INF2

Jérôme Arn

19.02.2019

Table des matières

1 Les classes

les classes sont des types complexe qui permettent de créer des instances. Les fonctions membres sont des méthodes. Elles sont constituées de deux parties : une partie public et une partie privée. Par défaut, si une déclaration est fait avant la partie privée, elle est considérée comme privée.

Une variable de type <Class> s'tuilise comme n'importe quel autre type. De ce fait on peut la passer paramètre ou l'utiliser comme constante. On accède aux membres de la classe par la notation pointée. Mais on ne peut pas accéder directement aux membres privés en dehors de la définition des méthodes. Le seul moyen pour obtenir leur valeur est d'utilisé des geter. On peut aussi utiliser des seter pour les modifier.

On peut accéder aux membres privés de deux manières, par leur nom ou le mot clé ́this->́qui est un pointeur vers l'objet et *thisĺ'objet lui même. Les fonctions membres peuvent être définies de deux manières. En ligne, c'est à dire directement dans la déclaration. Séparément, avec l'opérateur de résolution de portée <Class> : :. Ces dernières peuvent être notifiées avec un const à la fin de la ligne pour dire qu'aucunes donnée membre n'est modifiée par la-dite fonction. Si une fonction membre ne possède pas le const, elle ne pourra pas être appelé pour une constante de type <Class>.

2 Constructeurs

Un constructeur est une fonctio membre qui a le même nom que la classe, qui ne retourne pas de valeur et qui ne possède pas de type de retour et donc pas de return. Il est possible de surcharger les constructeurs. Un constructeur sans argument est appelé constructeur par défaut. Si une classe que nous avons déclarer n'en possède pas, le compilateur ajoute celui par défaut. Maisdans le cas contraire, le constructeur par défaut n'est plus du tout ajouté par le compilateur. Si on veut déclarer des objets sans arguments, il faut le redéclarer. Dans des cas spéciaux, il est aussi possible de l'interdire. L'initialisation par liste permet d'éviter d'avoir l'appel du constructeur vide ET l'appel du constructeur que nous avons déclarer. Car en effet lors de l'initialisation par affectation, le constructeur sans argument est appelé et ensuite le constructeur, que nous avons déclaré, est appelé. Toutefois il est possibe d'initialisé directement les membres lors de leurs déclarations.

3 Compilation séparée

Généralement la déclaration d'une classe se fait dans des fichiers séparés. Le header pour les déclarations de données et des fonctions membres et amies ainsi que la définition de certaines fonctions en ligne. Le cpp inclut la définition des autres fonctions membre ainsi que l'initialisation des variables statiques. Il est préférable de ne pas utiliser le namespace standard pour les fichiers de classe.

4 Surcharge d'opérateurs

Pour chaque classe différentes il faut effectuer une surcharge d'opérateur pour faire des additions, des soustractions, ou juste un affichage. Ces dernières sont des fonctions membres dont la syntaxe diffère d'un opérateur à l'autre. Si on veut faire une surcharge d'opérateur pour la multiplication d'un entier avec un vector, il faut déclarer la fonction comme amie. La déclaration se fera dans la classe. Lorsqu'un opérateur est commutatif, il est parfois nécessaire de déclarer une fonction qui définit l'opération. Et par la suite de déclarer une autre fonction qui appelle cette dernière. L'opérateur d'affectation doit être une fonction membre. Ce dernier retourne une référence sur l'objet qu'il affecte. Finalement pour la surcharge d'opérateur, il faut utiliser une fonction non-membre uniquement si le premier argument est vient d'une classe qu'on ne peut modifier.

5 Membres constants ou statiques

La déclaration d'une variable statique se fait dans le fichier header mais en dehors de la déclaration de la classe. Une fonction membre peut aussi être déclarée static. Cela implique qu'elle n'a pas accès aux membres non static, elle ne s'applique pas à un objet spécifique. Pour y accéder il faut l'opérateur de portée.

6 Membres particuliers

Les fonctions membres suivantes sont définies implicitement par le compilateur :constructeur par défaut, destructeur, constructeur de copie, opérateur d'affectation, constructeur de déplacement et opérateur de déplacement. Si et seulement si la classe n'en possède pas d'autre.

7 Généricité

Pour écrire une fonction générique, on rajoute Template <liste de paramètres> déclaration en dessus de la définition de la fonction. On peut déclarer des fonctions, des classes et des variables génériques.