

Algorithmique & Programmation

Tableaux multi-dimensionnels

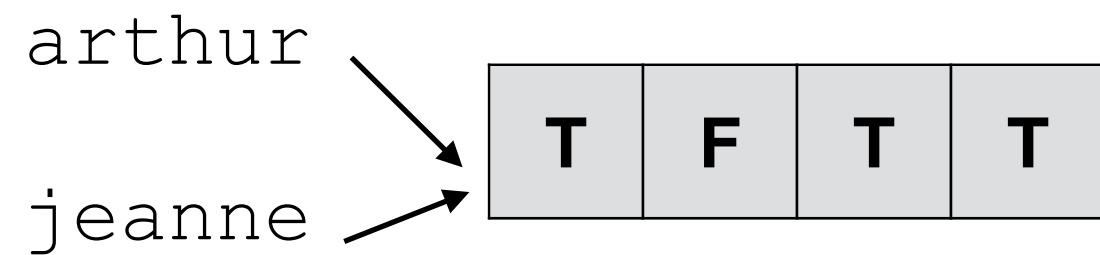
yann.secq@univ-lille.fr

ABIDI Sofiene, ALMEIDA COCO Amadeu, BONEVA Iovka, CASTILLON Antoine,
DELECROIX Fabien, LEPRETRE Éric, Timothé ROUZÉ, SANTANA MAIA Deise,
SECQ Yann

Les tableaux

- Gérer un ensemble de données de même type
- Caractérisé par le type des éléments contenus et leur nombre
- Chaque élément est accessible via un indice
- Une des structures de données la plus simple
- **MAIS ATTENTION A L'AFFECTATION AVEC DES TABLEAUX !**

Variable de type tableau = adresse et pas valeur



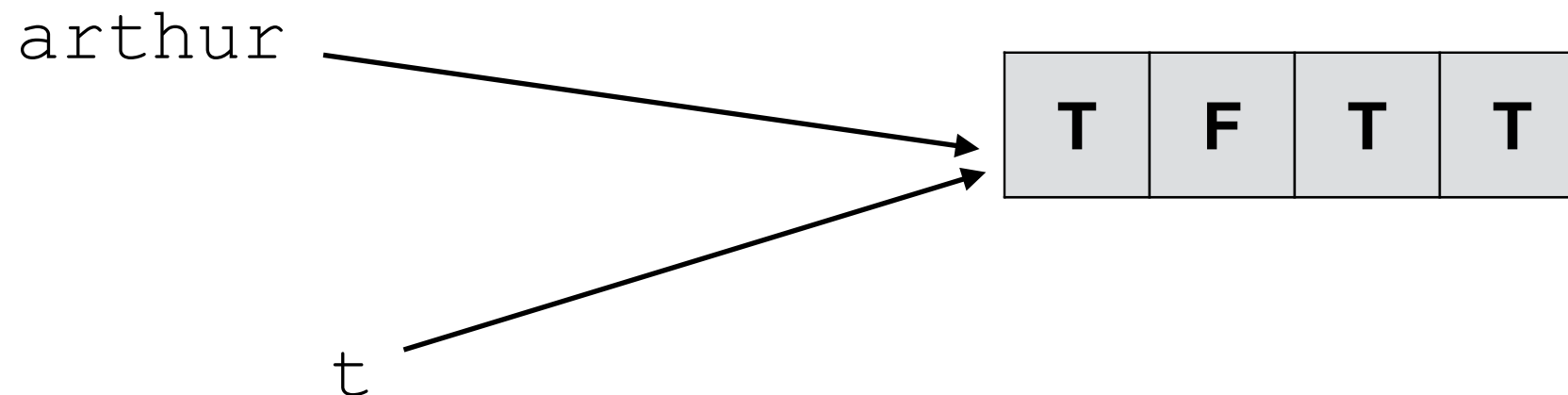
```
1 final boolean T = true, F = false;
2 boolean[] arthur = new boolean[]{T, F, T, T};
3 println(arthur);
4 boolean[] jeanne = arthur;
5 println(jeanne);
6 arthur[0] = F;
7 println(arthur);
8 println(jeanne);
```

```
3 [Z@76ed5528
5 [Z@76ed5528
7 [Z@76ed5528
8 [Z@76ed5528
```

```
3 true false true true
5 true false true true
7 false false true true
8 false false true true
```

Variable de type tableau = adresse et pas valeur

```
void permute(boolean[] t, int idx1; int idx2) {  
    boolean tmp = t[idx1]; t[idx1] = t[idx2];  
    t[idx2] = tmp;  
}  
void algorithm() {  
    boolean[] arthur = new boolean[]{T, F, T, T};  
    permute(b, 0, 1);  
    println(arthur);  
}
```



Opérations sur les tableaux

- **La déclaration:** donner le type des éléments stockés
- **L'allocation ou la création:** donner la taille (ie. Le nombre d'éléments) du tableau
- **L'initialisation:** donner les valeurs initialement contenues dans le tableau
- **L'utilisation:** accès en lecture ou en écriture aux différents éléments du tableau
- **L'accès à la taille:** obtenir le nombre d'éléments que peut contenir un tableau donné

Tableaux multi-dimensionnels

- Comment jouer à la bataille navale ?
- Nécessité de pouvoir utiliser des tableaux multi-dimensionnels : à 2 dimensions, à 3 dimensions ou à n dimensions ...
- Une dimension = un indice, donc n dimensions = n indices !

Tableau à 2 dimensions

	0	1	2	3	4	5
	43	34	2	3	5	1

2ième dimension (colonne) →

	0	1	2	3	4	5
0	43	34	2	-4	5	1
1	4	34	267	3	29	2
2	8	349	2	56	5	1
3	43	34	234	3	87	90

1er dimension (ligne) ↓

Dimensions et indices

Diagram illustrating a 2D array structure with dimensions and indices.

2ième dimension (colonne) (horizontal axis, indices 0 to 5)

1er dimension (ligne) (vertical axis, indices 0 to 3)

0	43	34	2	-4	5	1	← tab[0]
1	4	34	267	3	29	2	← tab[1]
2	8	349	2	56	5	1	← tab[2]
3	43	34	234	3	87	90	← tab[3]

The cell containing the value 234 is highlighted, corresponding to the index `tab[3][2]`.

Déclaration d'un tableau

- **Type et nombre de dimensions**

```
int[] <nomTableau>; // Tableau à 1 dimension d'entiers
```

```
int[][] <nomTableau>; // Tableau à 2 dimensions d'entiers
```

- **Allocation d'un tableau à n dimension:** `<nomTableau> = new int[<taille1>]...[<tailleN>] (n fois)`

- **Exemple de déclarations :**

```
int[][] notes = new int[5][7]; //Tableau à 2 dimensions d'entiers
```

```
boolean[] resultats = new boolean[10]; //Tableau à 1 dimensions de booléens
```

```
char[][][] chaine = new char[2][5][3]; // Tableau à 3 dimensions de caractères
```

Utilisation d'un tableau

- Pour accéder à une case, il suffit de connaître ses indices (ligne et colonne) :

```
<nomTableau>[<indiceLigne>][<indiceColonne>]
```

```
// retourne le contenu de la case d'indice (<indiceLigne>, <indiceColonne>)
```

- Exemple d'accès au contenu d'une case :

`notes[4][3]` : case se situant à l'intersection de la ligne d'indice 4 et de la colonne d'indice 3

`chaine[0][0][0]` : case d'indice (0, 0, 0)

Modifier une case

- Modifier une case d'un tableau à 2 dimensions:

`<nomTableau>[<idxLigne>][<idxColonne>] = <val>`

*range la valeur `<val>` dans la case d'indice (`<idxLigne>`,
`<idxColonne>`)*

- Exemple de modification du contenu d'une case :

`notes[4][3] = 5` : range la valeur 5 dans la case d'indice (4, 3)

`chaine[0][0][0] = '^'` : range '^' dans la case d'indice (0, 0, 0)

`notes[4][3] = notes[5][5]` : copie le contenu de la case d'indice (5, 5) dans la case d'indice (4, 3)

Taille d'un tableau à plusieurs dimensions ?

- Obtenir la taille d'un tableau:

```
int length(<unTableau>, int <dim>)
```

retourne la taille de la dimension <dim> de <unTableau>

- Exemple d'accès à la taille d'un tableau :

`length(notes, 1)`: retourne le nombre de lignes (ie. la première dimension) du tableau `notes` ici 5

`length(notes, 2)`: retourne le nombre de colonnes (ie. la deuxième dimension) du tableau `notes` ici 7

Exemple: les matrices

- Une matrice est un tableau de réels à deux dimensions
- Conception d'algorithmes permettant de:
 - créer une matrice
 - d'afficher une matrice
 - rechercher une valeur dans la matrice
 - d'additionner et de multiplier deux matrices

