



Classe ArrayList

Java possède plusieurs classes faisant partie des **collections** permettant de stocker des objets et de proposer des méthodes pour trier, interdire les doublons,....

La classe **ArrayList** fait partie des collections, elle permet de créer des objets pour stocker une quantité variable et évolutive d'objets.



Classe ArrayList

Création

- ArrayList non spécialisée, contient des références d'objets de toute nature

```
ArrayList maListe = new ArrayList(); //
```

- ArrayList spécialisée , ne peut contenir que des références d'objets instances de la même classe.

```
ArrayList <String> maListe2 = new ArrayList () ;
```



Classe ArrayList

Utilisation

```
String chaine1 = "Salut";  
ArrayList<String> liste = new ArrayList () ;
```

```
liste.add("Hello"); // ajout d'une référence d'objet  
liste.add(chaine1);  
liste.add(1 , "Hola"); /*Ajout d'une référence d'objet à la position 1  
Correspond à une insertion*/
```

```
int nb = liste.size(); //retourne le nombre de références d'objets (3)
```

```
liste.clear(); //suppression de tous les éléments
```

```
liste.remove(1) ; // suppression de la référence d'objet à la position 1
```



Classe ArrayList

Accès aux éléments

```
ArrayList <String> liste = new ArrayList () ;
```

```
liste.add("Hello");  
liste.add("salut");  
liste.add("Hola");
```

```
String chaine1 = liste.get(1); /* Retourne la référence d'objets à la  
                               position1* (chaine1 = "salut") */
```

```
liste.set(0,"Bonjour"); //remplace la référence de la position 0
```



Classe ArrayList

Parcourir un ArrayList

Dans une boucle for :

```
ArrayList<String>liste = new ArrayList () ;  
.....  
// Remplissage  
.....  
//Affichage  
for(int i = 0 ; i < liste.size() ; i++){  
    System.out.println(liste.get(i));  
}
```



Classe ArrayList

Parcourir un ArrayList

Dans une boucle for each :

```
ArrayList<String> liste= new ArrayList () ;  
.....  
// Remplissage  
.....  
//Affichage  
for(String s : liste){  
    System.out.println(s);  
}
```



Classe ArrayList

Parcourir un ArrayList

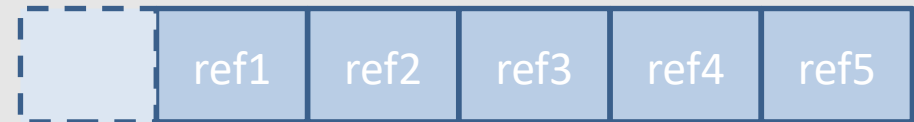
Utilisation d'un Iterator:

Un Iterator est une Interface (classe particulière) qui permet d'instancier des objets spécialisés dans le parcours de collection

```
Iterator it = liste.iterator();
```

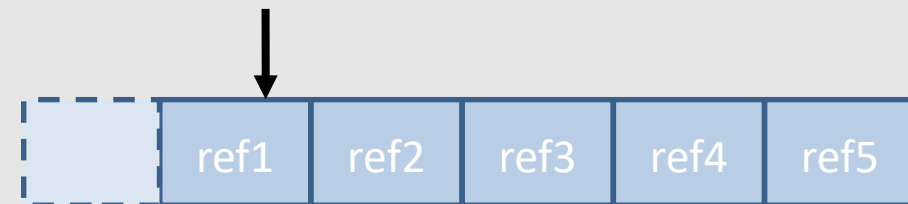
/*Création

Le curseur pointe avant la première référence/*



```
boolean b = it.hasNext(); //retourne true s'il reste une référence à parcourir
```

```
monObjet = it.next(); /*retourne la référence  
suivante (ref1) et déplace le curseur*/
```





Classe ArrayList

Parcourir un ArrayList

Avec Iterator:

```
ArrayList <String> liste = new ArrayList () ;
```

```
.....// Remplissage
```

```
//Affichage
```

```
Iterator it = liste.iterator();
```

```
while( it.hasNext() ){
```

```
    System.out.println(it.next());
```

```
}
```




Classe ArrayList

Exercice

- Créer un objet de type ArrayList non spécialisé
- Insérer des éléments de types divers:
char c = 'z' , int valeur = 25, String s = "chaîne de caractères"
double d = 32.25 , StringBuilder sb ;
- Afficher le contenu de l'ArrayList.
- Afficher le contenu de chacun des éléments de l'ArrayList en précisant sa classe d'origine avec cette méthode :

`liste.get(i).getClass();`

- Tester les différentes méthodes de l'ArrayList