

TP2 : Généricité et graphes

Ces travaux pratiques ont pour objectif de comprendre les mécanismes de généricité en Java.

1 Paires génériques

Créez une classe `Pair` paramétrée par deux types. Une instance de cette classe regroupera deux objets : une instance de chaque paramètre. Cette classe fournira les moyens de créer des paires ainsi que d'accéder à leurs composantes.

Le but de cet exercice est de se familiariser avec la syntaxe Java pour les génériques. Attention, programmeur C++, si les *templates* et les génériques ont des points communs, ils sont fondamentalement différents.

Une classe générique paramétrée par `T` peut-elle créer une nouvelle instance de `T` ? Expliquer. Qu'en est-il d'un tableau de `T` ?

2 Graphes

Définir l'interface d'un graphe orienté, paramétrée par le type de ses sommets. Elle doit permettre de construire des graphes (ajouter des sommets, des arêtes) et de les explorer (connaître les sommets, les successeurs d'un sommet *etc*).

Créez une classe satisfaisant cette interface et testez-la un peu.

Les classes du *package* `java.util` vous seront sûrement d'une grande utilité.

3 Algorithmes

Si l'interface des graphes est suffisamment riche, on peut programmer divers algorithmes sur les graphes sans se soucier de leur représentation concrète. Cette dernière ne servira que pour les tests.

Dans une classe séparée qui n'aura pas d'instances (*i.e.* un module), programmez les fonctions (statiques) suivantes.

- Une fonction `asDot`, qui change un graphe en une chaîne de caractères, au format Graphviz¹ par exemple.
- Une fonction de parcours, qui reçoit en argument un objet qui déterminera l'ordre de parcours voulu (les méthodes attendues de cet objet seront définies dans une interface).
- Des fonctions qui spécialisent la précédente pour obtenir un parcours en largeur d'une part et en profondeur de l'autre.
- *Ad lib.* Par exemple, la recherche de chemin entre deux sommets. Puis en ajoutant des poids aux arêtes, la recherche de chemin de poids minimal.

1. <http://www.graphviz.org/>