

TD 13 Java – Programmation Générique

Cours

1 – Comparer les méthodes `addAll` de la classe `java.util.Collections` et de l'interface `Collection`.

2 – Application : déclarer une liste de figures et ajouter en une fois plusieurs points (on réutilisera les classes `Figure` et `Point` des premiers TDs).

Exercice : algorithme de Kruskal

Le but de cette séance est d'écrire en Java l'algorithme de Kruskal vu en TD d'algorithme de manière générique sur n'importe quelle type de valeur marquant une arête.

1 – Ecrire la classe générique `AreteValuee` qui contient deux sommets (numéro entier) et une valeur d'un type générique qui implémente `java.lang.Comparable`

2 – Ecrire la classe générique `GrapheValue` qui contient des sommets de 1 à n et un ensemble d'arêtes valuées. Le constructeur de la classe initialise le nombre de sommets à une valeur passée en paramètre et l'ensemble des arêtes à l'ensemble vide. Ajouter une méthode pour ajouter une arête valuée en vérifiant les conditions vues en algorithmique.

3 – Ajouter la méthode `arbreMinimal()` qui renvoie un graphe valué qui est un arbre couvrant minimal du graphe à partir de l'algorithme de Kruskal

4 – Tester vos classes avec une application créant un graphe de quelques sommets avec des arêtes valuées par des objets comparables (`Integer`, `String`, ...)

NB : regarder la documentation des classes `Comparable` et `Collections`.