

Exercice 2 – Classes et pointeurs en C++

5 septembre 2017

Préparé par
Benjamin Lemelin

1 Travail à effectuer

Ouvrez le projet Visual Studio fourni avec cet énoncé. Ce projet contient plusieurs fichiers « **.h** » et « **.cpp** » à compléter.

Notez que les défis de cet exercice ne sont pas à prendre à la légère. Ne vous lancez pas dedans si vous ne savez pas comment vous y prendre ou si vous n'avez pas le temps.

1.1 Créer une classe représentant un tableau

En C#, il est possible de savoir la taille d'un tableau. Cependant, en C++, ce n'est pas possible. Créez donc une classe servant à représenter un tableau dynamique. Voici d'ailleurs sa structure :

IntArray
- array : int* - size : unsigned int
+IntArray(size : unsigned int) +get(index : unsigned int) : int +set(index: unsigned int, value : int) : void +size() : unsigned int

Faites un fichier « **.h** » et un fichier « **.cpp** ». Ne pensez pas au « **delete** » de votre tableau dynamique.

1.2 Créer le destructeur de la classe « IntArray »

En C++, il existe l'inverse des constructeurs : le **destructeur**. Le destructeur est appelé lorsque l'objet est détruit, soit parce qu'il l'a été explicitement (un appel à « **delete** »), soit automatiquement à la fin d'une fonction (en lien avec la mémoire statique).

IntArray
- array : int* - size : unsigned int
+IntArray(size : unsigned int) +~IntArray() +get(index : unsigned int) : int +set(index: unsigned int, value : int) : void +size() : unsigned int

Ajoutez un destructeur à votre classe, sachant que les destructeurs ne prennent aucun paramètre et débutent par un « **~** ».

Questions méritant réflexion :

- Que contient votre constructeur ? Sachant cela, que devrait faire votre destructeur ?

1.3 Modifier la classe « Console »

Ouvrez la classe « `Console` ». Modifiez cette classe de sorte que la méthode « `readIntArray` » retourne un « `IntArray` » au lieu d'un « `int*` ». Ne retournez pas un « `IntArray*` », mais bien un « `IntArray` ».

1.4 Lire un tableau d'entiers à partir de la console (Version 1)

Dans votre fonction « `main` », utilisez la fonction « `readIntArray` » pour lire un tableau d'entiers. La taille du tableau doit être choisie par l'utilisateur. Demandez donc la taille du tableau en premier.

```
Combien de nombres?  
2  
Entrez un nombre :  
5  
Entrez un nombre :  
8  
Appuyez sur une touche pour terminer le programme...
```

Questions méritant réflexion :

- Placez un « `std::cout` » dans le constructeur de « `IntArray` », tout simplement pour être notifié de sa construction. Combien de fois sera-t-il appelé selon vous ?

1.5 Créer un constructeur de copie de la classe « `IntArray` »

Le C++ est un langage où tout est passé « par valeur » (paramètres et retours de fonction). Toute variable sur la pile doit donc être copiée avant d'être transférée d'une fonction à une autre, et cela inclut aussi les objets créés sur la pile (comme vous avez pu le constater avec la fonction « `readIntArray` »).

Si votre fonction « `readIntArray` » avait retourné un « `IntArray*` », cela aurait été différent, car ce n'est pas l'objet qui est copié, mais bien le pointeur (l'adresse) vers l'objet qui est copié.

Par défaut, la copie d'un objet consiste en une simple copie de ses attributs vers le nouvel objet. Dans votre cas, cela implique que deux objets « `IntArray` » pourraient utiliser le même tableau (le même « `int*` »), car c'est l'adresse du tableau qui est copié et non pas le contenu du tableau.

IntArray
- array : int*
- size : unsigned int

+IntArray(size : unsigned int)
+IntArray(&IntArray : IntArray)
+~IntArray()
+get(index : unsigned int) : int
+set(index: unsigned int, value : int) : void
+size() : unsigned int

Ajoutez un constructeur de copie à votre classe. Faites une copie complète des valeurs, et non pas une simple copie de l'adresse du tableau. *Notez que le constructeur de copie est appelé automatiquement.*

4 Modalités de remise

Remettez le projet Visual Studio sur LÉA à l'heure et à la date indiquée par votre professeur. Faites attention à ne pas remettre de fichiers inutiles.