

# Rappels, les tableaux natifs en Java

Fabrice.Kordon@lip6.fr



# Tableaux à une dimension

 À la manière de C...



```
public class Tableaux {  
    public static void main(String []args){
```

```
        int t1[] = new int [2];  
        int [] t2;  
        int [] t3 = {1, 3,5 ,7};
```

```
        // initialiser t1
```

```
        t1[0] = 1;
```

```
        t1[1] = 2;
```

```
        // initialiser t2
```

```
        t2 = new int[]{3,4};
```

```
        t2[0] = 0;
```

```
        // afficher le tout
```

```
        System.out.printf("t1 = {%d, %d}\n", t1[0], t1[1]);
```

```
        System.out.printf("t2 = {%d, %d}\n", t2[0], t2[1]);
```

```
        System.out.printf("t3 = {%d, %d, %d, %d}\n»,  
                           t3[0], t3[1], t3[2], t3[3]);
```

```
    }
```

```
}
```

```
$ javac -Xlint Tableaux.java  
$ java Tableaux  
t1 = {1, 2}  
t2 = {0, 4}  
t3 = {1, 3, 5, 7}
```

# Tableaux à une dimension

 À la manière de C...



```
public class Tableaux {  
    public static void main(String []args){
```

```
        int t1[] = new int [2];  
        int [] t2;  
        int [] t3 = {1, 3,5 ,7};
```

```
        // initialiser t1
```

```
        t1[0] = 1;
```

```
        t1[1] = 2;
```

```
        // initialiser t2
```

```
        t2 = new int[]{3,4};
```

```
        t2[0] = 0;
```

```
        // afficher le tout
```

```
        System.out.printf("t1 = {%d, %d}\n", t1[0], t1[1]);
```

```
        System.out.printf("t2 = {%d, %d}\n", t2[0], t2[1]);
```

```
        System.out.printf("t3 = {%d, %d, %d, %d}\n»",
```

```
                t3[0], t3[1], t3[2], t3[3]
```

```
    }
```

```
}
```

```
$ javac -Xlint Tableaux.java  
$ java Tableaux  
t1 = {1, 2}  
t2 = {0, 4}  
t3 = {1, 3, 5, 7}
```

**Affichage?**

Mauvaise solution pour afficher le tableau

# Opérateurs sur les tableaux natifs

3

## Récupérer la taille d'un tableau

- `length`

## Recopier un tableau

- `clone()` ou `arraycopy()`

- ▶ <http://docs.oracle.com/javase/tutorial/java/nutsandbolts/arrays.html>

## Affichage d'un tableau complet

- `toString()`

# Tableaux multi-dimensionnels

## Cela n'existe pas en Java...

 On construit des «tableaux de tableaux»

```
public class MatriceSimple{
    public static void main(String []args){

        int [][] m1 = {
            {1, 0},
            {0, 1}
        };
        int [][] m2 = new int [2] [];

        m2[0] = new int [3];
        m2[1] = new int [3];

        System.out.println("Dimension de m2 = " +
            m2.length + " x " +
            m2[1].length);
    }
}
```

**Dimension N?**

Tableau de tableaux ...  
de tableaux

```
$ java MatriceSimple
Dimension de m2 = 2 x 3
```

# Tableaux multi-dimensionnels

## Cela n'existe pas en Java...

 On construit des «tableaux de tableaux»

```
public class MatriceSimple{
    public static void main(String []args){

        int [][] m1 = {
            {1, 0},
            {0, 1}
        };
        int [][] m2 = new int [2] [];

        m2[0] = new int [3];
        m2[1] = new int [3];

        System.out.println("Dimension de m2 = " +
            m2.length + " x " +
            m2[1].length);
    }
}
```

**Dimension N?**

Tableau de tableaux ...  
de tableaux

**Indépendance dans la  
2<sup>me</sup> dimension**

m2[0] et m2[1] peuvent avoir  
des tailles différentes

```
$ java MatriceSimple
Dimension de m2 = 2 x 3
```

# En guise de conclusion...

5

 **Les tableaux existent en natif dans Java mais...**

**... on leur préfère souvent des classes dédiées**

 Framework Collection

 **Présenté plus loin dans le cours**

 Car basé sur les templates