

Les tableaux en Java

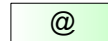
- Tableau de types primitifs et dim 1
- Tableau d'objets et dim 1
- Tableau de dimension 2 de types primitifs, rectangulaire et non-rectangulaires
- Travail à réaliser

1

Tableau de types primitifs

- `double[] monTab = new double[4];`

monTabPrimitifsNum



lien

0	0.
1	0.
2	0.
3	0.

Le tableau est un objet

XH

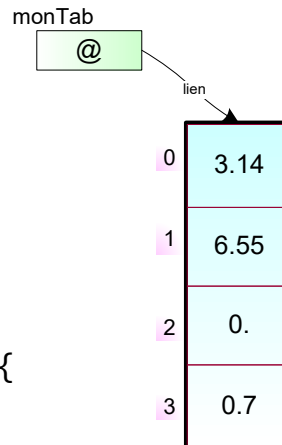
2



Tableau de types primitifs

- `monTab[0] = 3.14 ;`
- `monTab[1] = 6.55 ;`
- `monTab[3] = 0.7 ;`

```
//Itération sur le tableau  
for( int i=0; i<monTab.length; i++){  
    sysout.println(monTab[i]);  
}
```



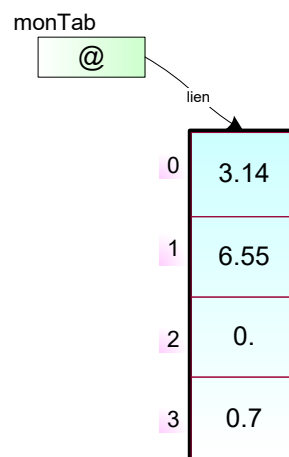
XH

3



Autre écriture pour l'initialisation du tableau : initialiseur

- `double[] monTab = {3.14, 6.55, 0., 0.7} ;`



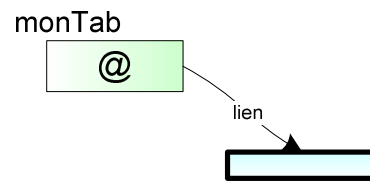
XH

4



Tableau de longueur nulle

- `double[] monTab = {} ;`



XH

5



XH

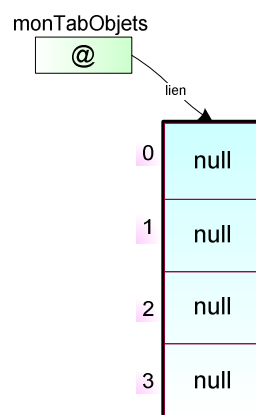
6

Tableau d'objets de dimension 1

7

Tableau d'objets

- `Livre[] monTab;`
- `monTab = new Livre[4];`



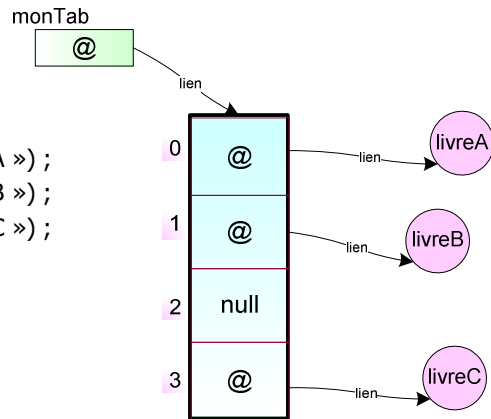
XH

8



Tableau d'objets

- `monTab[0] = new Livre(« livreA »);`
- `monTab[1] = new Livre(« livreB »);`
- `monTab[3] = new Livre(« livreC »);`



XH

9



Itération sur le tableau d'objets

```
//Itération sur le tableau
for( int i=0; i<monTab.length; i++){
    sysout.println(monTab[i]);
}
```

```
//foreach Java5
for( Livre var : monTab) {
    sysout.println( var );
}
```

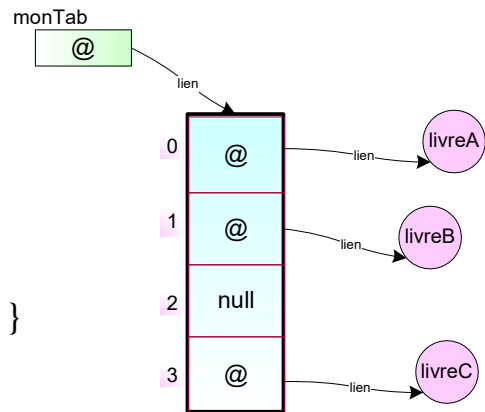
XH

10



Autre écriture pour faire la même chose ...

```
Livre [] monTab = {  
    new Livre( "livreA"),  
    new Livre( "livreB"),  
    null,  
    new Livre( "livreC") }
```



XH

11



XH

12



Tableau à deux dimensions de type primitif

Rectangulaire (matrice)
Non-rectangulaire

13



Tableau à deux dimensions de type primitif rectangulaire

- //matrice de 3 lignes et 4 colonnes
- **int[][] tab2 = new int[3][4];**
- **System.out.println("remplissage du *tableau avec qq valeurs*");**
- **tab2[1][2] = 67;**
- **tab2[2][3] = 12;**

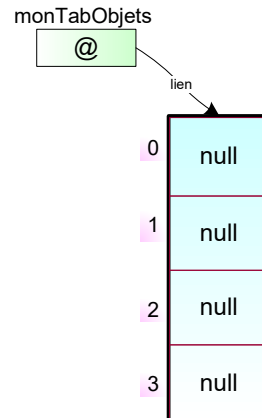
XH

14



Tableau à deux dimensions de type primitif non-rectangulaire

■ `int [][] monTab = new int [4][] ;`



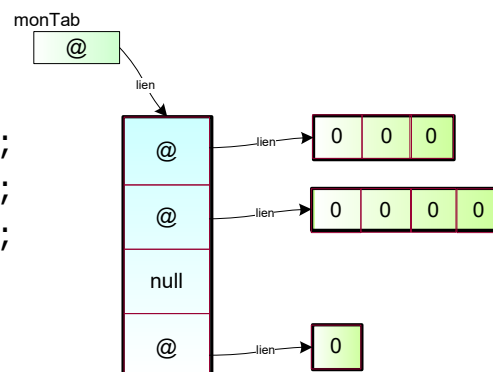
XH

15



Tableau à deux dimensions de type primitif

- `monTab[0] = new int[3] ;`
- `monTab[1] = new int[4] ;`
- `monTab[3] = new int[1] ;`



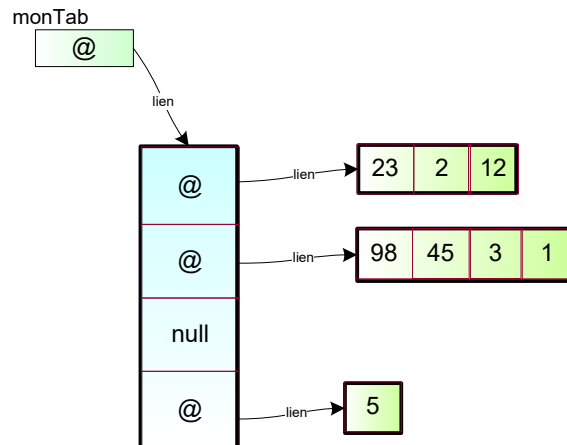
XH

16



Tableau à deux dimensions de type primitif

- `monTab[0][0] = 23;`
- `monTab[0][1] = 2;`
- `monTab[0][2] = 12;`
- `monTab[1][0] = 98;`
- `monTab[1][1] = 45;`
- `monTab[1][2] = 3;`
- `monTab[1][3] = 1;`
- `monTab[3][0] = 5;`



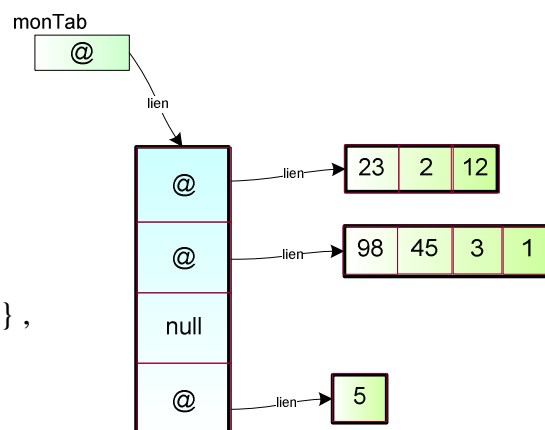
XH

17



Autre écriture pour faire la même chose

```
int [][] monTab =
    { { 23, 2, 12 } ,
      { 98, 45, 3, 1 } ,
      null ,
      { 5 }
    };
```



XH

18



Itération sur le tableau dim2

```
//Itération sur le tableau
for( int L=0; L<monTab.length; L++){
    for( int c=0; c<monTab[L].length; c++) {
        sysout.println(monTab[L][c]);
    }
}
```

XH

19



Et le foreach?

```
System.out.println("Voici le contenu du tableau :");
for( int[] L : tab2){
    if ( L != null ){
        for( int c : L)
            System.out.print(" " + c);
    }
    System.out.println();
}
```

- //Voir aussi la démo fournie sur le site ftp habituel

XH

20



Remarque sur la méthode Arrays.sort()

- sort() vérifie si les éléments de la collection sont Comparable (instance of)
- D'où, on est sûr qu'il y a une méthode compareTo() qui sera utilisée

XH

21



Travail à réaliser

- Essayez les exemples des slides

XH

22