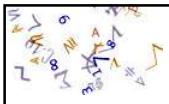
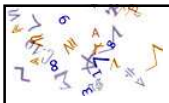


Département d'Informatique 2008/2009

Université de Cergy Pontoise

Les tableaux en JAVA à 1 dimension ...ou... à n dimensions !

UNIVERSITÉ de Cergy-Pontoise



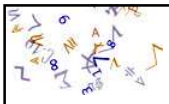
Introduction

Un tableau est une structure en mémoire dans lequel on peut stocker des données.

Toutes les données stockées dans un tableau doivent être de même type:

- **un tableau d'entiers**
- **un tableau de réels**
- **un tableau de chars**
- **un tableau de String**
- **un tableau de Voiture (si vous avez créer la classe Voiture)….mais nous verrons cela plus tard !**

UNIVERSITÉ de Cergy-Pontoise

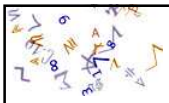


Un tableau en Java a toujours une taille fixée au départ de sa création :

Si vous indiquez au départ que le tableau a 10 cases, cela sera toujours le cas !

Si vous avez besoin de stocker un nombre variable de données….il faudra utiliser des structures comme les Objets de type Vector, ArrayList….
Nous reviendrons la dessus dans quelques séances …

UNIVERSITÉ de Cergy-Pontoise



Déclaration d'un tableau

•Pour créer un tableau, on utilise l'opérateur []

Dans les déclarations qui suivent, type est le type des éléments, type simple ou objet déjà défini.

Il n'est pas possible de "mélanger" les types dans un même tableau.

type[] tab ; déclare une variable pour référencer un tableau d'objets de type type

Exemple :

int[] tab; **double [] montableau;**

UNIVERSITÉ de Cergy-Pontoise

Création du tableau

nomdutableau= new type [taille] ;
construit (instancie) un objet tableau de taille fixée, sans l'identifier

type [] nomdutableau = new type [taille] ; le tableau est appelé nomdutableau et construit avec taille éléments de type type.

Faites bien attention au mot-clé new , nous le rencontrerons bientôt plus souvent !

Exemple :

- **String tableau[] = new String[50]; construit un tableau de 50 chaines, numérotées de 0 à 49.**
- **int toto[] = new int[10]; construit un tableau de nom toto de 10 entiers, numérotées de 0 à 9.**

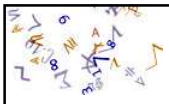
UNIVERSITÉ de Cergy-Pontoise

Est ce que vous suivez ?

1) Quelles sont les déclarations/créations de tableaux qui sont invalides ? Pourquoi ?

- ❑ **double[] tab**
- ❑ **char [] int;**
- ❑ **char [10] tableau;**
- ❑ **int [] montab= int [10];**
- ❑ **float [] letableau=new int [5];**
- ❑ **double[5] montab= new double [100];**

UNIVERSITÉ de Cergy-Pontoise



Les éléments du tableau

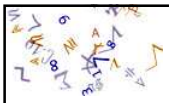
Les éléments du tableau sont indicés (numérotés) de 0 à taille –1.

Exemple :
int [] toto = new int[5]; toto est un tableau de 5 cases où les cases sont numérotées de 0 à 4 !

Si vous rencontrez l'erreur, ArrayIndexOutofBoundException….c'est que vous avez demandé une case du tableau qui n'existe pas !

Exemple : la case d'indice –1 ou la case d'indice 5 pour un tableau de 5 cases !

UNIVERSITÉ de Cergy-Pontoise



Combien a t on de cases dans ce tableau ?

La propriété length permet de connaitre la taille du tableau, très utile pour parcourir le tableau à l'aide d'une boucle :

```
double [] montab=new double[20];
System.out.println("Le nombre de cases de mon tableau est "+
montab.length) ;
```

C'est très utile pour parcourir un tableau à l'aide d'une boucle !

UNIVERSITÉ de Cergy-Pontoise

Comment savoir ce qui se trouve dans une case en particulier ?

En utilisant les crochets []…

tab[i] avec i est l'identifiant de la case qui vous intéresse

Exemple : Si mon tableau s'appelle montab et contient des entiers

5	8	12	4	16

Quel est le contenu de : tab[1] , tab[3], tab[0], tab[5] ?

UNIVERSITÉ de Cergy-Pontoise

Est ce que vous suivez ?

2) Que retourne les instructions pour le tableau montab suivant ?

1	5	78	4	5

- **montab.length**
- **montab[4]**
- **montab[1+1]**
- **montab[0]**
- **montab[6]**
- **montab[1]**

UNIVERSITÉ de Cergy-Pontoise

Remplissage du tableau

Il y a trois façons de faire :

- **en lui assignant des valeurs au départ**

```
int [] tab1 = {5, 10, 15, 20, 25};
String tab2[] = {"lundi", "mardi", "mercredi", "jeudi",
"vendredi"};
```

- **en lui assignant des valeurs après la création**

```
int [] tab1 = new int[3];
tab1[0]=5; tab1[1]=10; tab1[2]=15; tab1[3]=20; tab1[4]=25;
```

- **en lui assignant la même valeur dans chaque case après la création avec une boucle ….**

UNIVERSITÉ de Cergy-Pontoise

Parcourir un tableau et afficher ces valeurs

Pour afficher un tableau ,

System.out.println(tab); ne fonctionne pas !!!!!!!!!!

Il faut le parcourir avec une boucle ,

```
for (int i=0 ; i < tab.length ; i++)
 System.out.println("tab["+i+"] = "+ tab[i]) ;
```

UNIVERSITÉ de Cergy-Pontoise

Est ce que vous suivez ?

3) Qu'est ce qui ne va pas dans ce code ?

```
char [] montab=new char[5];
char[0]='1';

int [] montableau =new int[8];
montableau[4]='1';

int [] montableau =new int[8];
montableau[montableau.length]=5;

int [] montableau =new int[16];
montableau[1]=6.8f;
```

UNIVERSITÉ de Cergy-Pontoise

Tableaux à plusieurs dimensions

Java permet de créer plutôt des tableaux de tableaux.

Chaque élément d'un tableau, peut contenir un autre tableau et ainsi de suite, sans limitation de principe du nombre de dimension.

Déclaration :

```
type [] []  nomTab  = new type [taille1]  [taille2] ;
```

Exemple :

```
int [] [] tabEntier2 = new int[10] [100] ;
```

A noter que les dimensions ne sont pas forcément égales.

UNIVERSITÉ de Cergy-Pontoise

Qui est la ligne, qui est la colonne ?

Un tableau à 2 dimensions est un tableau de tableaux donc peu importe du moment que vous êtes cohérent avec vous même !

```
int[][] matrice=new int[5][6];
int[][] matrice=new int[6][5];
```

Pour accéder à une case …..

```
System.out.println(matrice[0][0]);
System.out.println(matrice[2][5]);
```

C'est le principe de la bataille navale !

UNIVERSITÉ de Cergy-Pontoise

Pour le remplir …..

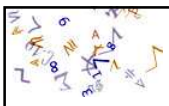
Faites attention aux crochets et à la déclaration (qui est le tableau de qui !)

```
int[][] matrice =
   {
      { 0, 1, 4, 3 } , { 5, 7, 9, 11}
   };
```

Equivalent de :

```
int[][] matrice =new int[2][4];
matrice[0][0]=0;
matrice[1][0]=1;
……..
```

UNIVERSITÉ de Cergy-Pontoise



Pour le parcourir ….. Des boucles imbriquées feront l'affaire !

```
int[][] matrice=new int[longueur][largeur];

for (int i=0; i<longueur; i++){
for (int j=0; j<largeur; j++){
        System.out.println(matrice[i][j]);
}}
```

UNIVERSITÉ de Cergy-Pontoise