

Recherche opérationnelle :

Modélisation

I Travail à réaliser :

L'objectif de ce projet est la réalisation d'une application C++ permettant de tester et d'illustrer un ensemble de propriété dans un graphe.

Le travail principal est l'implémentation des algorithmes de graphes vu en cours.

Les propriétés implémentés sont :

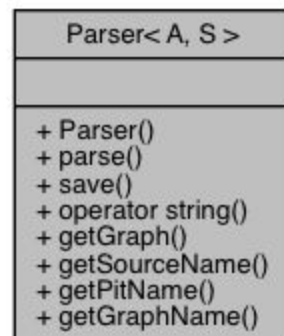
- Le parcours DFS
- La numérotations des sommets
- La détection des circuits
- Le tri topologique
- L'algorithme du plus court chemin
- Le diamètre du graphe.
- Création du graphe retour

Les graphes sont définis dans un fichier avec l'extension .gpr, regroupant les informations nécessaires.

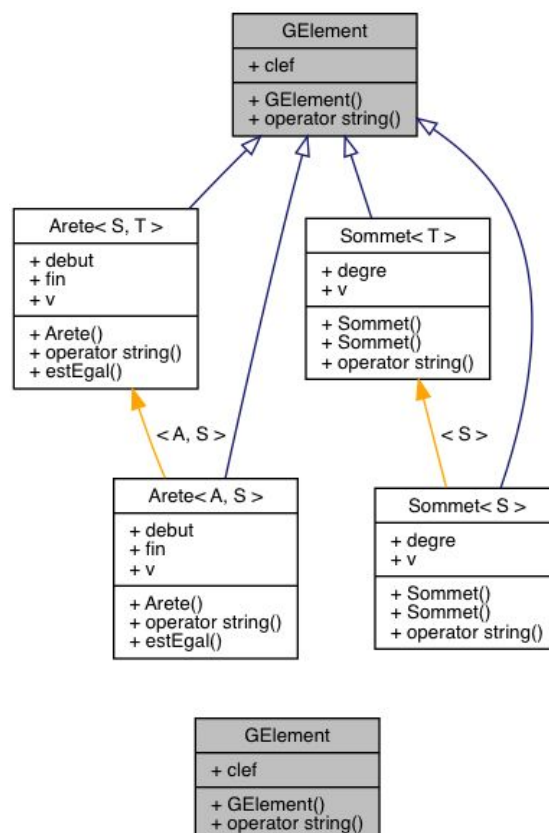
#Instance graphe1 à 4 sommets et 6 arcs	commentaire
ressource 1	nombre de ressources avec fenêtre
sectionSommets	
s1 0 0	sommet ; bornes inférieure et supérieure de la fenêtre
i1 0 5	
i2 0 7	
i3 0 8	
p1 0 10	
source	
s1	nom-sommet-source
puits	
p1	nom-sommet-puits
sectionArcs	
a01 s1 i1 4 3	nom-arc ; sommet-initial ; sommet-terminal ; coût ; temps
a02 s1 i2 8 2	
a03 i1 i2 4 3	
a04 i1 i3 2 6	
a05 i2 i3 3 4	
a06 i2 p1 2 6	
a07 i3 p1 3 4	
sectionGraphe	
graphe1 s1 p1	nom-graphe ; sommet-source ; sommet-puits

II Programmation :

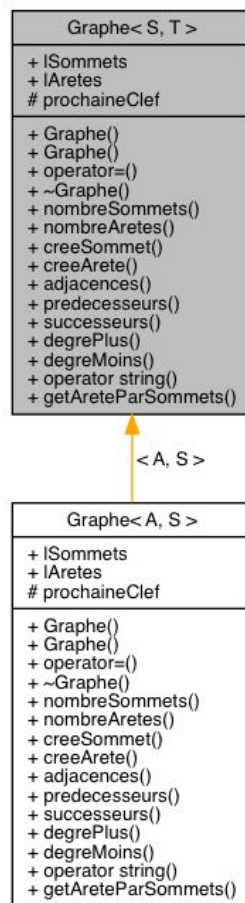
En premier lieu une classe "Parser" a été réalisé afin d'extraire les informations nécessaires à la réalisation du projet du fichier .gpr (informations sur les arêtes, sur les sommets, les coûts, les puits,etc...). De plus cette classe permet de charger et de sauvegarder un fichier texte au format gpr.



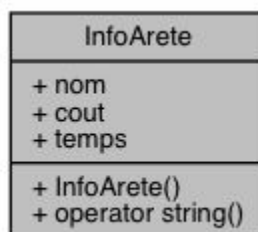
Une classe `GElement` représente la classe de base des éléments d'un graphe qui peuvent être des sommets ou des arêtes. Il permet de définir les clefs des éléments du graphe.



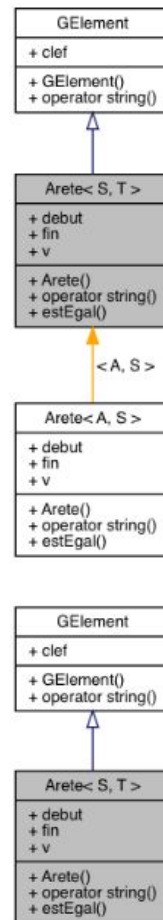
La classe Graphe permet la construction d'un graphe générique.



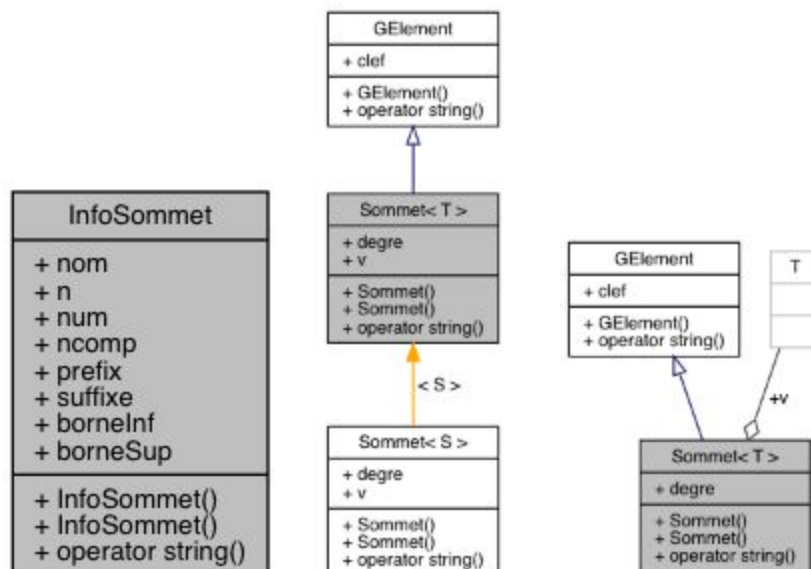
Une classe InfoArete permet de stocker le coût d'une arête spécifique.



Une classe Arete représente une arête de manière générale. Elle est définie par un sommet de début, un sommet de fin et une information V.



Le procédé est le même pour les classes Sommet et InfoSommet permettant respectivement de stocker l'information générale d'un sommet, et de stocker les informations d'un sommet spécifique.



L'utilisation des classes Template permet la généricité du code et par conséquent une adaptation et une évolution plus simple notamment pour l'implémentation des algorithmes de parcour du graphe.

Utilisation du programme :

Option	GNU long option	Meaning
-h, -?	--help	affiche l'aide
-d	--dfs	réalise un DFS
-p	--pcc	réalise une recherche du plus court chemin
-n	--num	numérote les sommets
-c	--circuit	réalise une detection de circuit
-t	--topo	réalise un tri topologique
-x	--connex	réalise une detection de connexité
-i <fileName>	--input=<fileName>	charge le fichier .gpr
-o <fileName>	--output=<fileName>	sauvegarde dans le fichier .gpr"

III Structure de données :

La structure de données est la suivante :

- Un graphe générique
 - Composé d'arêtes et de sommets généraux
 - Les sommets sont pris individuellement pour gérer leurs particularités
 - Les arêtes sont prises individuellement pour gérer leurs particularités comme le coût et les sommets qu'elle relie.

IV Tests et résultats :

Pour les tests ce sont les fichiers .gpr données avec le sujet qui seront utilisés.

Les résultats sont affichés un tableau avec chaque sommet, et ses arcs entrants et sortants

A l'exception du PCC où l'on peut afficher le graphe dans l'ordre (s > i1 > i4 > p).