# TD Informatique n$^o$1

Formation d'ingénieurs de l'Institut Galilée MACS-2
Septembre 2007

## 1 De C à C++

Ce premier TD a pour objectif de familiariser l'étudiant avec les nouveautés apportées par le C++. Nous laissons donc de côté dans un premier temps tout ce qui est lié à la méthodologie ou programmation objet.

Ce TD est prévu pour une séance avec un compte-rendu et les sources à rendre au prochain TD.

## 2 Création d'une bibliothèque gérant des vecteurs

Pour ce TD ( et les suivant ), nous allons créer une "mini" bibliothèque d'algébre linaire. Dans un premier temps, nous allons nous intéresser à la gestion d'un vecteur de dimension $n$.

Le "cahier des charges" pour les vecteurs consiste Ã  ce qu'on puisse appeler les fonctions suivantes :

- une fonction `newVector` qui prends en argument la dimension du vecteur et renvoie un nouveau vecteur de dimension $n$ que la fonction aura créée ;
- Une fonction `getDim` qui renvoie la dimension d'un vecteur passé en argument ;
- Une fonction `deleteVector` permettant de détruire un vecteur et libérer la mémoire qu'il occupait ;
- Une fonction `copyVector` qui recopie un vecteur passé en argument dans un vecteur passé également en argument. Les dimensions initiales des deux vecteurs peuvent être différentes ;
- Une fonction `cloneVector` permettant de créer un nouveau vecteur qui soit la copie du vecteur passé en argument ;
- Un opérateur qui additionne deux vecteurs et renvoie le résultat sous forme d'un vecteur ;
- Idem mais qui soustrait deux vecteurs ;
- Un opérateur qui calcule le produit scalaire de deux vecteurs ;
- Une fonction qui calcule la norme d'un vecteur ;
- Un opérateur faisant l'homothétie d'un vecteur par un réel ( et transforme ce vecteur : pas de création de nouveaux vecteurs ) ;
- Des opérateurs de flux permettant d'écrire ou de lire un vecteur.

# 3 A faire

## 3.1 Spécifier

Dans un premier temps, on cherchera les fonctions et les structures nécessaires pour réaliser le cahier des charges. On répondra donc à la question : "*Que dois-je faire avec mon vecteur pour répondre au cahier des charges ?*"

On se posera en particuliers les questions suivantes :

- Quels sont les manipulations non précisées mais inhérent au cahier des charges que je dois effectuer sur mon vecteur ( routine de copie, de tests, etc. ) ;
- Quels sont les paramêtres que je dois passer pour chaque routine que je dois écrire ;
- Que dois-je retourner à la fin de ma fonction ( `void` si il n'y a rien à retourner ) ;
- Quelles conditions doivent vérifier le maillage et les autres paramêtres d'entrés ï¿½ l'entrée de la routine ( précondition ) ;
- Quels sont les paramêtres d'entrée qui ne doivent pas être modifiés à la sortie de la routine ( les invariants ) ;
- Quelles conditions doivent vérifier en sortie de routine les paramêtres en entrées qui sont modifiés par la routine ainsi que la valeur retournée ( si on doit retourner une valeur ) ( postcondition ) ;

On obtiendra donc une liste de fonctions définies par leurs signatures ( nom – arguments – nombre – type – valeur de retour éventuel ) qu'on mettra dans un tableau avec les préconditions, les invariants et les postconditions associés à chaque routine sous le format suivant :

| `Nom de la fonction` | Arguments ( nom – type ) | Type valeur retournée | Préconditions | Invariants | Postconditions |
|---|---|---|---|---|---|
| | | | | | |

## 3.2 Mise en œuvre

On choisira la structure la mieux adaptée à notre problï¿½me. Puis, à l'aide d'un fichier d'entète ( extension .hpp ), d'un fichier mettant en œuvre les corps des fonctions ( extension .cpp ) et d'un `Makefile`, on produira une petite bibliothèque réutilisable dont chaque fonction sera testée dans un petit programme annexe ( en dehors du programme principal qui doit être écrit à part ).

## 3.3 Tester

Ecrire un petit programme de test où on s'assurera que chaque fonction écrite fonctionne bien.

A votre avis, quel(s) problème(s) rencontre-t'on avec la ligne de code suivante :

```
Vecteur t = u + v + w
```

Comment peut-on résoudre dans la limite de nos connaissances actuels le(s) problï¿½me(s) rencontré ?

## 3.4 Bilan

On donnera au TD suivant, outre le TD sous forme source ( accompagné de son Makefile ) un document électronique ( Word, LaTeX, etc. ) ou manuscrit contenant une partie pour la spécification et une autre partie faisant le bilan des difficultés rencontrées, de la façon dont elles ont été résolues et des parties du C++ vues dans ce TD qui vous semblent encore obscures.