

Corrigé du TP5

Disclaimer : Ce corrigé n'a pas pour but de vous fournir un détail des algorithmes répondant aux questions de l'énoncé, mais de vous résumer les notions clés dont vous aurez besoin pour résoudre ces exercices, les avertissements vis à vis de certaines erreurs classiques et les détails des passages les plus techniques, voire des précisions si vous souhaitez aller plus loin sur ces questions et plus généralement de synthétiser dans un même document la plupart des commentaires fait en TP afin qu'ils puissent profiter à tous. Pour obtenir un corrigé détaillé, il vous faudra donc vous référer à ce que vous avez pu faire en TP ou reprendre vous même les exercices à partir des notions présentées dans le présent corrigé pour écrire les algorithmes qui vous manquent.

En cas de difficultés, **n'hésitez pas à me contacter**, nous pourrions en discuter soit par mail soit en début de séance suivante. Ce corrigé n'a pour but que d'être une base de travail pour vous et de servir de complément par rapport à ce que vous avez fait en séance, il n'est en aucun cas fait pour se suffire à lui même.

Définition d'une structure

- * Une structure est un type de variables constituées de plusieurs attributs (ou champs). Ces attributs peuvent être de type différent, contrairement aux tableaux ou vecteurs.

Généralement, pour définir une structure, on utilise trois fichiers que l'on appelle nomStructure.h (un header file) , nomStructure.cpp et le fichier main.cpp que vous utilisez habituellement.

- * Pour créer le fichier .h il vous faut, lorsque vous créer un nouveau fichier, sélectionner le C/C++ header. Dans ce fichier doit figurer trois choses entre la deuxième et troisième de code déjà présents dans le fichier à sa création :

Les bibliothèques que vous allez utiliser.

La définition de la structure avec la syntaxe suivante :

```
struct nomStructure
{
    type    nomAttribut;
};
```

N'oubliez pas les ;

Et enfin la liste des entêtes (ou prototype) des fonctions. Ce n'est pas ici que vous devez les définir. Pour rappel, un entête de fonction est de la forme :

```
typeSortie nomFonction(typeArgument nomArgument);
```

N'oubliez pas le ; à la fin des entêtes.

- * Pour créer le fichier .cpp il vous faut, lorsque vous créer un nouveau fichier, sélectionner le C/C++ source. Dans ce fichier, vous définirez vos fonctions.

Attention à bien écrire **#include "Nobel.h"** au début de votre programme, sinon cela ne marchera pas.

- * Pour que votre fonction main reconnaisse votre structure, vous devez écrire **#include "Nobel.h"** au début de votre programme.

Attention, vous n'avez pas besoin de mentionner le fichier .cpp , seulement le .h

Vous pouvez alors utiliser les fonctions que vous avez définies comme n'importe quel fonction habituelle.

Utilisation d'une structure

- * Pour définir un objet d'une structure, vous pouvez le faire comme pour n'importe quel objet, le type étant le nomStructure :

```
nomStructure nomVariable;
```

- * Pour accéder à un attribut d'une variable de la structure, vous pouvez utiliser la syntaxe suivante :

```
nomVariable.nomAttribut
```

- * Il est possible de donner le même nom à des fonctions faisant des choses différentes. On dit alors que l'on surcharge les fonctions.

L'intérêt est de pouvoir appliquer une même fonction à des variables de type différent ou à un nombre différent de variables.

Boucles avec objet de type structure

- * Un objet d'une structure se manipule comme un objet classique, cependant, les seules fonctions que vous pouvez utiliser sur l'objet sont celles que vous avez définies.
- * Vous pouvez également manipuler les constituants de ces objets par une syntaxe du type `objet.nom_partie`. Vous pouvez utiliser ces composants comme n'importe quel élément du type auquel ils appartiennent.

Utilisation d'un menu

Lorsque l'on vous demande un menu, on s'attend à ce que, tant que l'utilisateur de l'a pas quitté, le programme lui propose de faire un nouveau choix parmi ceux proposés.

On attend donc de vous que vous fassiez une boucle.

Quelques conseils pour retravailler le TP

- * Il est important que vous retravailliez ce TP qui est une introduction à la notion plus complexe de classe que vous verrez plus tard dans le cours.

Il faut notamment bien comprendre ce qui doit figurer ou non dans les fichiers `.h` et `.cpp`, la syntaxe qui est associé, où sont définies les fonctions, quel fichier doit appeler lequel, etc ...

- * L'utilisation des structures peut sembler assez contre intuitive, ou en tous les cas, elle est assez différente de ce que vous avez pu faire en programmation jusque là.

Vous verrez plus tard l'intérêt de cette façon de faire, en attendant, essayez de vous familiariser avec cette approche.