

TP 11 : Surcharges d'opérateurs, templates

1 Surcharge d'opérateurs

1. Créer une classe contenant des tableaux à de réels à deux dimensions (de type `vector`) en données privées.
2. Créer le constructeur par défaut et le destructeur pour cette classe.
3. Créer un constructeur initialisant une instance de la classe avec des dimensions données et une valeur constante pour les coefficients du tableau. Ces paramètres seront passés en argument du constructeur.
4. Créer une surcharge de l'opérateur `[]` qui permet d'afficher un indice donné du tableau contenu dans une instance de la classe.
5. Créer une surcharge de l'opérateur `+`, telle que si **A** et **B** sont deux instances de la classe, de dimensions respectives n_A, m_A et n_B, m_B , alors **A+B** a pour données un tableau de dimensions $\max(n_A, n_B), \max(m_A, m_B)$ et ses coefficients contiennent la somme des coefficients de **A** et **B**, avec pour convention que si un coefficient n'existe pas, il vaut 0.
6. Créer une surcharge de l'opérateur `<<` qui permet d'afficher le tableau (contenu dans une instance de la classe) tout entier.

2 Création d'une classe template

Remarque : pour créer une instance d'une classe template, au moment d'appeler le constructeur, il faut bien préciser le type du paramètre T (du typename) entre des chevrons.

1. Créer une classe template dont la donnée (privée) est un tableau (vector) d'entier ou de réels.
2. Définir le constructeur par défaut pour cette classe, et un autre constructeur qui prend en paramètre un entier qui est la taille du vector, et un nombre, entier ou réel, qui est la valeur avec laquelle sont initialisés tous les coefficients du tableau.
3. Créer une fonction qui affiche les valeurs d'une instance de la classe.
4. Créer une fonction qui prend en argument un entier `ind` et un nombre `x`, entier ou réel, et modifie l'instance de la classe à laquelle elle est appliquée afin que le coefficient d'indice `ind` prenne la valeur `x`.
5. Créer dans le programme principal deux instances de la classe, l'une contenant des entiers et l'autre des réels, et tester les fonctions précédentes avec ces instances.