

Travaux dirigés de 2^{ème} année Sujet TD n°3

(Version 3.3 – 8/11/2005)

Objectifs

On aborde dans ce TD la surcharge d'opérateurs, ainsi que la surcharge de constructeurs, et l'allocation dynamique de tableaux.

Introduction

Dans ce TD on cherche à mettre en œuvre une classe Vecteur (au sens vecteur mathématique) de taille quelconque et les opérations mathématiques associées. Attention, cela n'a pas directement de rapport avec la classe générique `vector<T>` proposée par la STL (Standard Template Library)

1) modélisation avec UML

Représenter avec les diagrammes de classe, la classe Vecteur (attributs seulement).

2) Constructeurs et destructeur

Passez en C++. Déclarez tous les constructeurs utiles :

- Constructeur sans paramètre
- Constructeur à un paramètre : taille du vecteur
- Constructeur à 2 paramètres : taille du vecteur et un tableau de double pour l'initialiser

Montrez que ces 3 constructeurs peuvent être réduits en un seul en utilisant des paramètres avec valeur par défaut. Ecrivez la définition complète de ce constructeur. Ecrivez également :

- le constructeur par copie,
- le destructeur.

Note : Ajoutez des traces dans les constructeurs et le destructeur pour mettre en évidence les mécanismes d'appels automatiques lors des passages de paramètres par valeur et de retour de fonction de valeurs.

3) Somme

Ecriture d'une fonction réalisant la somme de 2 vecteurs (s'ils ont la même taille). Examiner les 2 cas :

- Fonction membre
- Fonction amie

Dans un deuxième temps, réaliser la somme en surchargeant l'opérateur +.

4) Autres opérateurs et méthodes

- Surcharge de l'opérateur =
- Surcharge de l'opérateur []
- Surcharge de l'opérateur << pour afficher la valeur d'un vecteur.
- Opérateur * (produit scalaire de 2 vecteurs)
- Opérateur * (produit d'un réel par un vecteur)
- Méthode retournant la norme d'un vecteur
- Méthode normalisant le vecteur
- Fonction calculant le cosinus de 2 vecteurs (c'est le produit scalaire de 2 vecteurs normalisés)

5) Programme principal

Faire un programme principal `td3.cc` mettant en œuvre des vecteurs :

- Créer des variables en utilisant tous les constructeurs possibles.
- Utiliser l'opérateur [] pour modifier des valeurs.
- Faire des affectations de vecteurs
- Utiliser chacune des méthodes et fonctions développées