

Chapitre 3: Les tableaux

HANENE CHETTAOUI HAMROUNI

MAÎTRE ASSISTANTE EN INFORMATIQUE

Les tableaux statiques

Un tableau en Java est un objet permettant de rassembler des données de même type sous un même identificateur.

Les éléments du tableau peuvent être des données de type primitif ou des références d'une classe donnée.

Les tableaux sont toujours manipulés par référence.

La taille d'un tableau est déterminée lors de sa création et ne peut être modifiée par la suite.

Les indices des tableaux commencent par 0.

Les tableaux statiques

Déclaration d'un tableau

TypeElement nomTableau [] ; *ou* *TypeElement [] nomTableau ;*

→ *int [] T1 ;*

→ *Personne [] T2 ;*

Allocation d'un tableau : *nomTableau = new TypeElement [TailleTableau] ;*

→ Les éléments sont alors initialisés avec leur valeur par défaut (zéro pour les éléments de type numérique, false pour les éléments de type boolean et null pour les éléments de type objet).

Les tableaux statiques

```
T1 = new int[5] ;
```

→ *Réservation de l'espace mémoire d'un tableau comportant 5 éléments de type entier. Chaque élément est initialisé à zéro.*

Remarque :

On peut regrouper la déclaration et l'allocation d'un tableau :

```
TypeElement [] nomTableau=new TypeElement [TailleTableau] ;
```

→ `int[] T1 = new int[5] ;`

Les tableaux statiques

Déclaration, allocation et initialisation

Pour les tableaux de petites tailles **(et dont les éléments sont connus à l'avance)**, on peut déclarer, allouer et initialiser les éléments du tableau :

TypeElement [] nomTableau = {val1, val2, , valn} ;

→ `int [] T2 = {5, 7, -12, 8} ;`

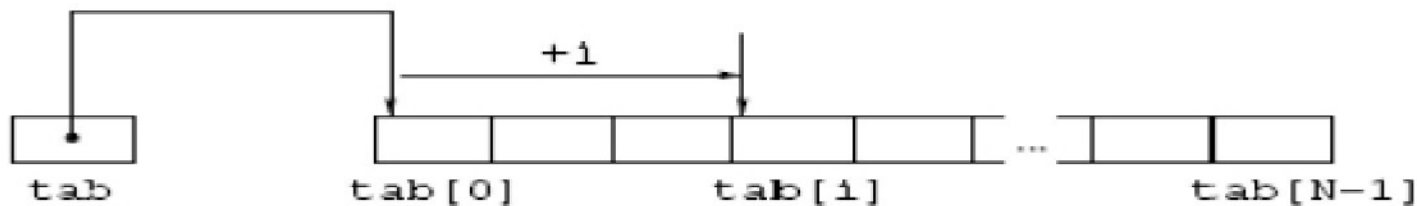
→ `String [] T3 = {"Abc", "X ;", "GTy"} ;`

Les tableaux statiques

Accès aux éléments :

Une fois le tableau créé, on accède à ses éléments en utilisant la même notation qu'en C :

tab[i] où *i* est une expression de type entier.



Taille d'un tableau :

Pour récupérer la taille d'un tableau, on utilise le champ `length` :

nomTableau.length

Les tableaux statiques

Les tableaux d'objets : Un tableau déclaré à partir d'une classe, contient des références à des objets de la classe en question. Chaque élément du tableau représente une référence à un objet.

```
public class TestPersonne{  
    public static void main(String [] args) {  
        Personne t[] = new Personne [2];  
        // déclaration d'un tableau de deux éléments  
        t[0] = new Personne (34, "Ben Salah", "ali", "12/12/2000", "Tunis"); // nouvel objet  
        t[0].afficher();  
        t[1] = new Personne (123, "Tounsi", "Mohamed", "12/12/2000", "Tunis");  
    }  
}
```


Les tableaux dynamiques : Vector

Les vecteurs en Java sont des tableaux d'objets dont la taille varie en fonction des éléments que l'on ajoute ou enlève.

Java fournit la classe **Vector** dans le package **java.util**

Un objet de la classe Vector est similaire à un tableau puisqu'il permet de stocker plusieurs valeurs.

Contrairement aux tableaux, un Vector ne requiert pas la manipulation d'indices pour gérer ses éléments. Cette gestion se fait à travers les méthodes de cette classe.

Les tableaux dynamiques : Vector

À importer : **import java.util.Vector;**

Déclaration et (1 manière de) construction

```
Vector<typeObjet> nomArray = new Vector<typeObjet> ();
```

```
Vector<String> noms = new Vector<String> ();
```

```
Vector<Integer> vectInteger = new Vector<Integer> ();
```

Les tableaux dynamiques : Vector

Taille d'un vecteur	int size()
Effacer le contenu du vecteur	void clear()
Ajout d'un élément à la fin du vecteur	boolean add(typeObjet o) void addElement(typeObjet o)
Supprimer un élément	typeObjet remove(int indice) boolean remove(typeObjet o)
Récupération du 1 ^{er} élément	typeObjet firstElement()
Récupération du dernier élément	typeObjet lastElement()

Les tableaux dynamiques : Vector

Récupération d'un élément dans une position	typeObjet elementAt(int indice)
Récupération de l'indice de la première occurrence	int indexOf(typeObjet o) int indexOf(typeObjet o, int position)
Changer un élément par un autre	void setElementAt(typeObjet o, int position)
Insérer un élément à une position	void insertElementAt(typeObjet, int)

Les tableaux dynamiques : Vector

Type primitif	Type référence (wrappers des types primitifs)
int	Integer
float	Float
double	Double
char	Character
boolean	Boolean
short	Short
long	Long
byte	Byte

Lecture des données à partir du clavier

Pour utiliser la classe Scanner, il faut d'abord l'importer :

```
import java.util.Scanner;
```

Ensuite il faut créer un objet de la classe Scanner :

```
Scanner sc = new Scanner(System.in);
```

Pour récupérer les données, il faut faire appel sur l'objet **sc** aux méthodes décrites ci-dessous. Ces méthodes parcourent la donnée suivante lue sur l'entrée et la retourne :

- **String next()** : donnée de la classe String qui forme un mot,
- **String nextLine()** : donnée de la classe String qui forme une ligne,
- **boolean nextBoolean()** : donnée booléenne,
- **int nextInt()** : donnée entière de type int,
- **double nextDouble()** : donnée réelle de type double.

Rq : il n'existe pas de méthode pour lire un caractère

Lecture des données à partir du clavier

```
import java.util.Scanner;
```

```
Scanner sc=new Scanner(System.in);
```

```
String str = sc.nextLine();
```

```
int i=sc.nextInt();
```

```
double j=sc.nextDouble();
```