

Recherche opérationnelle Modélisation

Rapport de projet

présenté en mars 2018 par

Robin Dumont
Bruno Muckli
Yiming Wei

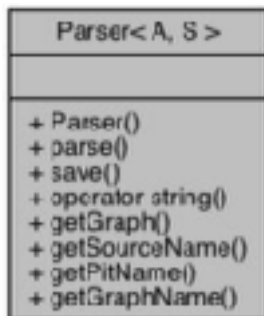
Sujet :

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 - L'algorithme du plus court chemin - Création du graphe retour Les graphes sont définis dans un fichier avec l'extension *.gpr*, regroupant les informations nécessaires.

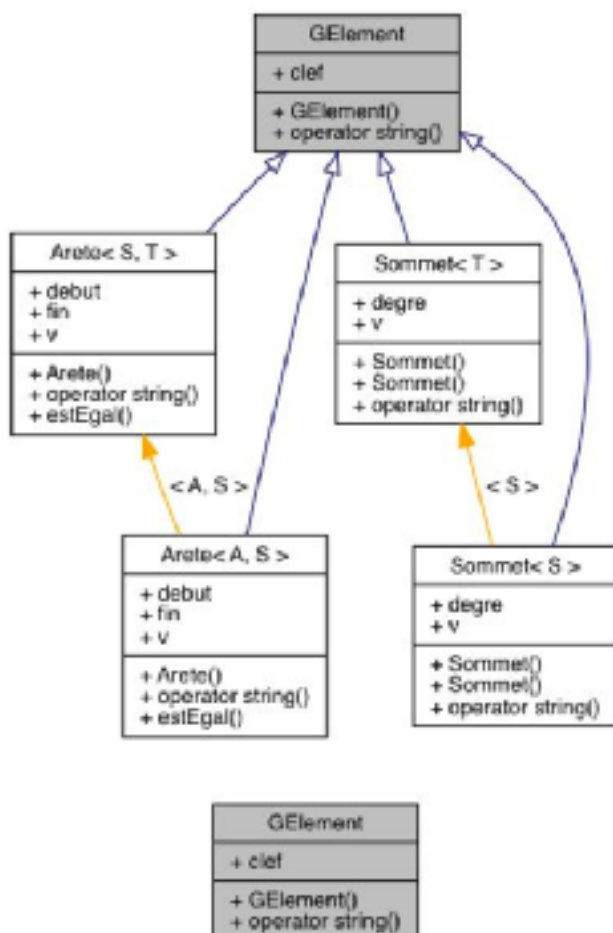
#Instance graphe1 à 4 sommets et 6 arcs	commentaire
ressource 1	nombre de ressources avec fenêtre
sectionSommets	
s1 0 0	sommet ; bornes inférieure et supérieure de la fenêtre
i1 0 5	
i2 0 7	
i3 0 8	
p1 0 10	
source	
s1	nom-sommet-source
puits	
p1	nom-sommet-puits
sectionArcs	
s01 s1 i1 4 3	nom-arc ; sommet-initial ; sommet-terminal ; coût ; temps
s02 s1 i2 8 2	
s03 i1 i2 4 3	
s04 i1 i3 2 6	
s05 i2 i3 3 4	
s06 i2 p1 2 6	
s07 i3 p1 3 4	
sectionGraphe	
graphe1 s1 p1	nom-graphe ; sommet-source ; sommet-puits

Réalisation :

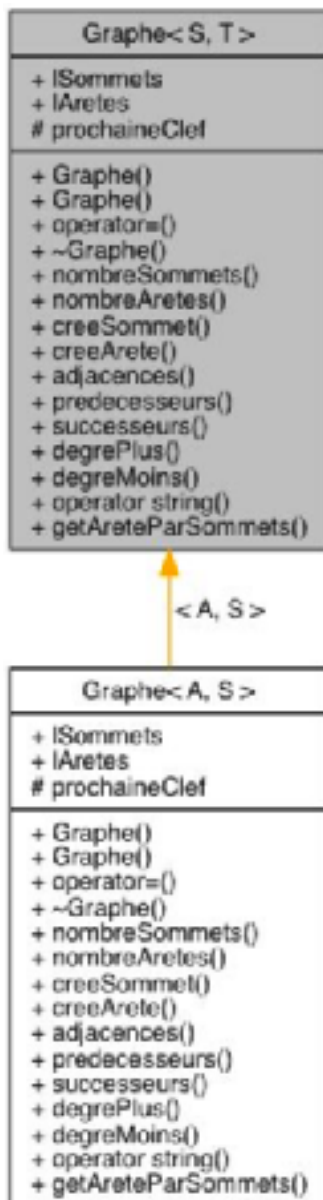
En premier lieu une classe `Parser` a été réalisé afin d'extraire les informations nécessaires à la réalisation du projet du fichier `.gpr` (informations sur les arêtes, sur les sommets, les coûts, les puits, etc...). Cette classe permet de charger et de sauvegarder un fichier texte au format gpr, en proposant des méthodes `parse` et `save`



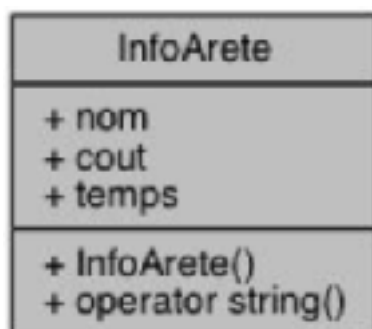
Une classe `GElement` représente la classe de base des éléments d'un graphe qui peuvent être des sommets ou des arêtes. Il permet de définir les clefs des éléments du graphe.



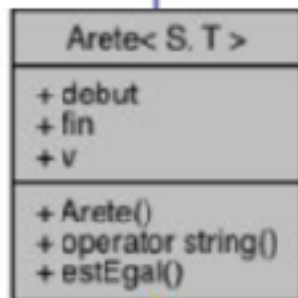
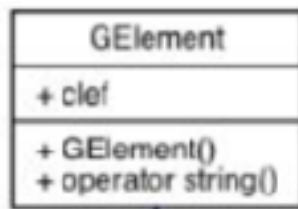
La classe Graphe permet la construction d'un graphe générique vide. On y ajoute ensuite les sommets, puis les arêtes reliant ces sommets.



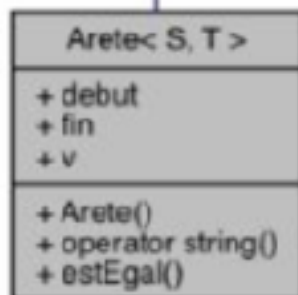
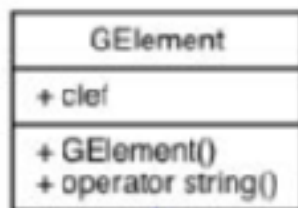
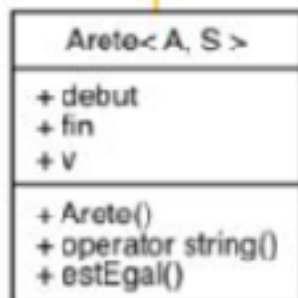
Une classe InfoArete permet de stocker le coût d'une arête spécifique.



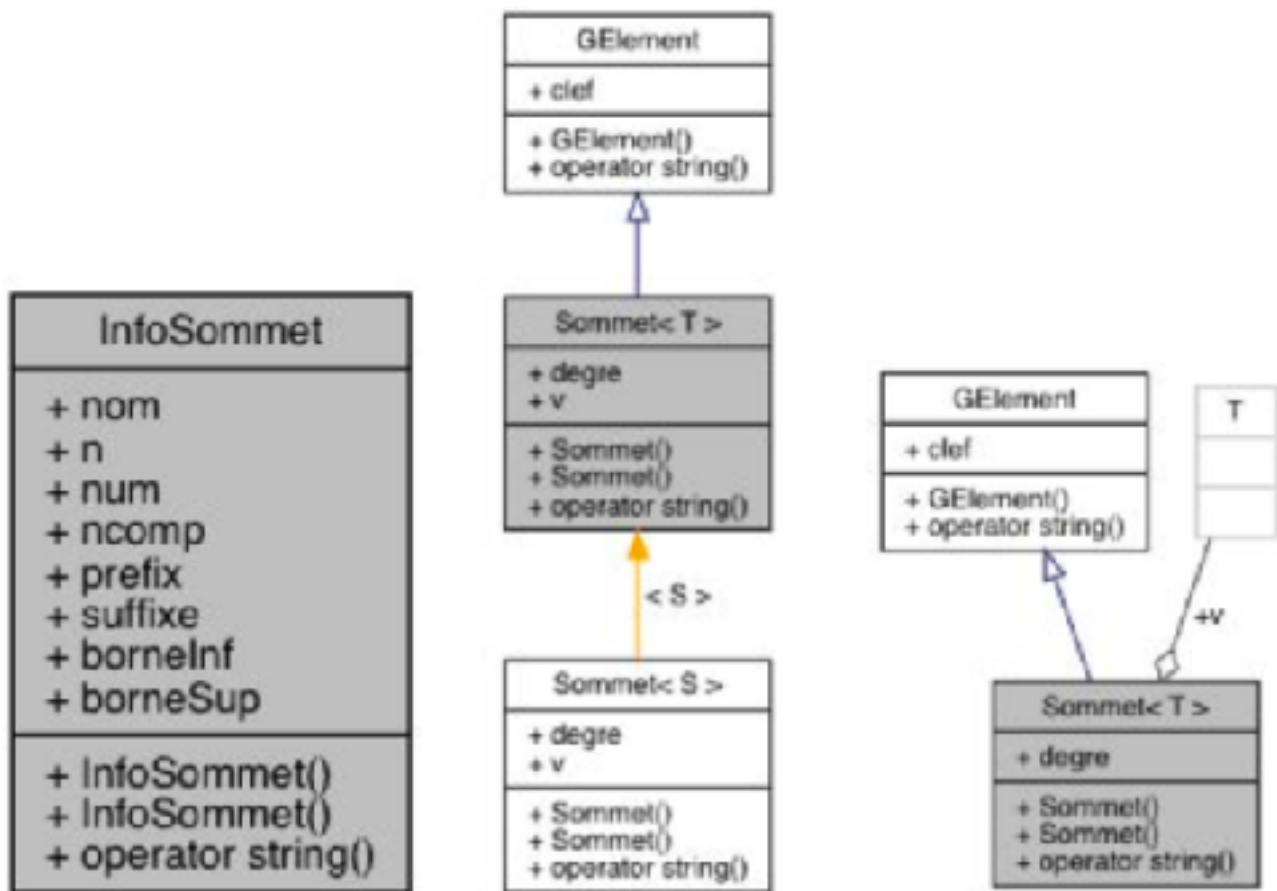
Une classe `Arete` représente une arête de manière générale. Elle est définie par un sommet de début, un sommet de `fin` et une information `v`.



<A, S >



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 parcours du graphe.

Utilisation

```

* ./rom --help

Usage: program [OPTIONS]...

Option          GNU long option      Meaning
-h, -?          --help                Show this message
-d              --dfs                réalise un DFS
-p              --pcc                réalise une recherche du plus court chemin
-i <fileName>   --input=<fileName>         charge le fichier .gpr
-o <fileName>   --output=<fileName>  sauvegarde dans le fichier .gpr

```

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.

Tests et résultats :

Pour les tests ce sont les fichiers *.gpr* données avec le sujet qui seront utilisés. Les résultats sont retournés dans un nouveau fichier *.gpr*. À l'exception du PCC où l'on peut aussi afficher le graphe textuellement.

```

--- Algorithme -----
Execution de l'algorithme de recherche du plus court chemin, version 2.0.0 ;-)

--- Chemin -----
> i1 > i4 > i5

```