Algorithmique & Programmation

La notion de tableau

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

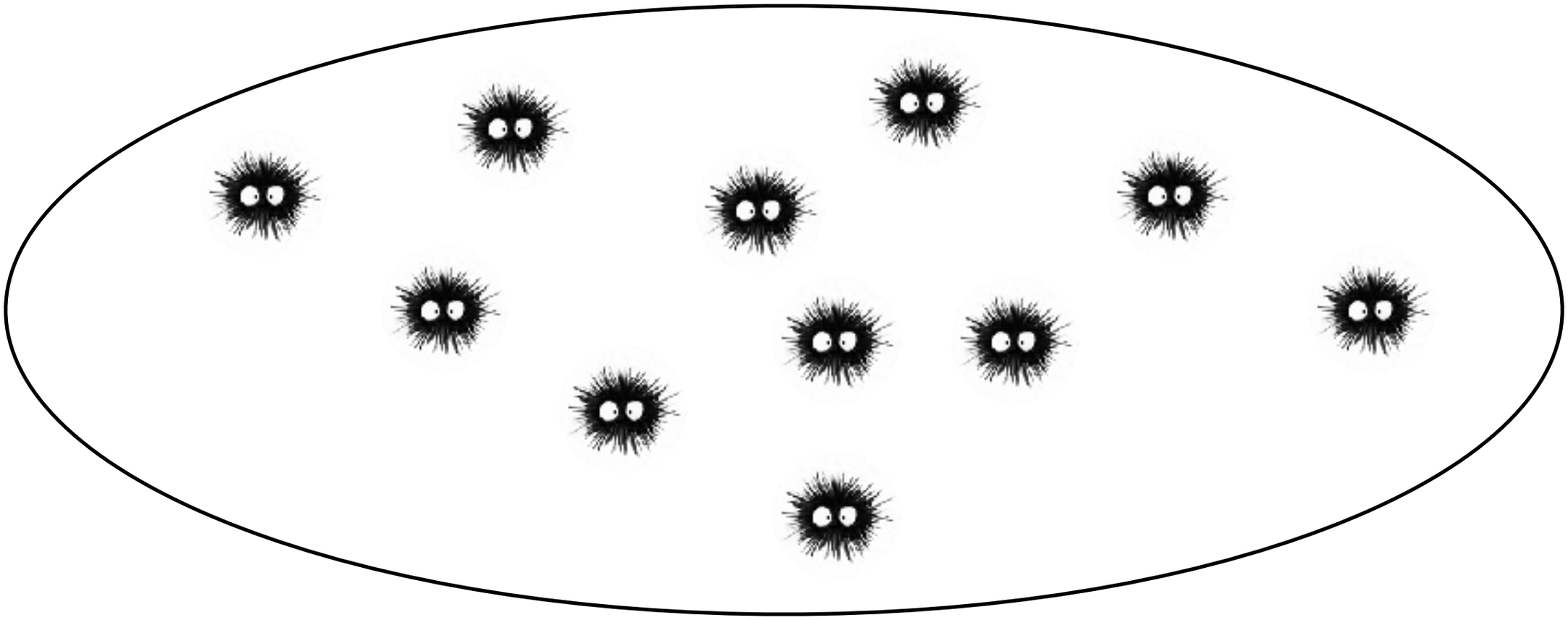
Saison 2023-2024 / R1.01+R1.02 : Initiation au développement

Ensemble d'informations

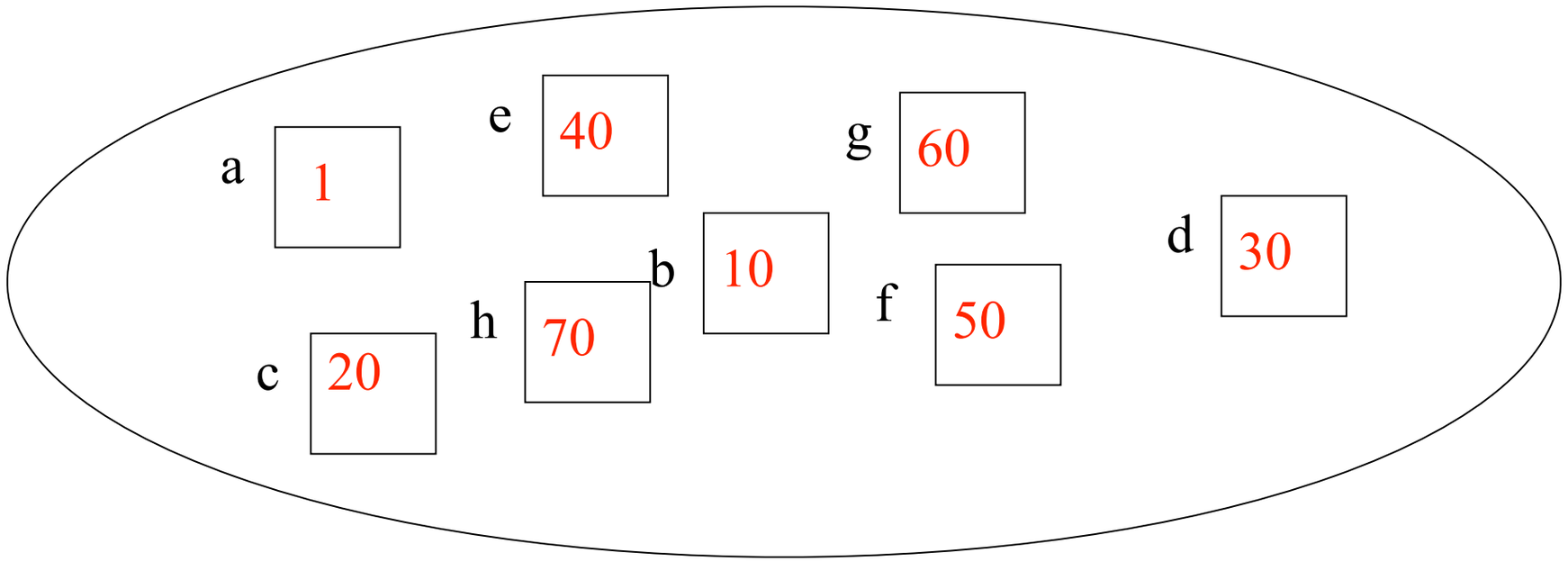
- Déclarer `n` variables du même type ?
- Actuellement, plutôt laborieux :
  - `int a, b, c, d, e, f, g, h, i, j, k;`
- Initialiser ces `n` variables est d'autant plus laborieux :
  - `a = 1; b = 10; c = 20; d = 30; e = 40` …
- Nécessité d'une structure particulière !
- Après les structures de contrôle, notre première structure de données :

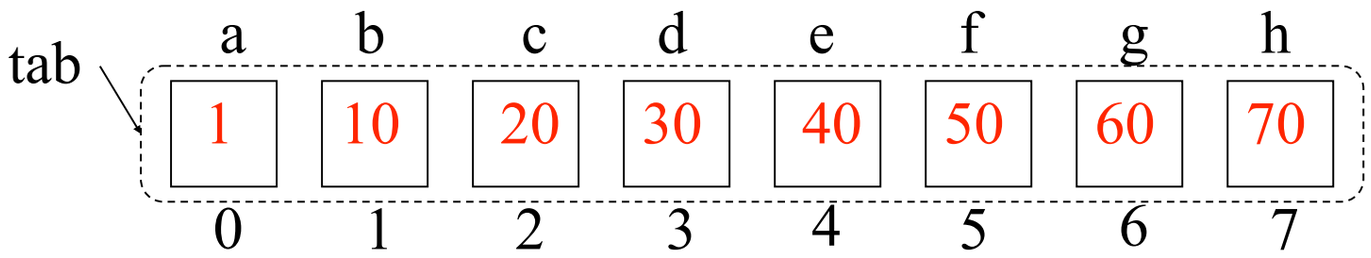
Les TABLEAUX

Ensemble de données



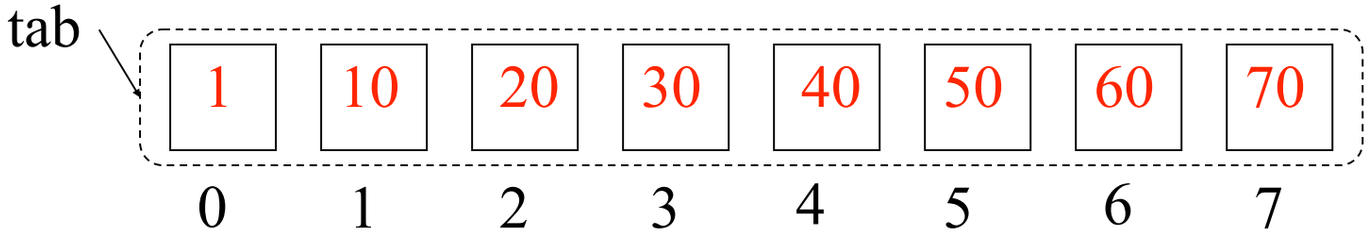
Comment connaître **le nombre d'élements** de cet ensemble ?
Comment **identifier un élément** dans l'ensemble ?
Comment **accéder à un élément** de l'ensemble ?
Comment **ajouter/enlever un élément** dans l'ensemble ?

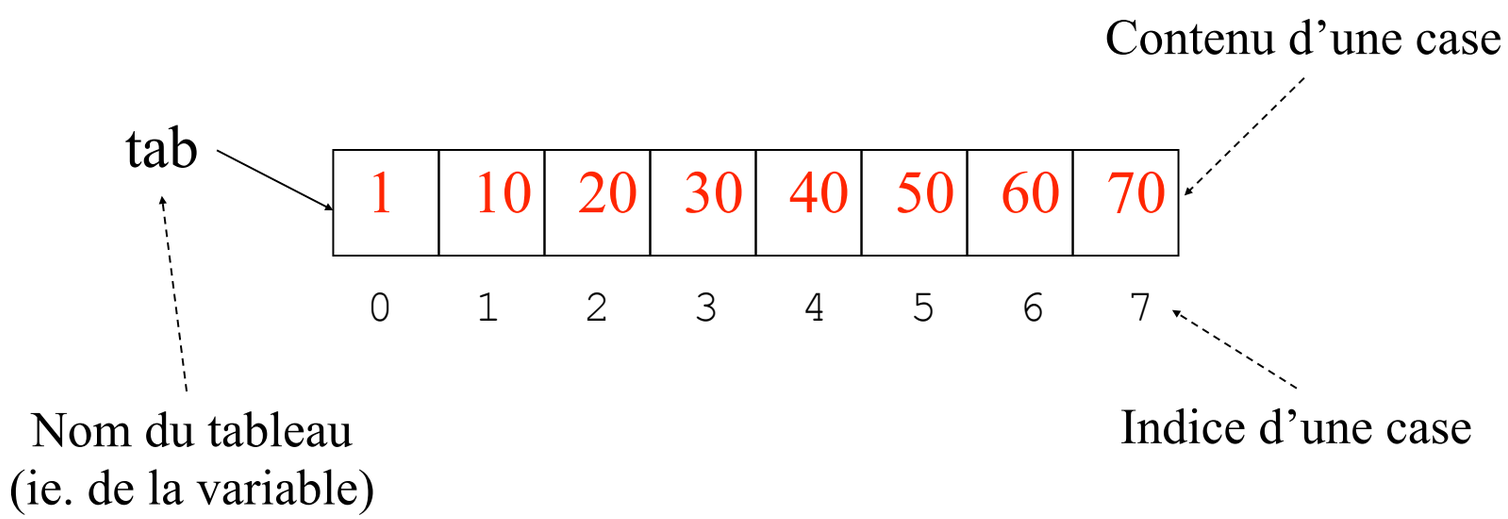

a
1
e
40
g
60
d
30
b
10
h
70
f
50
c
20

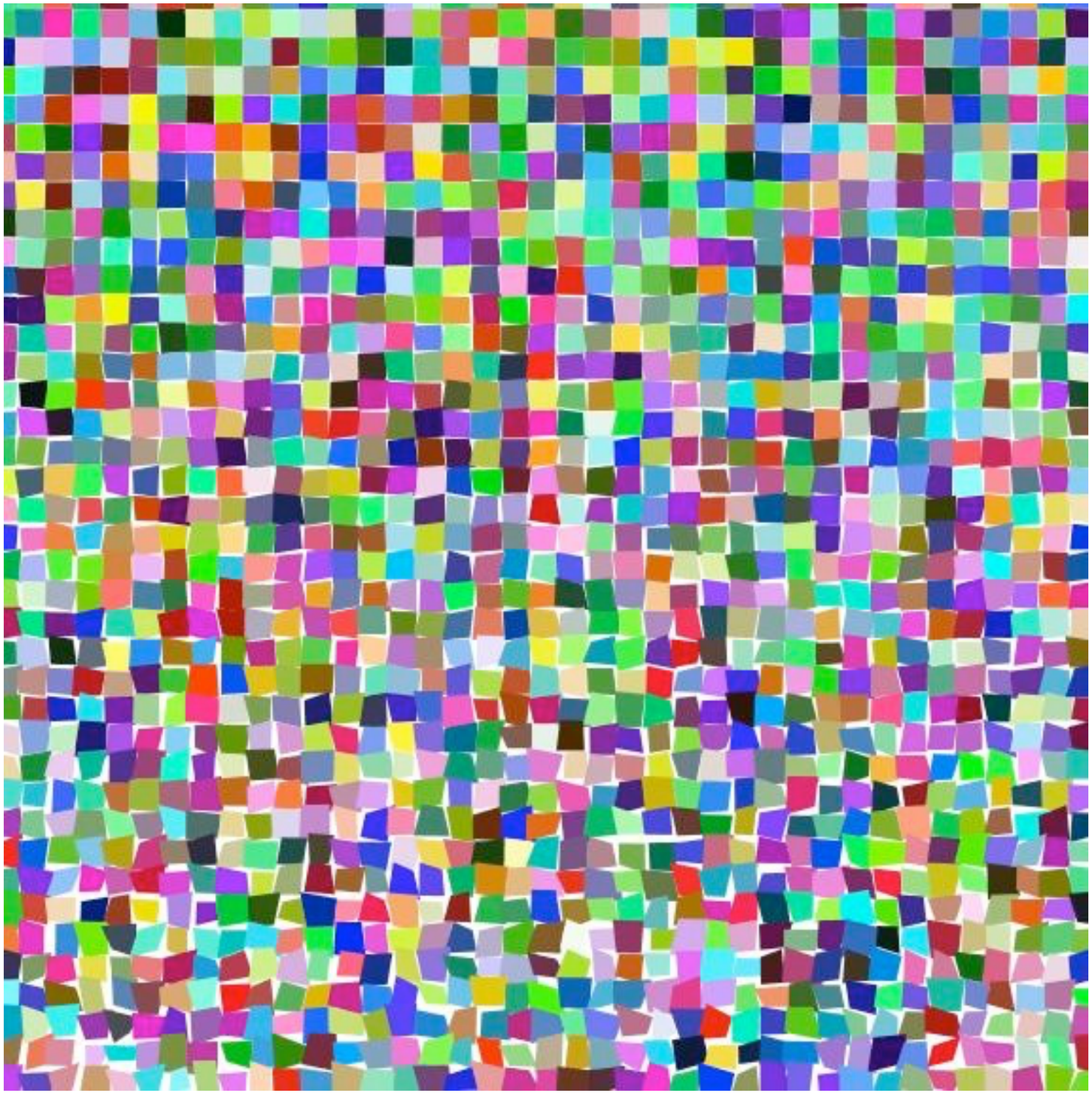

tab
a b c d e f g h
1 10 20 30 40 50 60 70
0 1 2 3 4 5 6 7


Capucines
Dahlias
Aubergines Variées
Chou Bâtif d'Etmp
Bégonias Variés
Tomates variées
Haricots d'Espagne
Pensées
Asperges variées
Liserons
Epinard de Viroflay
Chicorée de Bruffec
Chou Vilbas à pied court hâtif
Carotte courte Hollet
Haricot Empereur de Russie
Cardons Variés
Pois Prince Albert
Melons variés
Gérédine Maraval

Structure d'un tableau







Should array indices start at 0 or 1? My compromise of 0.5 was rejected without, I thought, proper consideration.

Stan Kelly-Bootle

Opérations sur les tableaux

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

Déclarer un tableau

- Déclaration d'un tableau
  - Préciser le type des données
  - `<type>[] <nomVariable>;`
- Exemples de déclarations:
  - `int[] notes;`
  - `boolean[] resultats;`
  - `char[] mot;`

(Le tableau n'a pas encore "d'existence réelle")

Allocation ou création d'un tableau

- Préciser la taille du tableau (nb éléments)
- `<var> = new <type>[<taille>];`

  Ex: `notes = new int[5]; resultats = new boolean[10]`

- ou en enchaînant définition et allocation:

  Ex: `char[] mot = new char[20];`

  Ex: `String[] libelles = new String [10];`

  Ex: `String[] mots = new String []{"bon", "jour"};`

Accéder à une valeur

- Pour accéder à une case, il faut connaître son indice
  - `<nomTableau>[<indice>]` : retourne le contenu de la case d'indice `<indice>`
- Exemple d'accès au contenu d'une case:
  - `notes[4]` : retourne le contenu de la case d'indice 4 (ie. la cinquième case !)
  - `resultats[0]` : retourne le contenu de la case d'indice 0 (ie. la première case du tableau !)

Modifier une valeur

- Pour modifier une case, il suffit de préciser la nouvelle valeur et un indice, puis d'utiliser une affectation « classique »
  - `<nomTableau>[<indice>] = <valeur>` : range la valeur <valeur> dans la case d'indice <indice>
- Exemple de modification du contenu d'une case:
  - `notes[4] = 5` : range la valeur 5 dans la case d'indice 4 (ie. la 5ème case !)
  - `resultats[0] = true` : range la valeur VRAI dans la case d'indice 0 (ie. la première case du tableau !)
  - `notes[4] = notes[5]` : range le contenu de la case d'indice 5 dans la case d'indice 4

Taille d'un tableau

- Obtenir la taille d'un tableau
  - `int length(<typeTableau> t)` : retourne la taille du tableau `t`
- Taille = nombre d'éléments que peut contenir le tableau
- Exemple d'accès à la taille d'un tableau:
  - `length(notes)` : retourne la valeur 5
  - `length(resultats)` : retourne la valeur 10
  - `length(chaines)` : retourne la valeur 20

Erreurs classiques ...

- En `Java`, la 1ère case a pour indice 0 !
- L'algorithme `iJava` s'arrête sur une erreur si
  - On accède à une case dont l'indice n'est pas valide
  - On range une valeur d'un type différent de celui du tableau
  - On accède au contenu d'une case n'ayant pas été initialisée

Notion de fonction

- **Paramètre = un type + un mode de passage**
- Modes de passage de paramètres:
  - en *lecture* ou *valeur* (lisible mais NON modifiable)
  - en *lecture/écriture* ou en *référence* (lisible ET modifiable)
- En `iJava`, les modes sont prescrits dans le langage en dépendant des types des paramètres

Passage par valeur

- Tous les types “simples” (dits *primitifs*) sont passés par valeur: `byte`, `short`, `int`, `long`, `float`, `double`, `boolean`, `char`.
- Pour l’instant, laissons `String` avec les types primitifs
- Passer une information par valeur revient à en faire une copie (cf. activité débranché en pré-TD5)
- On ne peut donc modifier la valeur dans la fonction !

Passage par référence

- Tous les autres types sont passés par référence : les tableaux (`String` mais oublions cela pour l'instant).
- Passer une information en référence revient à transmettre l'adresse de cette information en mémoire
- Plusieurs variables peuvent donc référencer la même case mémoire ...

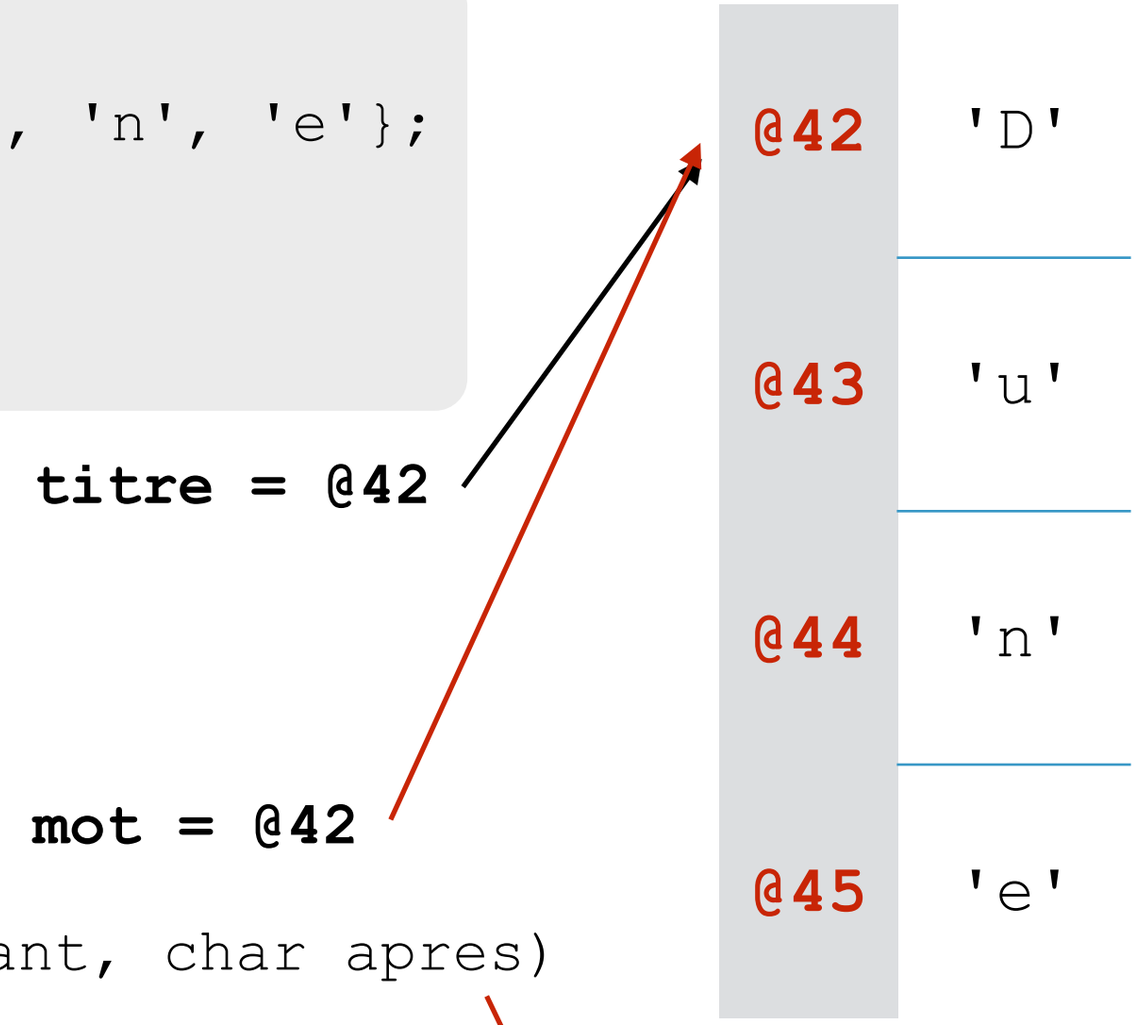
Passage par référence

```
class PassageRéférence extends Program {

  boolean remplace(char[] mot, char avant, char apres) {
    boolean changement = false;
    for (int idx=0; idx<length(mot); idx++) {
      if (mot[idx] == avant) {
        mot[idx] = apres;
        changement = true;
      }
    }
    return changement;
  }

  void algorithm() {
    char[] titre = new char[]{'D', 'u', 'n', 'e'};
    println(remplace(titre,'u','o');
    println(titre);
  }

}
```

Quels affichages sont produits ?

Que se passe-t-il en mémoire ?

```
void algorithm() {
  char[] titre = new char[]{'D', 'u', 'n', 'e'};
  println(remplace(titre,'u','o');
  println(titre);
}
```



Lors de l'appel à la fonction `remplace`, les valeurs des paramètres sont copiées

```
boolean remplace(char[] mot, char avant, char apres)
```

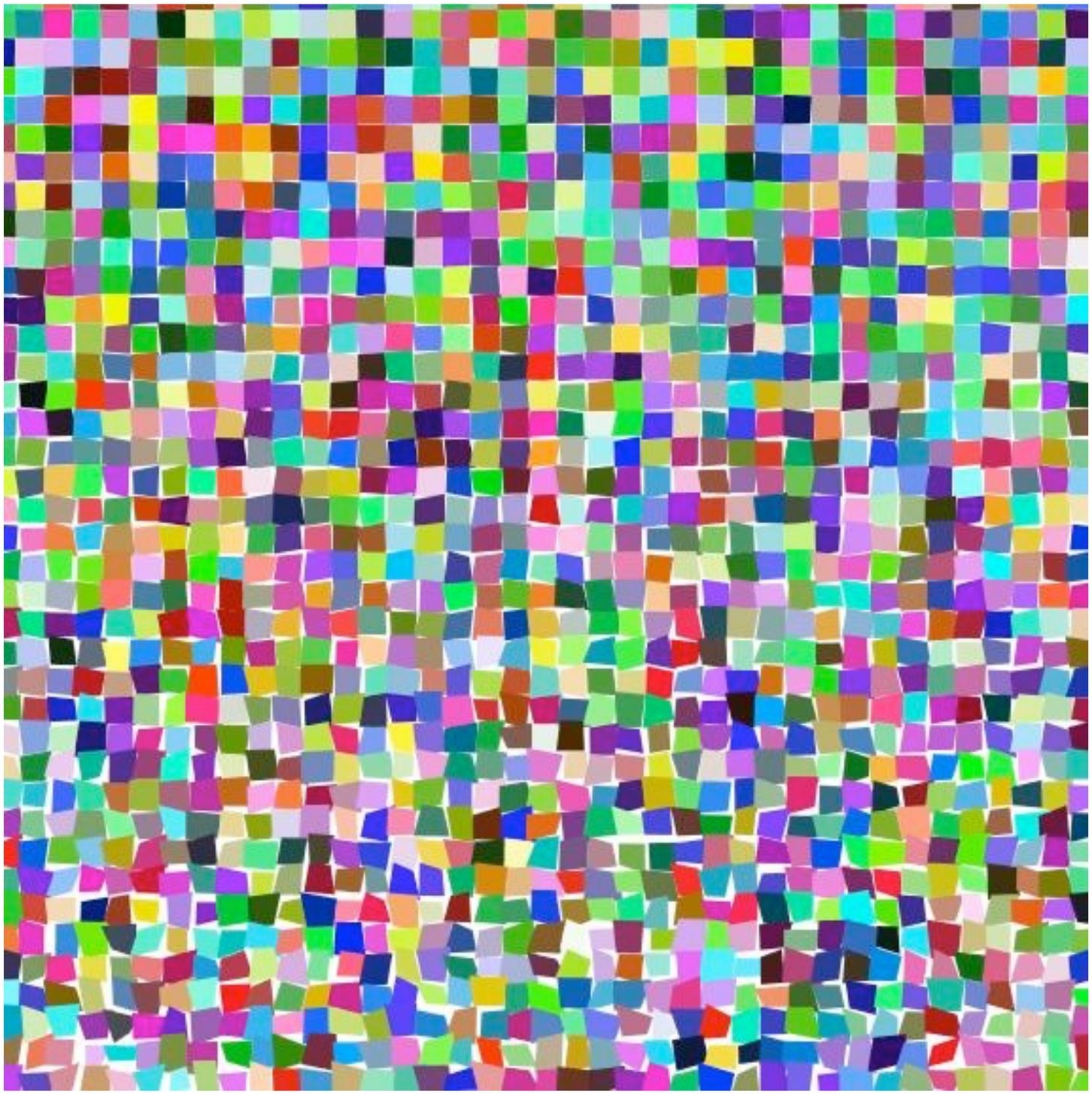
```
mot = @42        avant = 'u'        apres = 'o'
```

Synthèse: les tableaux

- **La déclaration d'un tableau**
  - identifie le type des éléments stockés
- **L'allocation** ou la création d'un tableau
  - précise le nombre d'éléments stockable
- **L'initialisation** des éléments d'un tableau
  - définie les valeurs initialement présentes
- **L'utilisation** d'un tableau
  - utilisation d'indice pour lire ou modifier une valeur
- L'accès à **la taille d'un tableau**
  - obtenir le nombre maximum d'éléments que peut contenir le tableau
- Lorsque l'on passe un tableau en paramètre, la fonction appelée peut modifier son contenu car on partage le même tableau !

Exemple

- On désire saisir 5 notes et calculer leur moyenne
- On suppose les notes valides
- On sépare la phase de saisie de la phase de calcul de la moyenne
- On souhaite utiliser un tableau



Doing linear scans over an associative array is like trying to club someone to death with a loaded Uzi.

Larry Wall (Perl creator)