

**Rapport du projet de C++ (partie Graphes)**

**ETUDIANTS :**

Anthony XAYADETH

Ghali ATIF

**SUPERVISEUR :**

Pierre VENDE

**2022 - 2023**

**Sommaire**

Contenu

[I. Modèle UML réalisé 3](#_Toc135331306)

[Analyse globale du diagramme de classes réalisé 3](#_Toc135331307)

[II. Analyse détaillée des classes créées 5](#_Toc135331308)

[1. Classe CGraphe (a changer !!!!) 5](#_Toc135331309)

[2. Classe CSommet (a changer !!!!) 5](#_Toc135331310)

[3. Classe CArc (a changer !!!!) 6](#_Toc135331311)

[4. Classe CFichierGraphe (a changer !!!!) 6](#_Toc135331312)

[5. Classes CException, CAfficherException, CGestionGraphe et CGestionArc 7](#_Toc135331313)

[a. CException et CAfficherException (a changer !!!!) 7](#_Toc135331314)

[b. CGestionGraphe et CGestionArc (a changer !!!!) 7](#_Toc135331315)

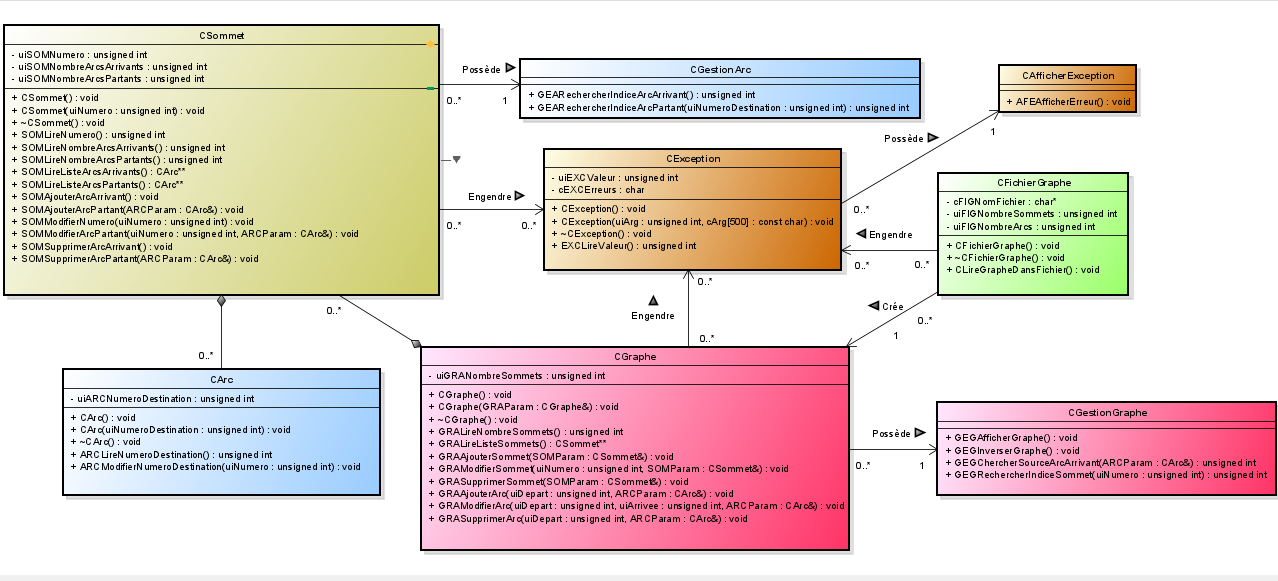
[III. Analyse sur la construction du code 8](#_Toc135331316)

[1. Les fichiers d’entête “.h” 8](#_Toc135331317)

[2. Les fichiers sources “.cpp” 8](#_Toc135331318)

# I. Modèle UML réalisé

## Analyse globale du diagramme de classes réalisé



Ce diagramme de classe se divise en 2 parties :

* Les classes principales : celles qui sont au cœur du logiciel de gestion de graphes.

🡪 **CGraphe**, **CSommet, CArc et CFichierGraphe**.

* Les classes secondaires : celles qui sont là pour compléter, séparer les entrées des sorties, ou corriger le fonctionnement des classes principales.

🡪 **CException** (pour corriger), **CAfficherException** (pour séparer les méthodes et l’affichage de **CException**), et **CGestionArc, CGestionGraphe** (pour compléter).

Explication du rôle des différentes classes principales :

* **CGraphe** : permet de construire et d’afficher le graphe, en s’occupant de l’ajout, de la modification et de la suppression des sommets qu’il contient,
* **CSommet** : permet de construire un sommet, et de gérer tous les arcs entrants/sortants qui interagissent avec le sommet,
* **CArc** : permet de construire un arc, avec un numéro de destination bien défini,
* **CFichierGraphe** : permet de manipuler un fichier texte « .txt » contenant les spécifications du graphe à créer.

Plusieurs remarques concernant les liens entre les classes :

* Pour des soucis de clarté et de lisibilité du diagramme, nous avons décidé de séparer les méthodes de **CSommet** et **CGraphe en 2 groupes** : d’un côté, les constructeurs, le destructeur, et les accesseurs (avec notamment les méthodes « Ajouter/Modifier/Supprimer »). De l’autre, les méthodes réalisant des traitements plus complexes ou l’affichage d’un résultat, ce qui explique la création de **CGestionArc** et **CGestionGraphe**.

Toutefois, **CGestionArc** et **CGestionGraphe ne sont pas déclarées et définies** (dans un .h et un .cpp) **à part de CSommet** et **CGraphe** respectivement. Par conséquent, puisque le code présent dans “CGraphe.cpp” contient, en plus de la définition de **CGraphe**, celle de **CGestionGraphe** (dans le fichier “CGestionGraphe.cpp”), un objet de la classe **CGraphe** possédera les méthodes de la classe **CGestionGraphe** (d’où l’association « Possède » entre les 2 classes).

Même raisonnement entre **CSommet** et **CGestionArc**.

* **CGraphe** est composé d’un tableau de **CSommet**, et chaque sommet contient 2 listes de **CArc** distincts : un pour les arcs arrivants, l’autre pour les arcs partants.
* Un objet de **CFichierGraphe** permet de créer un objet de **CGraphe**.
* **CGraphe**, **CSommet** et **CFichierGraphe** peuvent engendrer des exceptions (donc des instances de **CException**).
* **CGraphe** contient les entrées du graphe, tandis que **CAfficherGraphe** contient la sortie (autrement dit l’affichage) de ce dernier.

# II. Analyse détaillée des classes créées

## 1. Classe CGraphe (a changer !!!!)

Cette classe, héritant de **CListe**, est composée de 2 attributs supplémentaires :

* uiMATNombreColonnes : le nombre de colonnes de la matrice,
* uiMATNombreLignes : le nombre de lignes de la matrice.

Et de plusieurs méthodes :

* 2 constructeurs (par défaut et de recopie) et 1 destructeur,
* 1 méthode permettant de convertir un double en matrice, ce qui permettra de faire des produits entre une matrice et un réel,
* 2 accesseurs en lecture : “MATLireNombreColonnes” (pour lire “uiMATNombreColonnes”) et “MATLireNombreLignes” (pour lire “uiMATNombreLignes”),
* 6 méthodes pour gérer les colonnes et les lignes qui composent la matrice (“MATModifierColonne”, “MATAjouterColonne” et “MATSupprimerColonne” pour gérer les colonnes, “MATModifierLigne”, “MATAjouterLigne” et “MATSupprimerLigne” pour gérer les lignes).

## 2. Classe CSommet (a changer !!!!)

Elle est composée de 2 attributs :

* uiLISTaille : la taille de la liste,
* pTYMLISListe : la liste dynamique qui sera composée des éléments de la matrice.

Et de plusieurs méthodes :

* 2 constructeurs (par défaut et de recopie) et 1 destructeur,
* 1 méthode pour connaître la taille de la liste (“LISLireTaille”),
* 4 méthodes pour gérer les éléments présents dans la liste dynamique (“LISLireElement”, “LISModifierElement”, “LISAjouterElement” et “LISSupprimerElement”),
* 1 méthode pour vider la liste (“LISViderListe”), utile pour réinitialiser la matrice statique renvoyée par les surcharges d’opérateurs de **COperationMatrice**,
* 1 surcharge de l’opérateur [], pour définir son utilisation spécifiquement à l’utilisation d’une liste.

## 3. Classe CArc (a changer !!!!)

qfzf

## 4. Classe CFichierGraphe (a changer !!!!)

Cette classe possède 3 attributs :

* uiFIMNombreLignes : unsigned int
* uiFIMNombreColonnes : unsigned int
* pFIMFIMMatrice : TypeM\*\*

Le fichier texte, que manipule une instance de **CFichierMatrice**, contient :

* un nom de fichier,
* le type des éléments que contient la matrice à créer,
* le nombre de lignes et de colonnes de la matrice,
* les éléments de la matrice,

Pour lire correctement les fichiers, cette classe dispose :

* d'une fonction d'analyse syntaxique,
* de méthodes pour obtenir ou définir des éléments de la matrice.

Parmi les méthodes de **CFichierMatrice**, “FIMLireMatriceDansFichier” est d'une importance capitale pour le bon fonctionnement de la classe : elle permet de lire dans le fichier texte et d’extraire certaines informations utiles pour la construction de la matrice.

De plus, cette méthode vérifie d’abord si le fichier entré en argument est un fichier “.txt”, puis l’ouvre et lit ses informations. Cette fonction utilise plusieurs méthodes dont :

* FIMModifierDimensionLignes,
* FIMModifierDimensionColonnes.

**CFichierMatrice** dispose également de :

* FIMAfficherMatrice : affiche la matrice lue dans le fichier,
* FIMLireDimensionLignes : lit le nombre de lignes de la matrice, qui est écrit dans le fichier,
* FIMLireDimensionColonnes : lit le nombre de colonnes de la matrice, qui est écrit dans le fichier.

## 5. Classes CException, CAfficherException, CGestionGraphe et CGestionArc

### a. CException et CAfficherException (a changer !!!!)

La classe **CException** est composée de 2 attributs :

* uiEXCCodeErreurs : la valeur de l’exception levée,
* cEXCErreurs : contient le message d’erreur.

Ainsi que de quelques méthodes :

* 1 constructeur par défaut et 1 destructeur,
* 2 méthodes pour accéder en lecture à la valeur de l’exception (“EXCLireValeur”) et pour afficher le message d’erreur (“EXCAfficherErreur”).

Quant à **CAfficherException**, elle contient « AFEAfficherErreur », la méthode permettant d’afficher les codes et messages d’erreurs des exceptions levées.

### b. CGestionGraphe et CGestionArc (a changer !!!!)

- **CGestionGraphe**:est déclarée et définie dans la classe **CMatrice**. Elle possède 5 surcharges d’opérateurs (“=”, “+”, “-”, “\*” et “/”), qui permettent de définir les différentes opérations possibles entre matrices, et 2 méthodes particulières :

* OPMTransposerMatrice : pour transposer la matrice,
* OPMAfficherMatrice : pour afficher la matrice.

- **CGestionArc**:

# III. Analyse sur la construction du code

## 1. Les fichiers d’entête “.h”

Ce projet a nécessité l’utilisation de plusieurs fichiers d’entêtes pour déclarer les différentes classes utilisées, et 1 ou 2 fichiers de méthodes inlines “.ipp” (pour **CGraphe**, **CSommet**, **CFichierGraphe**, **CGestionGraphe** et **CGestionArc**) :

* CGraphe.h : contient « CGraphe.ipp », « CGestionGraphe.ipp », « CSommet.h » et la déclaration de **CGraphe**,
* CSommet.h : contient « CSommet.ipp », « CGestionArc.ipp », « CArc.h » et la déclaration de **CSommet**,
* CArc.h : contient « CException.h » et la déclaration de **CArc**,
* CFichierGraphe.h : contient « CException.h » et la déclaration de **CFichierGraphe**,
* CException.h : contient la déclaration de **CException**,

## 2. Les fichiers sources “.cpp”

Chacun des fichiers sources ci-dessous incluent le fichier d’entête au nom qui lui est lié :

* CGraphe.cpp : contient “CGraphe.h” et « CFichierGraphe.h »,
* CSommet.cpp : contient “CSommet.h”,
* CArc.cpp : contient « CArc.h »,
* CFichierGraphe.cpp : contient “CFichierGraphe.h”,
* CGestionGraphe.cpp : contient “CGraphe.h”
* CException.cpp : contient « CException.h »,
* CAfficherException.cpp : contient « CException.h »,
* ManipulationGraphes.cpp : contient la fonction principale du programme.