

PILE (STACK)

Une pile est une suite d'objets de même type, possédant un ordre bien précis, et dont le nombre est variable

(Une pile peut être vide!)

L'ajout et le retrait se font « au sommet »
de la pile

ajouter	↔	empiler	↔	push
retirer	↔	dépiler	↔	pop

L.I.F.O (Last In First Out)

Interface Pile :

`int taille()`

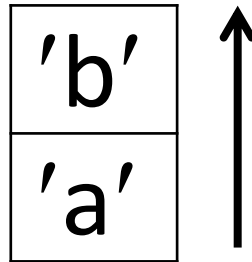
`boolean estVide()`

`void push(E element)`

`E pop() throws PileVideException`

`E sommet() throws PileVideException`

Exemple schématique d'utilisation d'une pile :



Implémentation de la pile via une table : redimensionnable (x2)

'c'
'b'
'a'

0	1	2	3
'a'	'b'	'c'	

taille = 3

Vecteur (VECTOR)

Un vecteur est une suite d'objets de même type, possédant un ordre bien précis, et dont le nombre est variable

(Un vecteur peut être vide!)

Vecteur (VECTOR)

L'ordre des éléments est important.
On accède à un élément via son **rang**.

On appelle **rang** d'un élément d'une structure linéaire, le nombre d'éléments situés avant lui. On abstrait ainsi la notion d'indice d'un tableau.

Interface Vecteur :

```
int taille()
```

```
boolean estVide()
```

```
String toString()
```

```
E element(int rang)  
           throws VecteurOutException
```

```
void insere(int rang, E element)  
           throws VecteurOutException
```


Interface Vecteur :

```
void ajoute(E element)
```

```
E replace(int rang, E element)  
           throws VecteurOutException
```

```
E supprime(int rang)  
           throws VecteurOutException
```

Implémentation du vecteur via une table : redimensionnable (x2)

0	1	2	3
'a'	'b'	'c'	

taille = 3