

DUT MMI – IUT de Marne-la-Vallée  
17/10/2014  
M1202 - Algorithmique

***Cours 3***  
***Méthodologie***  
***Tableaux et boucles***

# Sources

---

- *Le livre de Java premier langage*, d'A. Tasso
- Cours INF120 de J.-G. Luque
- Cours FLIN102 de l'Université Montpellier 2
- Cours de J. Henriet : <http://julienhenriet.olymp-network.com/Algo.html>

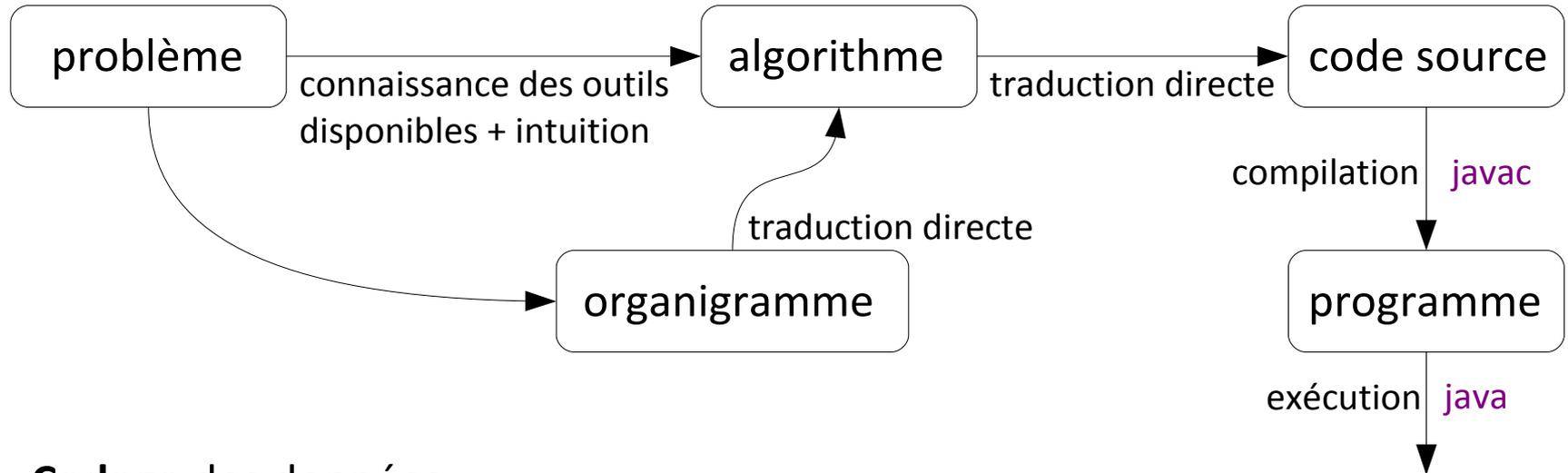
# Plan du cours 3 – Tableaux et boucles

---

- Résumé des épisodes précédents
- Méthodologie : lire et comprendre un algorithme
- Méthodologie : concevoir un algorithme
- Les tableaux
- Lecture du contenu d'un tableau

# Résumé des épisodes précédents

Du problème au programme pour le résoudre :



**Codage** des données :

- Pour chaque **type** de **variable** (entiers, flottants, chaînes de caractères, couleurs, booléens), une méthode de **codage** en binaire est choisie (en Java : `int`, `float`, `double`, `String`, `Color`, `boolean`, ...)
- Définition d'**opérations de base** pour chaque type de données (en Java : `+`, `-`, `*`, `/`, `%`, `&&`, `||`, `!`, ...)

# Méthodo : Lire et comprendre un algorithme

Premiers éléments à identifier :

- qu'est-ce que l'algorithme **prend en entrée** ? **Combien** de variables, de quel **type** ?
- qu'est-ce que l'algorithme **renvoie en sortie** ? **Rien** ? Ou bien **une** variable ? De quel **type** ?

Ensuite :

- quels sont les autres **algorithmes appelés** par l'algorithme ?

Enfin :

- faire la **trace** de l'algorithme, c'est-à-dire l'essayer sur un **exemple** (... ou plusieurs pour passer au moins une fois par toutes les instructions de l'algorithme) et voir ce que valent **toutes les variables à chaque étape** (et noter ces valeurs dans un tableau contenant une ligne par variable et une colonne par étape),
- noter en particulier le **résultat obtenu en sortie** pour une **entrée testée**.

# Méthodo : Concevoir un algorithme

---

Premiers éléments à identifier :

- quels sont les **outils à disposition** ? (pour ces outils : données en entrée, type de données en entrée, résultat en sortie, type de résultat en sortie, résultat attendu sur un exemple...)
- quel est le **comportement attendu** pour mon algorithme ? (données en entrée, type de données en entrée, résultat en sortie, type de résultat en sortie, résultat attendu sur un exemple...)

Ensuite, résoudre le problème en utilisant ces outils :

- comment résoudre le problème **étape par étape** ? (essayer sur l'exemple testé)
- est-ce que les **outils à disposition** sont **utilisables** pour réaliser chaque étape ?

Enfin :

- comment **structurer** l'utilisation des outils à disposition ? (**combinaison** des différents outils à l'intérieur de structure de **boucles**, de **tests**, utilisation d'un **organigramme**...)
- comment **décomposer** le problème ? (et **reformuler** chaque sous-problème pour le résoudre avec les outils à disposition, écrire un algorithme par sous-problème)

+ fiche méthodo : Vérifier un algorithme !

# Plan du cours 3 – Tableaux et boucles

---

- Résumé des épisodes précédents
- Méthodologie : lire et comprendre un algorithme
- Méthodologie : concevoir un algorithme
- **Les tableaux**
- Lecture du contenu d'un tableau

# Les tableaux

Les tableaux sont des variables qui contiennent **plusieurs variables de même type**, stockées chacune dans une des cases du tableau.

Par exemple,

Un **tableau d'entiers** :

4
5
1
23
8
9

Un **tableau de chaînes de caractères** :

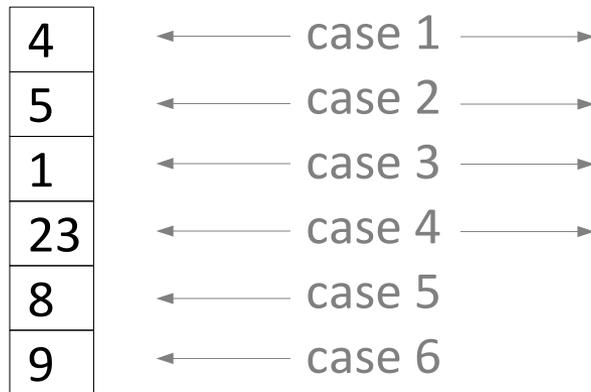
"chaine1"
"chaine2"
"blabla"
"toto"

# Les tableaux

Les tableaux sont des variables qui contiennent **plusieurs variables de même type**, stockées chacune dans une des cases du tableau.

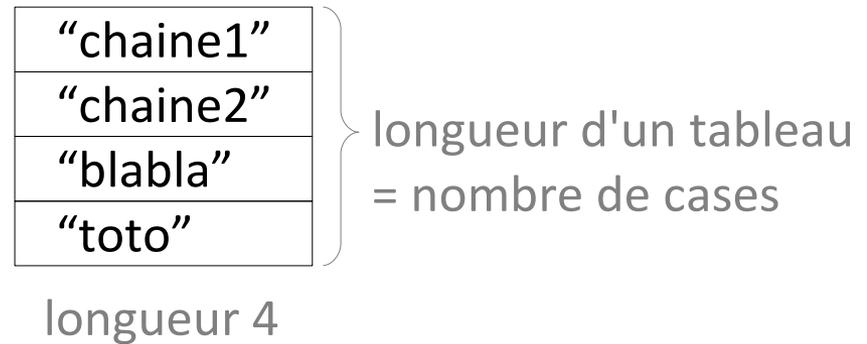
Par exemple,

Un **tableau d'entiers** :



longueur 6

Un **tableau de chaînes de caractères** :



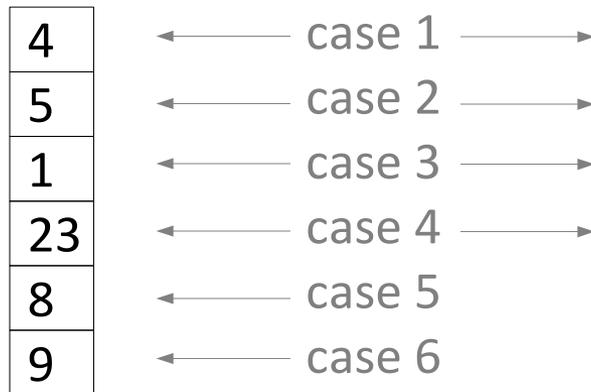
# Les tableaux

*en pseudo-code*

Les tableaux sont des variables qui contiennent **plusieurs variables de même type**, stockées chacune dans une des cases du tableau.

Par exemple, Variables : *tableau1*, un tableau d'entiers,  
*tableau2*, un tableau de chaînes de caractères

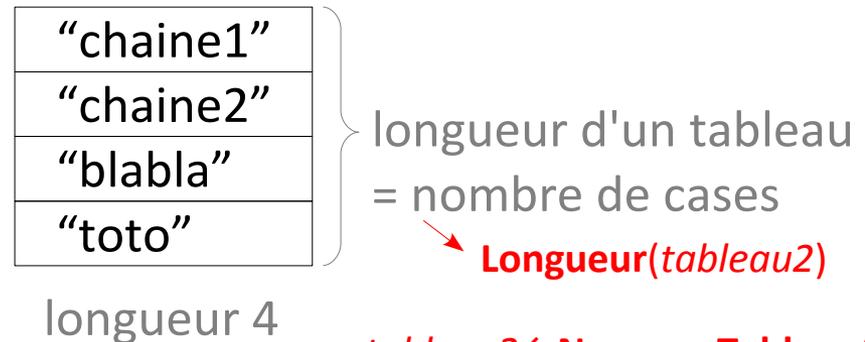
Un tableau d'entiers :



longueur 6

```
tableau1 ← NouveauTableau(6)
Case(tableau1,1) ← 4
Case(tableau1,2) ← 5
Case(tableau1,3) ← 1
Case(tableau1,4) ← 23
Case(tableau1,5) ← 8
Case(tableau1,6) ← 9
```

Un tableau de chaînes de caractères :



```
tableau2 ← NouveauTableau(4)
Case(tableau2,1) ← "chaine1"
Case(tableau2,2) ← "chaine2"
Case(tableau2,3) ← "blabla"
Case(tableau2,4) ← "toto"
```

Plus court :

```
tableau1 ← {4,5,1,23,8,9}
tableau2 ← {"chaine1","chaine2","blabla","toto"}
```

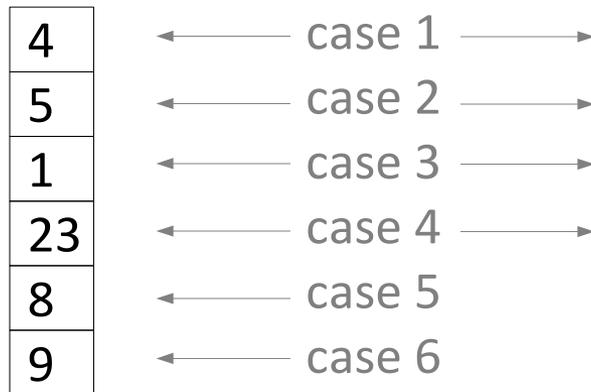
# Les tableaux

*en Java*

Les tableaux sont des variables qui contiennent **plusieurs variables de même type**, stockées chacune dans une des cases du tableau.

Par exemple, `int[] tableau1; String[] tableau2;`

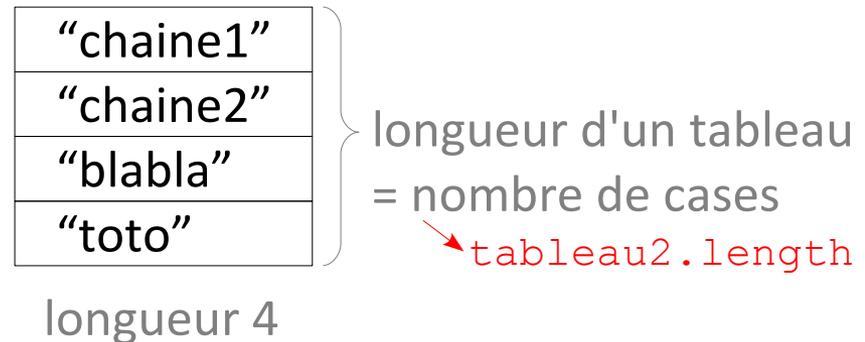
Un tableau d'entiers :



longueur 6

```
tableau1=new int[6];  
tableau1[0]=4;  
tableau1[1]=5;  
tableau1[2]=1;  
tableau1[3]=23;  
tableau1[4]=8;  
tableau1[5]=9;
```

Un tableau de chaînes de caractères :



```
tableau2=new String[4];  
tableau2[0]="chaine1";  
tableau2[1]="chaine2";  
tableau2[2]="blabla";  
tableau2[3]="toto";
```

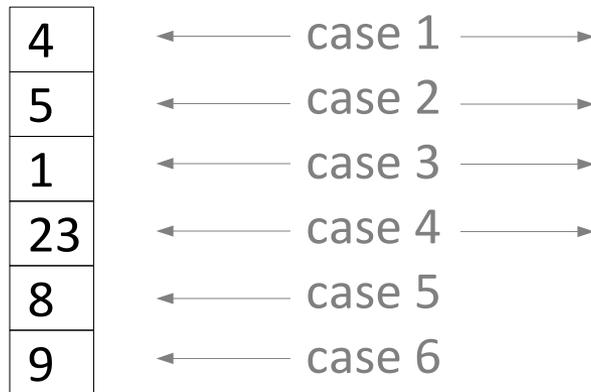
# Les tableaux

*en Java*

Les tableaux sont des variables qui contiennent **plusieurs variables de même type**, stockées chacune dans une des cases du tableau.

Par exemple, `int[] tableau1; String[] tableau2;`

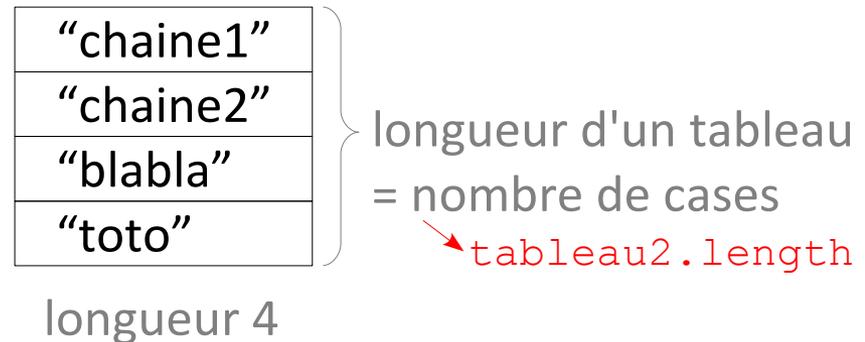
Un tableau d'entiers :



longueur 6

```
tableau1=new int[6];
tableau1[0]=4;
tableau1[1]=5;
tableau1[2]=1;
tableau1[3]=23;
tableau1[4]=8;
tableau1[5]=9;
```

Un tableau de chaînes de caractères :



longueur 4

```
tableau2=new String[4];
tableau2[0]="chaine1";
tableau2[1]="chaine2";
tableau2[2]="blabla";
tableau2[3]="toto";
```

Attention, cases du tableau `t` numérotées de 0 à `t.length-1` en Java.

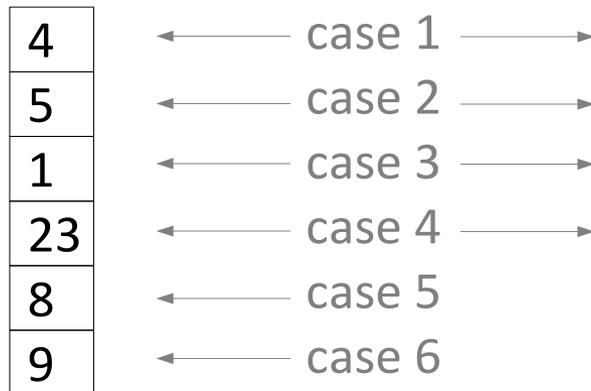
# Les tableaux

*en Java*

Les tableaux sont des variables qui contiennent **plusieurs variables de même type**, stockées chacune dans une des cases du tableau.

Par exemple, `int[] tableau1; String[] tableau2;`

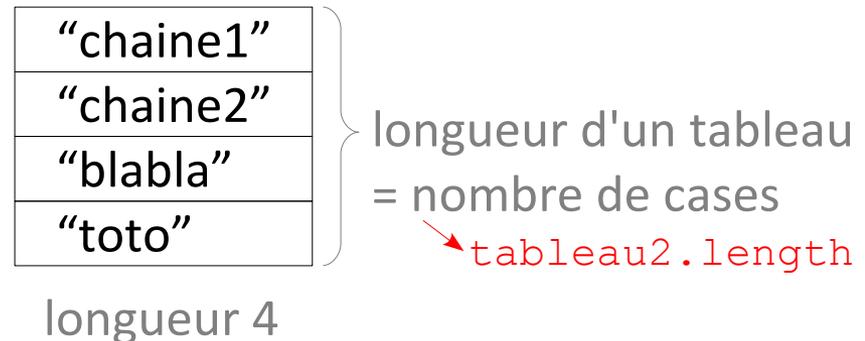
Un tableau d'entiers :



longueur 6

```
tableau1=new int[6];  
tableau1[0]=4;  
tableau1[1]=5;  
tableau1[2]=1;  
tableau1[3]=23;  
tableau1[4]=8;  
tableau1[5]=9;
```

Un tableau de chaînes de caractères :



```
tableau2=new String[4];  
tableau2[0]="chaine1";  
tableau2[1]="chaine2";  
tableau2[2]="blabla";  
tableau2[3]="toto";
```

Plus court (**déclaration + initialisation**) :

```
int[] tableau1 = {4,5,1,23,8,9};  
String[] tableau2 = {"chaine1","chaine2","blabla","toto"};
```

# Les tableaux

---

Pour lire le contenu d'un tableau...  
il faut une **boucle pour aller lire chaque case** !

Si le tableau a été prévu trop court au début, **impossible de changer sa longueur**... il faut une boucle pour le recopier dans un tableau plus grand !

Possibilité de créer des **tableaux de tableaux**...

Manipulation et expériences en TD/TP...

# Plan du cours 3 – Tableaux et boucles

---

- Résumé des épisodes précédents
- Méthodologie : lire et comprendre un algorithme
- Méthodologie : concevoir un algorithme
- Les tableaux
- Lecture du contenu d'un tableau

# Affichage du contenu d'un tableau d'entiers

---

Algorithme **AfficheTableau**

```
public static AfficheTableau( tableau1) {
```

# Affichage du contenu d'un tableau d'entiers

Algorithme **AfficheTableau**

**Variable d'entrée** : tableau d'entiers *tableau1*

**Variable** : entier *i*

Début

$i \leftarrow 1$

Tant que  $i < \text{Longueur}(\text{tableau1})+1$  faire :

**Affiche**(Case(*tableau1*,*i*))

$i \leftarrow i+1$

Fin TantQue

Fin

```
public static void AfficheTableau(int[] tableau1){
    //Afficher les cases du tableau tableau1
    int i;
    i = 0;
    while (i<tableau1.length){
        System.out.println(tableau1[i]);
        i = i+1;
    }
}
```