

# Projet C++ : Question Individuelle 09

Mathieu Boutolleau – Paul Fayoux

---

## 1 – Choix de réalisation, mise en œuvre des pratiques de Génie Logiciel :

- Gestion des versions des sources,
- Mise en place de tests unitaires,
- Écriture de commentaires pertinents,
- Respect d'une convention de nommage,
- Gestion de types et fonctions génériques,
- Levage d'exception lors de situation anormal mais prévisible de fonctionnement,
- Affichage d'avertissement lors de situation supposé incohérente mais non critique,
- Développement favorisant la maintenabilité et l'évolution de l'application.

## 2 – Diagramme de Classe : (voir annexe)

Nous repartons de notre projet sur la partie matrice et de ses 4 classes principales : `Ctable2D`, `Cmatrix`, `Cfile` et `Cexception`. Plus 1 classe main et 3 classes pour les tests unitaires.

Pour l'importation des matrices, nous utilisons une version améliorée de la classe `Cfile`. Elle reprend celle que nous avons pour la partie graphe. À savoir : elle traite un fichier possédant une certaine syntaxe :

`<nom de l'information><délimiteur><valeur>` (syntaxe des fichiers dans le sujet).

Cette classe permet de transformer le fichier lu en un tableau de chaîne de caractère qui contient les données du fichier. Grâce à un constructeur spécifique de `Ctable2D`, on peut créer des matrices via un fichier. Contrairement à l'ancienne version du programme l'export de matrice n'est plus possible.

En ce qui concerne le travail demandé pour la question mystère, nous avons ajouté une méthode qui effectue les calculs nécessaires pour le changement de base. Ainsi, nous avons ajouté le code nécessaire à l'appel de cette méthode au main et on a factorisé le code pour améliorer la lisibilité (déplacement du code faisant les calculs de la partie 1 du projet sur les matrices dans la méthode `computationOnMatrices`).

### 3 – Spécificités de l'application :

Nous avons ajouté à notre programme une fonctionnalité permettant à l'utilisateur de choisir entre faire les calculs sur les matrices, ou faire un changement de base pour un vecteur. Pour faire ce changement de base, l'utilisateur doit fournir en paramètre (dans cet ordre) le fichier contenant le vecteur et les matrices de changement de base.

Le vecteur doit être le premier argument passé en paramètre, les matrices de changement de base doivent ensuite être placées dans l'ordre des changements de bases successifs à faire.

Suite à la modification du code pour l'intégration de la fonctionnalité de changement de base :

- Nous avons corrigé un bug dans la fonction de multiplication de matrices,
- On affiche un avertissement si les matrices de changement de base ne sont pas carrées.
- Mais également si l'utilisateur ne passe pas en premier argument un vecteur qu'il souhaite exprimer dans une autre base.

4 – Annexes

