

Initiation à la bibliothèque graphe de l'ISTIC

But

Cette première séance doit vous permettre de manipuler une bibliothèque de graphes, avec le "prétexte" d'implémenter l'algorithme de Roy-Warshall. Cette bibliothèque est la réécriture en Java d'une bibliothèque plus ancienne, écrite en Eiffel (la réécriture étant relativement récente, on peut parler d'une version "beta").

Les interfaces et classes se trouvent dans : `/share/esir1/gra/biblio`, et leur documentation dans `/share/esir1/gra/biblio/javadoc`.

Travail demandé

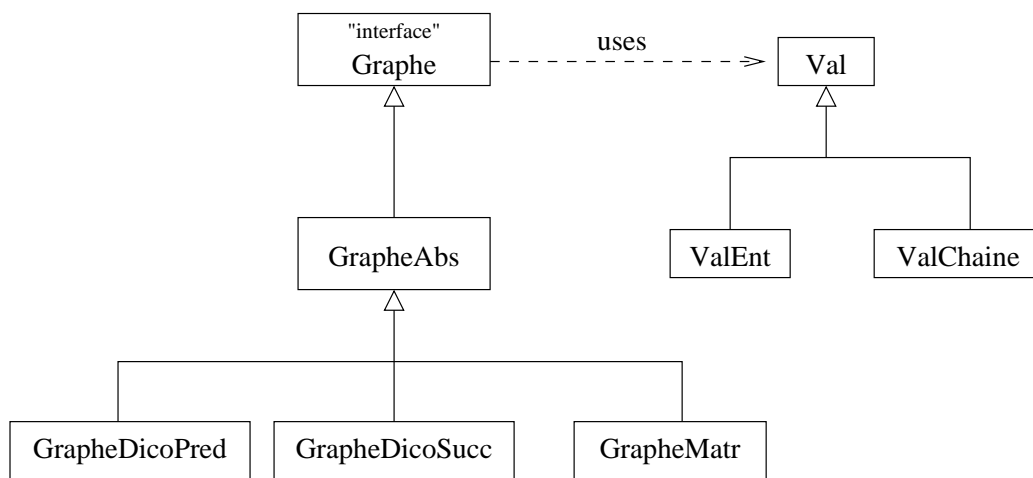


FIGURE 1 – Le diagramme de classes UML de la bibliothèque de graphes.

1. Lire la documentation de la bibliothèque, en particulier celle de l'interface `Graphe`.
2. Lire et comprendre la classe d'application `TP1` fournie dans `/share/esir1/gra/tp_roy_warshall`.
3. Copier `TP1.java` vers un répertoire perso. Compiler puis exécuter ; le programme vous demande de saisir le graphe au format "dictionnaire de successeurs", spécifié dans le commentaire de la méthode `lireSuccs()` de l'interface `Graphe`.
4. Modifier le programme de manière à lire puis écrire le graphe sur des fichiers texte au lieu de l'entrée et de la sortie standard, en créant des objets `FileInputStream`, `FileOutputStream` et `PrintStream`.
5. Ecrire la méthode `fermeture(Graphe g)` qui ajoute au graphe `g` passé en paramètre les arcs correspondant à sa fermeture transitive, suivant l'algorithme de Roy-Warshall. Appliquer cette méthode au graphe lu en entrée avant son affichage par la méthode `ecrireArcs()`.

6. Modifier la méthode `fermeture()` de manière à créer la table de routage correspondant à la fermeture transitive. Celle-ci sera matérialisée en affectant à chaque arc (x,y) le numéro du successeur de x dans le graphe initial, suivant l'un des chemins reliant x à y dans celui-ci. En termes Java, ce numéro sera encapsulé dans un objet de type `ValEnt` associé à l'arc (x,y) .