

Chapitre 4 : Les tableaux et les matrices

Juan Carlos QUEZADA

INSA Strasbourg

Tableaux

Définition

- Représenter un "ensemble" de valeurs de même type
- chaque valeur est identifiée dans le tableau par un indice

Les tableaux informatiques sont très proches de la notion de vecteur en mathématique.

Exemple tableau

Tableau de 5 flottants	:	2.0	-1.2	2.0	0.0	5.3
		↑	↑	↑	↑	↑
indices	:	0	1	2	3	4

Utilisation des tableaux

- 1 Déclaration : type de tableau
tableau en <type des éléments> []
- 2 Création d'un tableau :
nouveau tableau de <nombre d'éléments> <type des éléments>
- 3 Accès aux éléments :
<tableau> [<indice>]
- 4 retrouver la taille d'un tableau :
taille(<tableau>)

Pseudo-langage

Variable t en entiers []

nouveau t de 20 entiers

N ← taille(t)

Déclaration et création de tableaux en Java

Déclaration

```
type [] nom;
```

Exemple

```
int[] t;
```

Cette déclaration précise que t est destiné à contenir la référence à un tableau d'entiers. Le nombre d'éléments du tableau n'est pas précisé, et pour le moment, aucune valeur n'a été attribuée à t .

Création

Une fois déclaré, un tableau doit être créé avec l'instruction *new*, qui initialise les éléments du tableau à zéro :

```
nom = new type [taille];
```

Exemple

```
t= new int [5];
```

Les tableaux en Java

Déclaration et création

Il est possible de déclarer et créer un tableau en « fusionnant » les 2 instructions précédentes.

```
type [] nom = new type [dimension];
```

Exemple

```
int[] t = new int[5];
```

Accès individuel aux éléments d'un tableau

Pour accéder à un élément d'un tableau, on utilise le nom du tableau suivi de l'indice associé entre crochets, en tenant compte que la première position correspond à l'indice 0.

```
t[0] = 15 ; // j'affecte 15 au premier élément  
t[3] = 0 ; // et un 0 à au quatrième élément du tableau
```

Les tableaux en Java

Affectation de tableaux

Le langage JAVA permet aussi de manipuler globalement des tableaux. Considérons le code suivant pour créer deux tableaux d'entiers, *t1* et *t2*.

```
int[] t1 = new int[3];
for (int i=0 ; i<3 ; i++){
    t1[i] = i ;
}
int[] t2 = new int[2] ;
for (int i=0 ; i<2 ; i++){
    t2[i] = 10 + i ;
}
t1=t2;
```

Dorénavant, *t1* et *t2* désignent le même tableau. Il est important de remarquer que l'affectation de références de tableaux n'entraîne pas de copie des valeurs des éléments.

Les tableaux en Java

La taille d'un tableau : *length*

- La déclaration d'un tableau (création d'une référence) ne précise pas la taille, et cette taille peut évoluer au fil de l'exécution d'un programme, par des instructions *new*.
- *length* permet de connaître le nombre d'éléments d'un tableau de référence donnée

Exemple

```
int[] t = new int[3] ;  
System.out.println(« taille de t : » + t.length) ;  
// affiche 3  
t = new int[5] ;  
System.out.println(« taille de t : » + t.length) ;  
// affiche 5
```

Matrices

Définition

Ce sont de tableaux à plusieurs indices (plusieurs dimensions)

Utilisation :

Pseudo-langage

Tableau `t[2][3]` en Entier

Les matrices en Java

Déclaration et création

```
type[][] nom = new type [nombre de lignes][nombre de colonnes]
```

Exemple

```
int[][] t = new int [12][24] // matrice de 12 x 24 entiers
```

De même que pour les tableaux à une dimension, une matrice peut être déclarée puis créée ou déclarée et créée en une seule instruction.

Nombre de lignes et de colonnes d'une matrice

- `t.length` permet de donner le nombre de lignes de la matrice `t`
- `t[0].length` permet de donner le nombre de colonnes de la matrice `t`.

Les matrices en Java

Accès aux éléments individuels d'une matrice

De même que pour les tableaux à une dimension, on accède à un élément de matrice grâce au nom de la matrice suivi de l'indice de ligne entre crochets et de l'indice de colonne entre crochets. Attention, les indices commencent à zéro.

Traitement d'une matrice

Une matrice se traite par l'imbrication de deux boucles *pour*. L'une pour fixer la ligne et la seconde pour fixer la colonne. Par exemple, le code suivant permet de remplir une matrice 2*3 de 1 :

```
int[][] t = new int[2][3] ;
for(int i=0 ; i<2 ; i++){ //fixe la ligne
    for(int j=0; j<3; j++){//fixe la colonne
        t[i][j]=1 ; //on affecte 1 à tous les éléments
    }
}
```

Génération des nombres aléatoires

Extrait de la Javadoc de la class Math :

`Math.random()` renvoie un double plus grand ou égal à 0.0 et inférieur à 1.0

Si on veut, par exemple, générer un nombre entier entre 12 (inclus) et 29 (exclus), il faut procéder comme ceci :

```
int lower = 12;
int higher = 29;
int random = (int)(Math.random() * (higher-lower)) + lower;
```

Exercices

Exercice 1 : Simulation algorithme

Simuler le fonctionnement de l'algorithme de saisie de notes et calcul de moyenne en utilisant le tableau suivant, où une colonne est réservée à chaque variable. Ce tableau permet de visualiser les valeurs successives prises par chaque variable. En particulier, la valeur d'une variable à la fin de la simulation est la dernière valeur de la colonne correspondant à la variable en question.

Note[0]	Note[1]	Note[2]	...	Note[11]	i	Som	Moy

Exercices

Exercice 2 : 1er exercice avec les tableaux

En 1952, le recensement de la population française fournissait la distribution suivante des couples en fonction de leur nombre d'enfants :

Nombre d'enfants par couple	Fréquence
0	0.36
1	0.26
2	0.19
3	0.10
4	0.05
5	0.04

Écrire un algorithme qui calcule le nombre moyen d'enfants par couple en 1952 en France.

Exercices

Exercice 3

- Écrire un algorithme qui remplit un tableau avec les 100 premiers entiers naturels.
- Ajoutez 1 à toutes les valeurs de rang pair de ce tableau.
- Soustrayez 1 à toutes les valeurs de rang impair.
- Affichez l'intégralité du tableau.

Exercices

Exercice 4 : Histogramme

- Écrire l'algorithme qui permet de calculer un histogramme. Cet algorithme prendra comme paramètre un tableau d'entiers de n'importe quelle taille et dont les valeurs appartiennent à l'intervalle 1...100. Pour cela, on utilisera la fonction *Math.random()* qui permet de générer un nombre aléatoire.
- Le programme calculera l'histogramme de ces valeurs, c'est-à-dire, la quantité de fois que les valeurs 1...100 apparaissent dans le tableau original.
- Exemple : Pour le tableau 1-2-3-5-3-1, le programme devra calculer

valeur	fréquence
1	2
2	1
3	2
4	0
5	1
...	...
100	0

Exercices

Exercice 5 : Le crible d'Ératosthène

- Le crible d'Eratosthène permet de déterminer les nombres premiers inférieurs à une certaine valeur N . On place dans un tableau unidimensionnel `Tab` les entiers compris entre 1 et N .
- L'algorithme consiste, pour chaque élément `Tab[i]` du tableau, à rechercher parmi tous les suivants (d'indices $i+1$ à N), ceux qui en sont des multiples et à les éliminer (à les remplacer par 0, par exemple). Lorsque l'ensemble du tableau a subi ce traitement, seuls les nombres premiers du tableau n'ont pas été éliminés.
- Écrire en pseudo-langage l'algorithme de ce crible et le traduire en Java, en recherchant par exemple les nombres premiers inférieurs à 500.

Exercice 6 : Le sapin décoré

Introduction

L'objectif de cet exercice est d'afficher à l'écran un sapin de Noël décoré, dont le nombre de lignes a été entré par l'utilisateur. Le schéma suivant illustre le résultat pour 6 lignes :

```
      %  
     ...  
    .....  
   .%.%.  
  .....%  
 %%.%.%
```

L'affichage du sapin n'utilise que de simples caractères alphanumériques.

Exercice 6 : Le sapin décoré

Création et affichage d'un triangle de trois lignes

L'objectif de cette partie est d'afficher la forme suivante (triangle à trois lignes) :

```
  .  
 . .  
. . . .
```

Il est courant d'utiliser pour l'affichage d'image des tableaux à 2 dimensions : chaque case correspondant à un pixel. On va commencer construire l'algorithme correspondant à l'initialisation suivante :

```
00100  
01110  
11111
```

Exercice 6 : Le sapin décoré

Création et affichage d'un triangle de trois lignes (suite)

- Ainsi, déterminer le type de tableau à déclarer ainsi que les 2 boucles à imbriquer pour générer cette initialisation.
- Remarque : se concentrer sur les cases différentes de 0. Donner l'ensemble des éléments de raisonnements qui ont permis d'aboutir au résultat.
- A partir de l'initialisation de la question précédente, compléter l'algorithme pour générer l'affichage illustré en figure 2.

Création et affichage d'un triangle de n lignes

- Déterminer la relation entre le nombre n de lignes d'un triangle et le nombre de colonnes nécessaires.
- A partir de cette relation modifier la déclaration/initialisation du tableau de manière à ce que ce soit l'utilisateur qui rentre le nombre de lignes.
- Mettre en place une boucle de contrôle de saisie du nombre de lignes.

Exercice 6 : Le sapin décoré

Placer des décorations au hasard :

A partir des éléments précédents il est possible d'afficher un sapin "non décoré" (c.à.d. un triangle de n lignes constitué de points).

L'objectif de cette partie est d'afficher un sapin décoré c'est-à-dire d'afficher d'autres caractères que des points, le signe %, par exemple :

- Modifier l'algorithme de sorte que les cases initialisées à 1 ne contiennent pas forcément 1 mais un nombre aléatoire compris entre 1 et 6.
- Modifier la partie affichage de l'algorithme de manière à afficher un caractère donné en fonction des différentes valeurs des cases du tableau.