

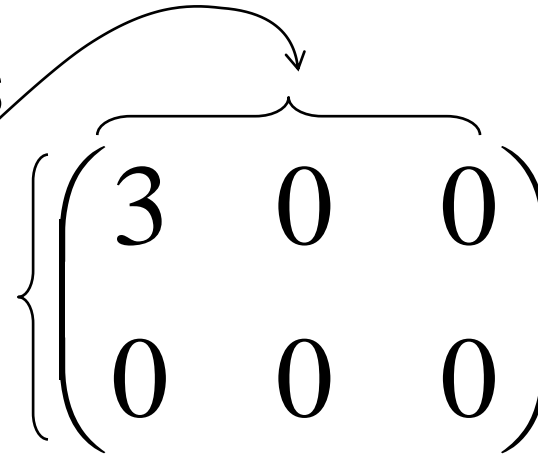
Parcourir, passer en argument et retourner des tableaux

8.3 OPÉRATIONS SUR LES TABLEAUX

Tableaux à plusieurs dimensions

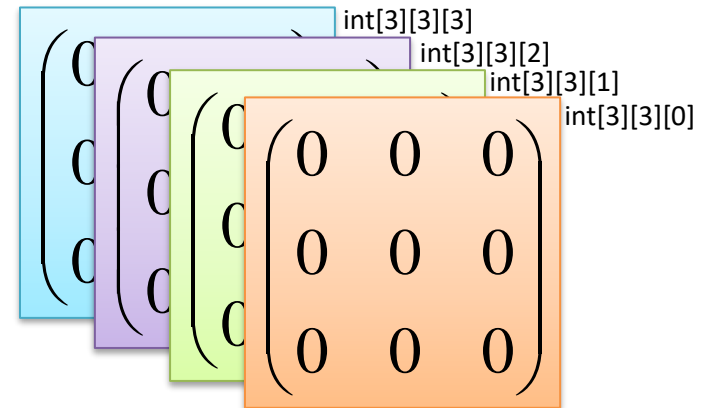
- Tableaux multidimensionnels

```
int[][] array = new int[2][3];  
array[0][0] = 3;
```



- Pas limités à 2 dimensions !

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

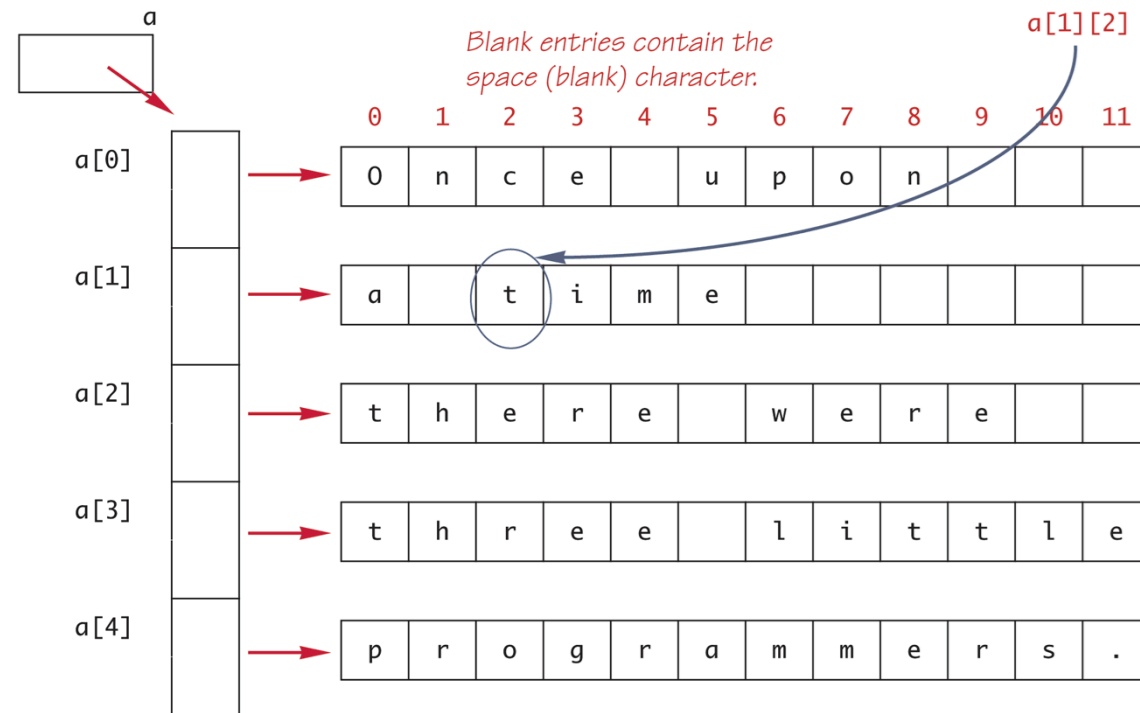


Tableaux à plusieurs dimensions

Display 6.17 Two-Dimensional Array as an Array of Arrays

```
char[][] a = new char[5][12];
```

Code that fills the array is not shown.



(continued)

Tiré de Savitch, *Absolute Java*, 4^{ème} éd.

Tableaux statiques

- En Java, contrôle dynamique des bornes.

```
int[] anArray = new int[10];  
anArray[22] = 34;
```

→ Génère une erreur à **l'exécution**
→ *ArrayOutOfBoundsException*

Exception in thread "main" [java.lang.ArrayIndexOutOfBoundsException:](#)
[Index 22 out of bounds for length 10](#)
at course.arrays.ArrayDemoInClass.main([ArrayDemoInClass.java:12](#))

Parcourir tous les éléments d'un tableau

Taille n

```
for(int i = 0; i < foo.length; i++) {  
    Do something with foo[i]  
}
```

Taille nxm

```
for(int i = 0; i < foo.length; i++) {  
    for(int j = 0; j < foo[i].length; j++) {  
        Do something with foo[i][j]  
    }  
}
```

Taille nxmxq

```
for(int i = 0; i < foo.length; i++) {  
    for(int j = 0; j < foo[i].length; j++) {  
        for(int k = 0; k < foo[i][j].length; k++) {  
            Do something with foo[i][j][k]  
        }  
    }  
}
```

Tableaux en argument de fonction

- Syntaxe standard

```
static double computeAverage(double[] v) {  
    double result = 0;  
  
    for (int i = 0; i < v.length; i++) {  
        result = result + v[i];  
    }  
  
    result = result / v.length;  
    return result;  
}
```

Tableaux comme valeur de retour

Il est possible de retourner un tableau depuis une fonction

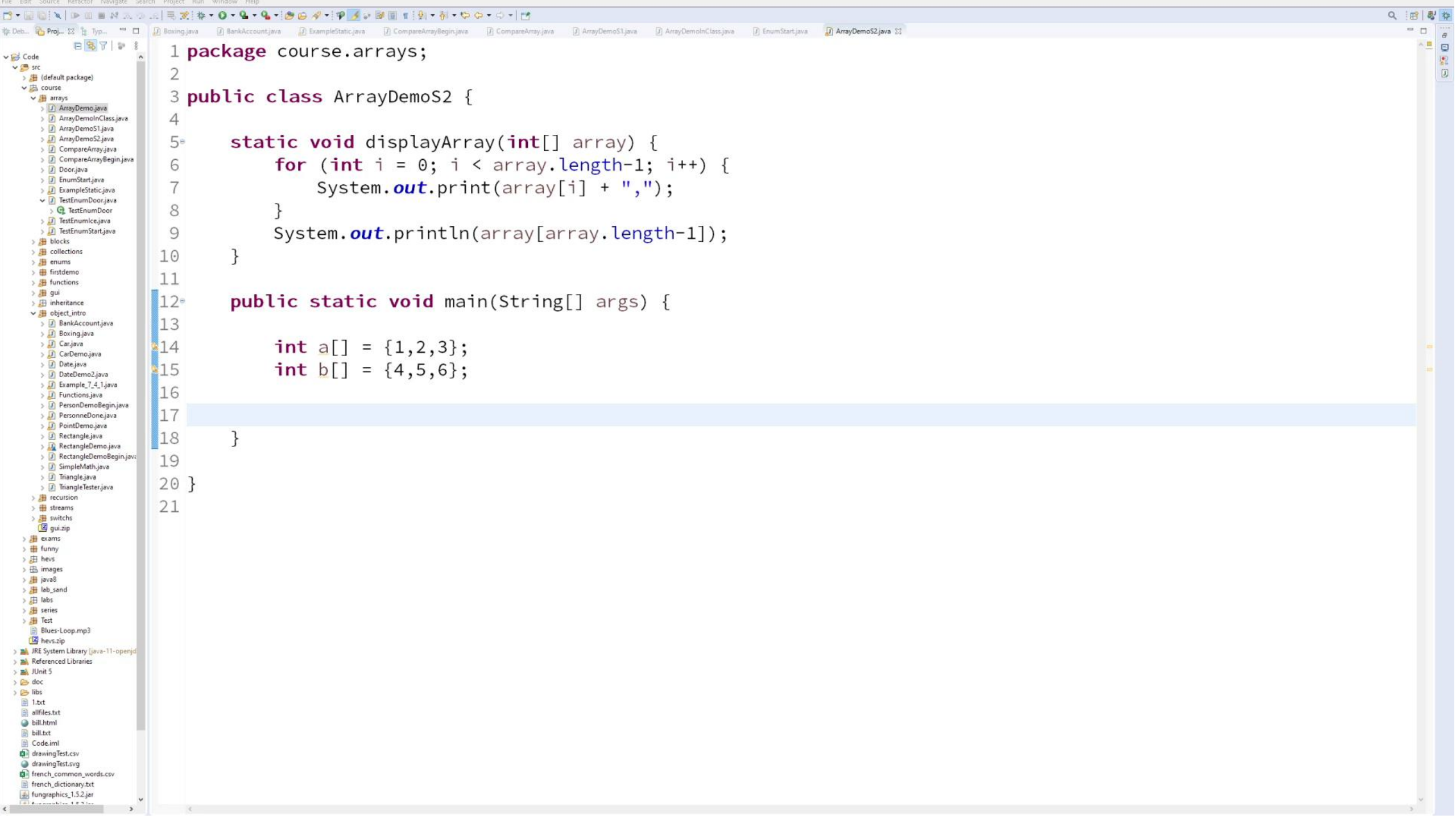
```
TYPE[] Method_Name(Parameter_List)
{
    TYPE[] temp = new TYPE[ArraySize];
    <Some code to fill temp goes here>
    return temp;
}
```

```
public static int[] incrementedArray(int[] a, int increment)
{
    int[] temp = new int[a.length];
    for (int i = 0; i < a.length; i++)
        temp[i] = a[i] + increment;

    return temp;
}
```

Danger : '=' avec tableau

- Un objet tableau est une référence
 - un nom de tiroir mais pas le tiroir lui-même!
- Opérateur = fait que assigner le nom
 - Ne crée pas un nouveau tiroir !
 - Ne copie pas le contenu du tiroir !
 - $a = b$ signifie que a *référence* même tiroir que b



```
1 package course.arrays;
2
3 public class ArrayDemoS2 {
4
5     static void displayArray(int[] array) {
6         for (int i = 0; i < array.length-1; i++) {
7             System.out.print(array[i] + ",");
8         }
9         System.out.println(array[array.length-1]);
10    }
11
12    public static void main(String[] args) {
13
14        int a[] = {1,2,3};
15        int b[] = {4,5,6};
16
17
18    }
19
20 }
21
```

Danger: '=' avec tableau

- Comment réaliser une vraie copie ?

'==' avec tableau

- Pour les mêmes raisons, égalité n'est pas contrôlable avec == !

```
double[] a = {1.2, 2.3, 5.3, 4.4};  
double[] b = {1.2, 2.3, 5.3, 4.4};  
boolean c = a == b;
```

- Vérifier élément par élément avec une boucle !

Conclusion



- Deux structures de données
 - Types énumérés
 - Tableaux

Fin de la matière pour 1^{er} semestre
Labos
Révision classes et tableaux
Traitement d'image
Mini-projet jeu