TD/TP 1 C++

Polytech' Paris-Sud
Et4 info
Aleksandr Setkov et Marc Evrard

9 janvier 2015

1 Parie 1 : Nombres complexes

1. Créer une structure qui stocke un nombre complexe `Complex` sous la forme de deux réels (`double`)
2. Implémenter une fonction `add` qui reçoit deux nombres complexes, fait une addition et retourne le résultat
3. Modifier la fonction en ajoutant un paramètre `bool printRes` pour choisir d'afficher ou de ne pas afficher le résultat. Donner `false` comme valeur par défaut pour ce paramètre
4. Implémenter une fonction `add` qui reçoit trois nombres complexes et stocke le résultat dans le troisième
5. Implémenter une fonction `main` pour tester les fonctions précédentes

2 Partie 2 : Classe Vector

1. Développer une classe `Vector` qui permet de manipuler un tableau de type `int` :

Nom	Type
`size`	`int`
`v`	`int *`

   (a) Déclarer la classe avec ses membres et ses méthodes
   (b) Implémenter les méthodes de la classe
   (c) Allouer la mémoire en utilisant `new`
   (d) Contrôler les indices pour éviter d'accéder à la partie de la mémoire non-allouée
   (e) Libérer la mémoire avec le destructeur

2. Implémenter `main()` pour utiliser la classe

3. Implémenter `inline` la fonction `isInLimits` qui vérifie que **l'indice** est bien compris entre les limites `lim1` et `lim2`. On utilisera cette fonction dans les méthodes de la classe.

4. Implémenter une **méthode statique** pour calculer `globale`, le nombre d'éléments de tous les vecteurs.

3 Partie 3 : Classe Matrix

1. Implémenter une classe `Matrix` qui représente une matrice $n \times m$

Nom	**Type**
`n`	`const int`
`m`	`const int`
`M`	`std::vector< std::vector<int> >`

2. Implémenter les méthodes `getElement` et `setElement`

3. Faire une fonction **amie** `mulMatVec` qui multiplie une matrice par un vecteur :

```
int mulMatVec(const Matrix &_M, const Vector &v, Matrix &res)
```