

ANNEXE 17 - Les collections – Stack , Vector , List, etc.

Nous avons vu comment utiliser les piles (classe `Stack`)

- * On peut créer une pile « normale » pouvant être composée de n'importe quel type d'objet : `Stack pile = new Stack();`

- * Pour ajouter un élément : `pile.push (new Cravate("999"));`

- * Pour récupérer un objet → on doit transtyper l'objet dans son type d'origine :

`Cravate c = (Cravate) pile.pop();` // retourne un Object par défaut

- * Depuis JDK 1.5 , on peut créer des `Stack` (et d'autres types de Collections) à l'aide de génériques, permettant ainsi de restreindre le contenu de la collection à un certain type seulement. Le type est passé en paramètre entre chevrons :

`Stack<Cravate> pile2 = new Stack();`
`pile2.push (new Cravate ("555"));`
`Cravate c = pile.pop();` → plus nécessaire de transtyper à la sortie de la pile

Les Vectors

- * Une `Stack` est une mise en œuvre simple d'une classe plus générale : `Vector`
- * Un `Vector` vise à représenter un tableau d'objets, chaque item du tableau pouvant être accessible à partir de son index dans le tableau.
- * Cependant, la capacité d'un `Vector` peut s'agrandir au fur et à mesure qu'on ajoute des items dans le `Vector` → allocation dynamique de mémoire

EX. :

Constructeurs :

`Vector premier = new Vector()`




`Vector deuxieme = new Vector(2)`




`Vector<Double> troisieme = new Vector(3, 2)`





Méthodes principales :

void add (Object o) : 


void insertElementAt (Object o, int index) : 

Object get (int index) : 

Object remove (int index): 

boolean remove (Object o): 


boolean contains (Object o): 

void trimToSize(): 

EXEMPLES :


Les trois vecteurs :

```
premier.add ( new Integer ( 5 ) );
```

```
premier.size() ?? ( nombre d'éléments du Vecteur ) = 
```

```
premier.capacity() ?? ( capacité totale du Vecteur ) = 
```


```
deuxieme.add ( new Character ( 'u' ) ) ;
```


NOUVEAU JDK 1.5 → AUTOBOXING : 


```
deuxieme.add ( 't' );
```


```
Character lettre = new Character ( 'y' );
```


```
deuxieme.insertElementAt ( lettre, 1 );
```


```
deuxieme.indexOf ( 't' ); ?? = 
```

```
deuxieme.size(); ?? = 
```

```
deuxieme.capacity(); ?? = 
```

```
troisieme.insertElementAt ( 22.2 , 7 ); ?? = 
```

```
deuxieme.remove ( lettre ); ?? = 
```

```
deuxieme.indexOf ( new Character ( 'u' ) ); ?? = 
```

```
premier.remove ( 4 ); ?? = 
```