5. Tableaux et chaines de caractères

Mr IGHIL Mohamed **INSIM Boumerdes**

2025

1. Les Tableaux

1.1 Présentation

- Un **tableau** est une liste **ordonnée** et **numérotée** d'éléments du même type.
- Chaque élément est associé à un numéro appelé **index**. L'indexation commence par **0**, ce qui signifie que le **premier** élément est associé à un index **0**, le **deuxième** à **1**, etc.

1.1 Tableau pour un nombre fixe d'éléments

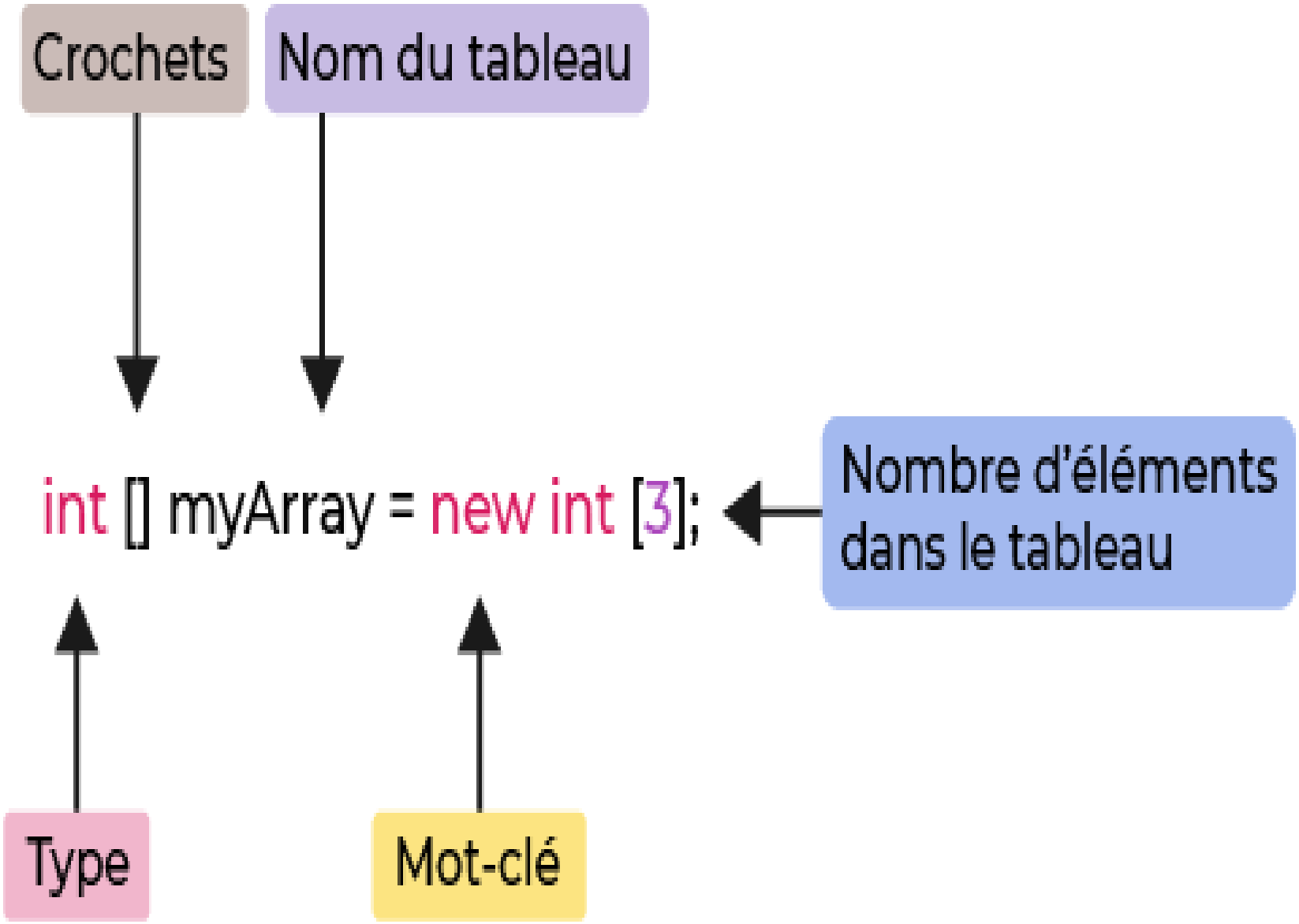
- **Déclarer** un tableau utilise la même syntaxe que pour n'importe quelle variable.
- Par exemple, On peut déclarer un tableau d'entiers avec la syntaxe suivante :

int[] number;

- Le **type** des éléments que le tableau contiendra, suivi de [] .
- Le **nom** de la variable qui doit expliciter clairement l'intention du tableau.

Instanciez ensuite le tableau :

Number = new int[7] ;


Crochets
Nom du tableau
int [] myArray = new int [3];
Nombre d'éléments dans le tableau
Type
Mot-clé

Création tableau avec toutes les variables et les valeurs:

```
Number = new int[]{6,2,3,7,3,4,1};
System.out.println(number[0]);
```

- On peut utiliser des **tableaux multidimensionnels** :
- String[][] tab = new String[30][12];
- tab[9][5]=« IGHIL Mohamed";

1.2 Tableau pour un nombre d'éléments pas fixe

Les tableaux sont bien pratiques, mais ils ont leurs **limites** :

- ils ont une **taille fixe** ;
- On ne modifier que les valeurs **existantes**

Exercices

2. Les chaines de caractères

Les chaînes de caractères littérales de Java (par exemple "abc"), sont représentées comme des instances de la classe ***String***.

La classe ***String*** est une classe spéciale :

- les chaînes de caractères peuvent se **concaténer** à l’aide de l’opérateur **+**, ou à l'aide de la méthode ***concat***.

- les instances peuvent ne pas être créées explicitement
- ***String s = "abc"*** ; au lieu de
- ***String s = new String("abc") ;***

Exemple :

```
String texte = "Bonjour";
```

le compilateur Java génère un **objet** de type **String** avec le contenu spécifié.

Il est donc possible d'écrire :

```
String texte = "Java Java Java".replace('a','o');
```

Exemple :

```
String texte = "Java Java Java";

texte.replace('a','o');

System.out.println(texte) ;

texte = texte.replace('a','o');

System.out.println(texte) ;
```

Résultat :

Java Java Java

Jovo Jovo Jovo

2.1 La classe String

- Une chaîne de caractères en Java, est encapsulée dans une instance de type java.lang.String

```
private String UneChaine;

Void majuscule(String chaine)

    {

      UneChaine = chaine.toUpperCase()

         }
```

3. Les opérations sur les chaînes de caractères

- La classe String possède de nombreuses méthodes pour effectuer des traitements sur la chaîne qu'elle encapsule et/ou sur d'autres chaînes.

3.1. Le test de l'égalité

- La classe String redéfinit la méthode **equals**() héritée de la classe Objet pour retourner **true** si le paramètre n'est pas **null** et si c'est un objet de type String qui encapsule une chaîne ayant la même séquence de caractères en tenant compte de la casse.

```
String texte1 = "texte 1";
String texte2 = "texte 2";
if ( texte1.equals(texte2) )
   {
      // les deux chaînes sont égales //
   ...
    }
```

```
String texte1 = "texte";

String texte2 = "TEXTE";

System.out.println(texte1.equals(texte2));

System.out.println(texte1.equalsIgnoreCase(texte2));
```

Résultat :

false

true

```
String texte1a = "texte 1";

String texte1b = "texte 1";

String texte2a = "texte 2";

String texte2b = new String("texte 2");

System.out.println(texte1a == texte1b);

System.out.println(texte1a == texte2a);

System.out.println(texte2a == texte2b);
```

Resultat:

true

false

false

```
String message = "PRODUITS À 10 €";

System.out.println(message.toLowerCase());
```

produits à 10 €

```
String message = "Produits à 10 €";

System.out.println(message.substring(11));

System.out.println(message.substring(0, 9));
```

Résultat :

10 €

Produits

```
String message = "Produits à 10 €";

System.out.println(message.length());
```

Résultat :

15

String[] noms = "nom1,nom2,nom3".split(",");

System.out.println(noms[1]); Copy

La méthode split() de la classe String permet de couper une chaîne de caractères selon le séparateur fourni en paramètre.

Résultat :

nom2

```
String chaine = "Java";

char car = chaine.charAt(2);

System.out.println(car);
```