IN201 : 9 – Généricité

Author : Christophe Garion <garion@isae.fr>
Public : SUPAERO 2A
Date :

Résumé

Ce TP a pour but de vous faire créer et manipuler un type générique.

1 Objectifs

Les objectifs de ce TP sont les suivants :
- comprendre l'utilisation de paramètres de type formels ;
- définir et utiliser un type paramétré ;
- comprendre les problèmes dûs au *type erasure*.

2 Problématique

On cherche ici à modéliser et à implanter un ensemble d'objets qui sont comparables entre eux (comme par exemple des entiers). Cet ensemble d'objets sera un type générique utilisant un paramètre de type formel. De plus, on souhaite pouvoir créer des ensembles d'ensembles comparables.

3 Conception de la classe `Ensemble`

Les ensembles que nous allons considérer seront des ensembles d'objets comparables entre eux. Les ensembles eux-mêmes seront également comparables entre eux. L'interface `Comparable` fournit un type représentant des objets comparables. Tous les objets de type `Comparable` possèdent une méthode `compareTo` qui renvoie un entier. On consultera la documentation javadoc de l'interface `Comparable` pour plus de détails.
On souhaite pouvoir effectuer les opérations suivantes sur un ensemble :
- ajouter un élément à l'ensemble ;
- enlever un élement de l'ensemble ;
- obtenir un itérateur sur l'ensemble ;
- récupérer le minimum de l'ensemble ;
- construire l'union de deux ensembles de types compatibles.

Proposer un diagramme UML de la classe `Ensemble`.

4 Implantation de la classe `Ensemble`

Pour réaliser le TP, vous allez devoir créer un projet Java sous Eclipse en utilisant votre dépôt Subversion. Si vous avez configuré votre dépôt pour qu'il soit disponible dans la vue *SVN Repository Exploring* d'Eclipse, vous créérez votre projet en faisant un *checkout* **à partir du répertoire TP9** de votre dépôt en cliquant droit dessus. N'oubliez pas de configurer votre *build path* pour pouvoir utiliser les éventuelles archives JAR placées dans le répertoire `lib`.

On développera la classe `Ensemble` en prenant soin de tester au fur et à mesure les méthodes que l'on développe.
On se demandera en particulier quel est le type de l'attribut de la classe `Ensemble` permettant de stocker les éléments (on pourra choisir une instance d'une collection).
On pourra également lever des exceptions si cela est nécessaire (minimum d'un ensemble vide etc).

5 Implantation d'une classe `TestEnsemble`

Créer une classe `TestEnsemble` permettant d'exécuter un scénario qui :
- créé un ensemble d'objets de type `Double` place 5 objets de type `Double` dedans ;

- créé un ensemble d'objets de type `Double` place 2 objets de type `Double` dedans ;
- affiche les minima de chacun des ensembles ;
- construit l'union des deux ensembles et affiche son minimum.

On construira de plus une méthode statique `somme` prenant en paramètre un ensemble d'objets de type `Number` et renvoyant la somme des éléments (on utilisera la méthode `doubleValue` de `Number`). On vérifiera que la valeur renvoyée en utilisant le premier ensemble est correcte.

License CC BY-NC-SA 3.0