



Les collections

Gestion d'ensembles d'objets

- Tableaux

```
Base tab1[] = new Base[5];
for(int i=0; i<tab1.length; i++){
    if(i%2 == 0)
        tab1[i] = new Base(i);
    else
        tab1[i] = new Fille(i);
}

for(int i=0; i<tab1.length; i++){
    System.out.println(tab1[i].infos());
}
```

- Taille fixe
- Pas de manipulation aisée des objets dans le tableaux

Tableaux d'objets

- On peut mettre n'importe quoi dedans
- Mais il faut caster pour accéder aux méthodes
 - Exception possible à l'exécution

```
Object tab2[] = new Object[5];
for(int i=0; i<tab2.length; i++){
    if(i%2 == 0)
        tab2[i] = new Base(i);
    else
        tab2[i] = new Fille(i);
}
// tab1[0] = new Object();
for(int i=0; i<tab1.length; i++){
    System.out.println(((Base)tab2[i]).infos());
}
```

Structures de données

- Quelles structures de données connaissez-vous ?
 - Pour la manipulation de collection de données
 - Par exemple pour avoir une taille dynamique
 - Ou pouvoir trier, lister ...

On va faire une liste chaînée

- Pour voir comment ça fonctionne
- Une liste chaînée contient
 - Une donnée membre data
 - Une donnée membre suivant
- Une liste chaînée peut
 - Ajouter un élément en début
 - Ajouter un élément en fin
 - Insérer un élément après un élément présent
 - Supprimer un élément
 - Être parcouru itérativement

Les classes

- Classe Liste
 - Méthodes
 - InsérerTete
 - Insérer(Nœud ElementPrécédent, Objet data)
 - Supprimer(Nœud ElementPrécédent)
 - SupprimerTete
 - Element Suivant
 - AuDebut
 - Donnée membres
 - Nœud tête
 - Nœud courant

Les classes suites

- Classe Nœud
 - Donnée membres
 - Object data
 - Nœud suivant
 - Méthodes
 - A vous de voir si getters/setters

Plutôt que de réinventer

- Java propose un ensemble de classes
 - Permettant la gestion d'ensemble
 - Suivant les structures de données classique

Implementations						
		Hash Table	Resizable Array	Balanced Tree	Linked List	Hash Table + Linked List
Interfaces	Set	HashSet		TreeSet		LinkedHashSet
	List		ArrayList		LinkedList	
	Map	HashMap		TreeMap		LinkedHashMap