

POLYTECH TOURS

64 avenue Jean Portalis

37200 TOURS, FRANCE

Tél +33 (0)2 47 36 14 14

[www.polytech.univ-tours.fr](http://www.polytech.univ-tours.fr/)

**Projet C++ 2023/2024**

**Projet Graphe Orienté /**

**Etudiant :**

Yassine ELLATIFI

Milan HERGOTT

**Sommaire**

Table des matières

[1. Introduction 2](#_Toc164763301)

[2. Conception UML (ajouter classe inverse) 3](#_Toc164763302)

[2.1 Le diagramme de classe 3](#_Toc164763303)

[3. Review de code 4](#_Toc164763304)

[3.1 Classe CgrapheOriente 4](#_Toc164763305)

[3.2 Classe Cfichier 4](#_Toc164763306)

[3.3 Classe CSortie 4](#_Toc164763307)

# Introduction

Au cours du 6ème semestre de la 3ème année du cycle ingénieur Polytech. Il nous est demandé au sein du cours de Conception Orientée Objet 2 de réaliser une librairie de classe et de fonction permettant de manipuler des Graphes orienté et non orienté. Pour ce projet tutoré, il est nous également imposé de proposer une implémentation flexible pour permettre à l’utilisateur de spécifier lui-même les informations qu’il veut stocker.

Pour effectuer ce projet nous avons essayer au mieux de suivre les principes fondamentaux du génie logiciel tel que l’encapsulation des données ou le respect d’une convention de nommage. En poursuivant cette volonté nous avons dû dans un premiers temps, comprendre au mieux les attentes et objectif de ce projet a l’aide de diagrammes UML, seulement après avoir obtenu une compréhension et une visualisation suffisante du projet nous nous sommes lancé dans le développement des différentes classe et méthodes.

C’est pour cela que ce rapport est articulé en quatre parties de la manière suivant : dans un premier temps nous allons nous pencher sur la conception UML du projet ce qui pourrait s’apparenter a la partie descendant du cycle en V.

Ensuite nous allons effectuer une review de code non exhaustive afin de connaitre le fonctionnement des principales méthodes de ce projet, elle parlera également des tests sur les méthodes qui ont pu être effectué.

La troisième partie quant à elle est adressée directement à l’utilisateur, cette dernière permet de comprendre comment utiliser le programme de la manière qui est attendus par les développeurs.

Enfin pour finir nous allons conclure sur ce projet discuter de nos principales difficultés de ce qu’on pense de notre projet et enfin de si nous pensons avoir atteint les objectifs.

# Conception UML (ajouter classe inverse)

Pour notre conception UML nous nous somme limité a un diagramme de classe, pour ce projet nous avons pensse que la réalisation d’un digramme d’activité ou un digramme des cas d’utilissation n’était pas pertinent. D’apres nous un diagramme d’activité ici permet pleinement de comprendre le projet.

## Le diagramme de classe

Pour finir notre démarche de conception UML nous avons décidé de faire un diagramme de classe ce qui est d’une grande utilité pour la compréhension des relations entre les classes, et pour le passage au code.

Pour notre diagramme des classes nous avons essayer des suivre au mieux les « mantra » du génie logiciel spécifiquement celui de l’encapsulation des données et l’idée de contrat de service pour les classes.

En effet ici nous avons choisi de créer des classes séparées pour l’affichage du graphe, la lecture d’un graphe depuis un fichier et enfin pour inverser un Graphe. Tout cela pour respecter le principe de contrat de service de la classe CGrapheOriente.

Nous avons pris la liberté de modifier le type du nom des Sommet, en effet d’après le sujet le nom des sommets devait être un entier mais par volonté de rendre le code réutilisable nous avons élargi cette perspective en permettant de donner des chaine de caractère.

Une image contenant capture d’écran, ligne, conception

Description générée automatiquement

Figure 1 : Diagramme de classe

# Review de code

## Classe CgrapheOriente

## Classe Cfichier

L’une des principales attentes de ce projets est la lecture d’un graphe depuis un fichier suivant un format précis. Le formalisme que respecte le fichier est composé de balise tel que “ Numero=1” ou encore “Debut=1, Fin=4”. Par conséquent la principale méthodes de cette classe est l’analyse syntaxique (parser) d’un fichier pour la creation d’un graphe donné par l’utilisateur.

Pour realiser ces methodes nous avons utilise les bibliotheque fstream et iostream qui permettent de parcourir des fichier avec des methodes tel que getline ou encore find.

Nous avons decide dajouter une balise suplementaire dans le fichier du graphe cette derniere permet de determiner si le graphe est un graphe Orienté ou Non Oriente. Evidement nous aurions pu egalement permettre a l’utilisateur de parametrer ceci directement depuis le terminal a l’aide des commandes cin et d’une verfication syntaxique de l’entrée. Cependant comme les seules entrée de l’utilisateur se faisait a travers le fichier il nous est parut logique de continuer dans cette lancée.

Cette nouvelle balise se presentent comme suit :

TypedeGraphe=<1 si Non oriente 2 si Orienté>

Si cette balise n’est pas correctement ou n’est pas implementé dans le fichier alors le graphe sera par defaut un graphe orienté

## Classe CSortie

L’une des principales attente de ce projet est l’affichage d’un graphe. Pour cela nous avons décidé de créer les methodes d’affichage dans la classe CSortie en dehors de la classe CgrapheNonOriente. Ceci pour respecter le principe du genie logiciel de contrat de service.

La classe Csortie possède une méthode inhérente au projet, en effet la méthode INVAffichage permet un affichage normalise d’un graphe de la forme :

Une image contenant texte, capture d’écran

Description générée automatiquement

Cette affichage permet de connaitre les sommet d’un graphe notifier par “Information sur le sommet <nom du sommet>”, et de connaitre les arc et sommet lié a notre sommet.

L’affichage “-->” permet de suivre les arc sortant de notre sommet et de connaitre le sommet cible. Et inversement, l’affichage “<--“ permet de connaitre les arc entrant ainsi que les sommet lie a ces arcs.