solution la plus simple ?

- La méthode la plus simple est d'ajouter la méthode processRequest à la classe Manager.

