

Le lecteur de fichier dot a été implémenté sous la forme d'un package nommé *dotReader* dont la classe principale est le fichier *dotReader.java*.

Ce package est inclus dans le projet Java *TP Algo*, qui contient 4 implémentations des graphes qui héritent l'une de l'autre dans l'ordre suivant: Graphe de base - Graphe Pondéré - Graphe Coloré - Graphe Orienté.

Cette classe contient sept attributs dont trois sont les types de graphes où nous pouvons enregistrer les données du fichier dot: graphePondéré (*grapheP*), grapheColoré (*grapheC*) et grapheOrienté (*grapheO*).

De plus, pour chaque graphe il existe une variable booléenne (*flag*) permettant de savoir quel type de graphe a été choisi pour le fichier dot destiné à la lecture.

Pour dernier, l'attribut (*path*) est où le chemin du fichier est enregistré.

Pour les méthodes, ce lecteur dispose d'une fonction void *readDot()* qui se charge de transformer un fichier dot en une instance de graphe pertinent.

Cette méthode s'appuie sur les méthodes secondaires suivantes:

- *boolean isWeightedNode(String line)*: identifie un arête pondéré.
- *boolean isColoredNode(String line)*: identifie la couleur d'un **sommet**.
- *boolean isNotWeighted(String line)*: permet d'identifier un arête sans poids.
- *Vector<Integer> findNodes(String line)*: renvoie un vecteur contenant les

informations pour un arête non pondéré.

- *public Vector<Integer> findNodesWeighted(String line)*: pareil que la précédente, avec la seule différence que cette fois l'arête est pondéré.

- *public Map.Entry<Integer,String> findColor(String line)*: revoie une paire avec l'identifiant du sommet et sa couleur.

- *public void getNodes(Vector<Vector<Integer>> nodes, String ligne)*: détermine la nature de l'arête et appelle la fonction d'extraction pertinente (*findNodes* ou *findWeightedNodes*).

Enfin, la méthode principale *readDot()* procède de la manière suivante:

- lit le chemin du fichier dot et l'ouvre dans un objet *File*;
- détermine si l'entête du fichier a la bonne syntaxe et détermine aussi si il s'agit d'un graphe orienté ou pas;
- pour les deux cas, la méthode parcourt toutes les lignes du fichier, et extrait les sommets, les éventuels poids et couleurs et enregistre le tout dans un vecteur *nodes* et une hashmap *colors* contenant les sommets et leurs couleurs correspondantes.
- lorsque la recherche se termine, le programme détermine si le graphe est coloré, ou/et s'il est pondéré, pour procéder en suite à remplir le graphe pertinent avec les données dans *nodes* et *colors*.

Le package contient une classe main avec une simple interface sur ligne de commande qui donne la possibilité de choisir son propre fichier en fournissant le chemin, ou de tester le programme avec des fichiers type situés dans le dossier *testing*.