



Programmation Orientée Objet

Les tableaux et les chaînes de caractères

Mme Hassna BENSAG
Email: h.bensag@gmail.com

La déclaration des tableaux

- Les tableaux permettent de stocker un ensemble fini d'éléments d'un type particulier. L'accès à un élément du tableau se fait en utilisant son indice. Le premier élément d'un tableau possède l'indice 0.
- Pour déclarer un tableau en java , on utilise l'une des syntaxes suivantes :

```
int tab[] = new int[10];
```

- Ou

```
int [] tab = new int[10];
```

- Ou

```
int [] tab;  
tab = new int[10];
```

La déclaration des tableaux

- Pour créer un tableau à plusieurs dimensions, il faut déclarer un tableau de tableau en utilisant une paire de crochet pour chaque dimension.

```
int matrice[][] = new int[4][3];
```

- La taille des tableaux de la seconde dimension peut ne pas être identique pour chaque occurrence, comme le montre l'exemple suivant :

```
int tab2[][] = new int[3][];  
tab2[0] = new int[3];  
tab2[1] = new int[2];  
tab2[2] = new int[1];
```

Manipulation des tableaux

- L'initialisation explicite d'un tableau

```
int tab3[] = {4, 5, 8, 19};  
  
int tab4[][] = {{4, 5}, {2, 9}, {0, 6}};
```

- Pour parcourir et manipuler un tableau on peut utiliser soit la boucle **for** ou la boucle **foreach**.

- À l'aide de la boucle **for**:

```
int tab3[] = {4, 5, 8, 19};  
for (int i = 0; i < tab3.length; i++) {  
    System.out.println(tab3[i]);  
}
```

- À l'aide de la boucle **foreach**:

```
int tab3[] = {4, 5, 8, 19};  
for(int v: tab3){  
    System.out.println(v);  
}
```

Les chaînes de caractères

- Les chaînes de caractères en Java sont des objets qui sont soutenus en interne par un tableau de caractères.

- Pour définir une chaîne de caractères, on utilise l'une des syntaxes suivantes :

```
String str= "Java est orienté objet";
```

- Ou

```
String str= new String( original: "Java est orienté objet");
```

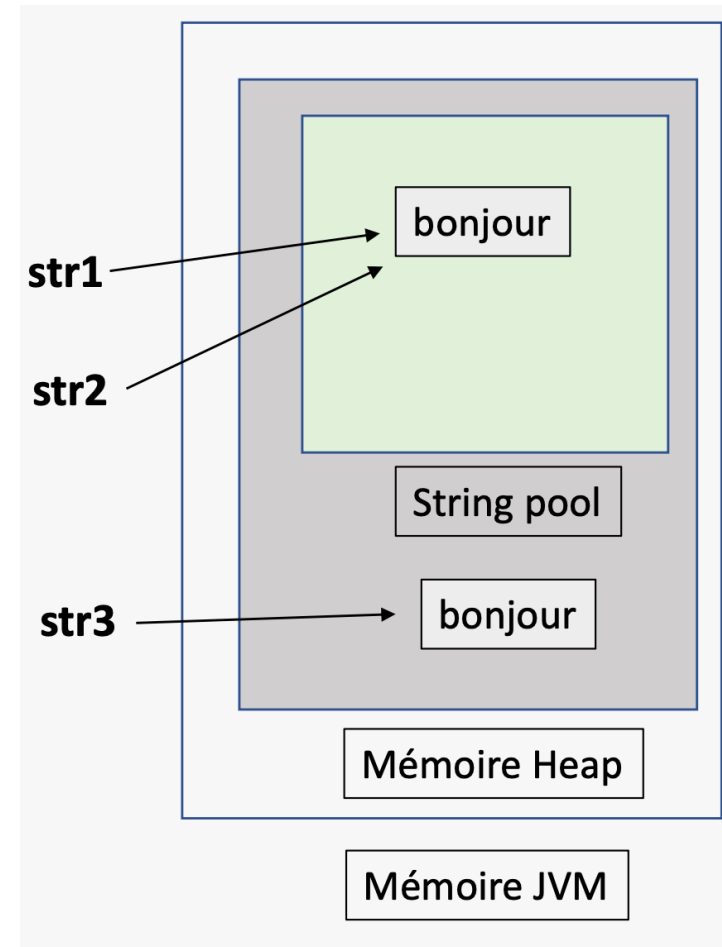
- Les objets String ne sont pas modifiables. Chaque fois qu'une modification est apportée à une chaîne, une chaîne entièrement nouvelle est créée.

Les chaînes de caractères

- Chaque fois qu'un littéral de chaîne est créé, la machine virtuelle Java (JVM) vérifie le pool de constantes String.
- Si la chaîne existe, une référence au pool de constantes String est renvoyée.
- S'il s'agit d'un nouveau String, cet objet est créé dans le pool de constantes String.

```
String str1= "bonjour";  
String str2= "bonjour";  
String str3= new String( original: "bonjour");
```

```
System.out.println(str1==str2);  
System.out.println(str1==str3);  
System.out.println(str1.equals(str3));
```



Les chaînes de caractères

- Les méthodes de la classe String:

Méthode	Description
length()	Renvoie le nombre de caractères dans la chaîne.
charAt(int i)	Renvoie le caractère au ième indice.
substring (int i, int j)	Renvoie la sous-chaîne de i à j-1 indice.
concat(String str)	Concatène la chaîne spécifiée jusqu'à la fin de cette chaîne.
indexOf (String s)	Renvoie l'indice dans la chaîne de la première occurrence de la chaîne spécifiée.
equals(String str)	Compare cette chaîne à la chaîne passée en paramètre
compareTo(String str)	Compare deux chaînes lexicographiquement.
toLowerCase(), toUpperCase()	Renvoie respectivement une chaîne en minuscule et en majuscule
trim()	Renvoie la copie de la chaîne, en supprimant les espaces blancs aux deux extrémités. Cela n'affecte pas les espaces au milieu. -
replace (char oldChar, char newChar):	Renvoie une nouvelle chaîne en remplaçant toutes les occurrences de oldChar par newChar.

Les chaînes de caractères

```
String str="Java est orienté objet";
System.out.println(str.length());
System.out.println(str.charAt(5));
System.out.println(str.substring(5,10));
System.out.println(str.concat( str: "!!!"));
System.out.println(str.indexOf("est"));
String str1="chaîne1";
System.out.println(str1.equals("chaîne2"));
System.out.println(str1.compareTo("chaîne1"));
System.out.println(str.toUpperCase());
System.out.println(str.replace( target: "Java", replacement: "Python"));
```

Résultats

```
22
e
est o
Java est orienté objet!!!
5
false
0
JAVA EST ORIENTÉ OBJET
Python est orienté objet
```


Diviser une chaîne de caractères

- On peut diviser une chaîne de caractères en utilisant la méthode `split` en fonction d'un délimiteur. L'exemple suivant permet de diviser une chaîne en considérant l'espace comme délimiteur:

```
String tabstr[]=str.split(" ");  
for(String s:tabstr){  
    System.out.println(s);  
}
```

- Ce programme affiche le résultat suivant:

```
Java  
est  
orienté  
objet
```

Manipulation des chaînes avec StringBuilder et StringBuffer

- Les classes **StringBuilder** et **StringBuffer** fournissent une alternative à la classe **String** pour créer des chaînes de caractères modifiables.
- Les deux classes possèdent les mêmes méthodes. La différence est que les méthodes de la classe **StringBuffer** sont synchronisées et donc elle est « **Thread safe** ».
- **StringBuilder** est plus rapide et préférable par rapport à **StringBuffer** pour les programmes à un seul thread.

• Exemple:

```
public class StringBuilder {  
    public static void main(String[] args) {  
        //créé un objet StringBuilder  
        StringBuilder str=new StringBuilder();  
        //ajout la chaîne "Bonjour"  
        str.append("Bonjour ");  
        str.append("tout le monde");  
        //affiche la chaîne  
        System.out.println(str.toString());  
        //supprime la chaîne entre 0 et 6  
        str.delete(0,7);  
        System.out.println(str.toString());  
        //insere la chaîne "Bonsoir" dans l'indice 0  
        str.insert( offset: 0, str: "Bonsoir");  
        System.out.println(str.toString());  
        //inverse la chaîne  
        str.reverse();  
        System.out.println(str.toString());  
    }  
}
```

Résultats

Bonjour tout le monde
tout le monde
Bonsoir tout le monde
ednom el tuot riosnoB