

# Les Matrices : projet *ArrayTest*

## Définition

Les matrices sont des sortes de conteneurs à variables. Toutes les données dans une même matrice doivent être du même type, on parle ainsi de matrices d'entiers, de matrices de chaînes, etc.

## Utilité

On peut déclarer 200 variables de type *int* en 200 lignes et une matrice de 200 *int* en 1 ligne...on gagne ainsi en rapidité, lisibilité et légèreté du code. De plus les matrices sont utiles si l'on doit par exemple stocker les prénoms de plusieurs personnes, puis leur nom de famille, puis leur âge... on crée à cet effet 3 matrices. Enfin, il est plus utile de travailler avec les matrices car on connaît le nombre d'éléments qu'elles contiennent (on peut donc par exemple créer un matrice de prénoms puis une autre de noms de famille sans oublier aucune entrée.

## Syntaxe

Déclaration d'une matrice :

- `String mots[ ] ;`
- `String [ ] mots;`

Dans les deux cas, le programme crée une matrice de chaînes ayant pour nom *mots*. On note la syntaxe très proche des déclarations de variables.

Initialisation d'une matrice :

- `String[ ] nomDesJoueurs = new String[10];`
- `nomDesJoueurs[0] = "franck" ;`
- `nomDesJoueurs[1] = "john" ;`

Ces commandes créent un matrice de chaînes de nom *nomDesJoueurs* et contenant 10 « cases ». Elles donnent ensuite la valeur *franck* à la première de ces « cases » et *john* à la seconde.

Note : les valeurs des matrices sont numérotées à partir de 0.

- `String[ ] prenom = {"franck", "john", "gerthrud", "moloss", "grota"};`

Cette commande crée une matrice de nom *prenom* et lui attribue directement des valeurs. La matrice contient donc 5 « cases ».

## Application : Projet *ArrayTest*

Le programme *ArrayTest* crée une matrice contenant les prénoms de programmeurs célèbres. Il affiche cette matrice, puis crée une autre matrice de la même grandeur (avec le même nombre de « cases »). Cette seconde matrice contient les noms de famille correspondant à chaque prénom. Le résultat est affiché.

```
class ArrayTest { // classe principale

    // création de la première matrice de prénoms
    String[] firstNames = { "Dennis", "Grace", "Bjarne", "James" };

    // la seconde matrice contient le même nombre d'arguments que la première
    String[] lastNames = new String[firstNames.length];

    void printNames() { // méthode pour afficher le contenu de la 2nd matrice
```

```

int i = 0; // i du type integer et égal à 0

// afficher la valeur i+0=0 de la matrice 1 et de la matrice 2
// ceci consiste donc à afficher le contenu de la première case
//des 2 matrices
System.out.println(firstNames[i]
    + " " + lastNames[i]);

// (i=0)i=i+1, donc de la seconde case des matrices
i++;
System.out.println(firstNames[i]
    + " " + lastNames[i]);
// (i=1) i=i+1, donc de la seconde case des matrices
i++;
System.out.println(firstNames[i]
    + " " + lastNames[i]);
i++;
System.out.println(firstNames[i]
    + " " + lastNames[i]);
}

public static void main (String arguments[]) {
    ArrayTest a = new ArrayTest(); // appel de la classe
    a.printNames(); // appel de la méthode printNames()
    System.out.println("-----");
    // définition des valeurs des cases de la 2nd matrice
    a.lastNames[0] = "Richie";
    a.lastNames[1] = "Hopper";
    a.lastNames[2] = "Stroustrup";
    a.lastNames[3] = "Gosling";
    a.printNames(); // appel de la méthode printNames()
}
}

```

```

C:\WINDOWS\Bureau\JAVA\SITE\Exemples\ArrayTest>java ArrayTest
Dennis null
Grace null
Bjarne null
James null
-----
Dennis Richie
Grace Hopper
Bjarne Stroustrup
James Gosling

```