

## C.10

Utiliser des éléments  
de conception détaillée

## Objectif

Déterminer un conteneur adapté pour une collection d'objets.

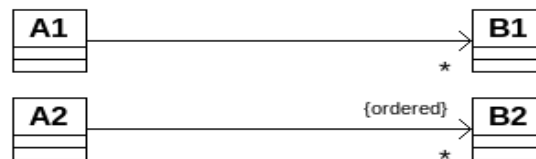
## Préambule

Les associations navigables se traduisent par du code qui dépend notamment de la multiplicité de l'extrémité concernée, mais aussi de l'existence d'une contrainte {ordered} ou d'un qualificatif.

Une multiplicité « \* » vas se traduire par un attribut de type collection de références d'objets. La difficulté consiste à choisir la bonne collection parmi de très nombreuses classes de base. Le choix d'un langage paraît déjà décisif.

Bien qu'il soit possible de créer des tableaux d'objets, ce ne sera pas forcément la bonne solution. On préfère recourir à des collections parmi lesquelles les plus utilisées sont :

- Java : ArrayList et HashMap
- C# : ArrayList, SortedList et HashTable
- C++ (STL) : vector, list, map et set



## Méthode

On propose une méthode basée sur une série de questions qui permettra de choisir un conteneur adapté de la STL.

## Exemple : les jeux de plateau

On va concevoir la relation Plateau-Case. Les cases feront toujours partie du Plateau et celui-ci gèrera leur création et leur destruction : les cases sont donc dans une relation de **composition** avec le Plateau.

Puis, il faut traduire la navigabilité \* entre Plateau et Case. Cela revient à choisir un conteneur pour la collection de cases. La difficulté consiste à choisir le bon conteneur parmi les très nombreuses classes de base fournis avec les langages et leur environnement. En suivant le document en annexe, on obtient :

Tests	Oui/Non
L'ordre est important	<b>oui</b> (on a une système de coordonnées x,y)
Trié par clé	<b>oui</b> (une coordonnée)
Doublons autorisés	<b>non</b> (unicité de la case sur le plateau)
Clés et valeurs séparées	<b>oui</b> (une coordonnée donne une case)
Conteneur choisi	▷ <b>Map</b>

