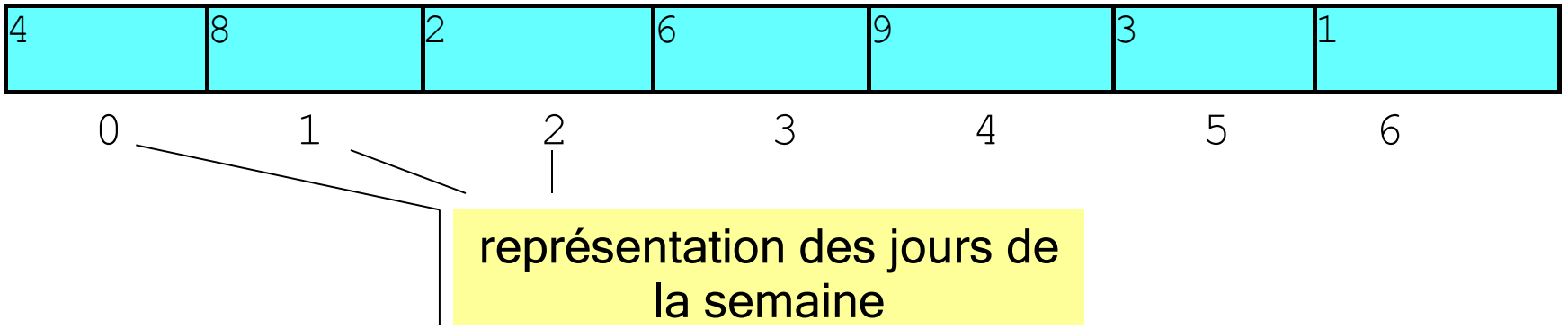
Chapitre 6 : les Tableaux

Notion de tableau
Les tableaux en Java
Affectation, égalité entre tableaux
Tableaux multidimensionnels

Notion de tableau

Pour pouvoir écrire des programmes un peu plus intéressants, il est nécessaire d'appliquer des traitements à des séquences de données. Une manière classique est de les rassembler dans un tableau.

Dans l'exemple qui suit, on réunit dans un tableau le nombre d'heures d'ensoleillement de chaque jour de la semaine.

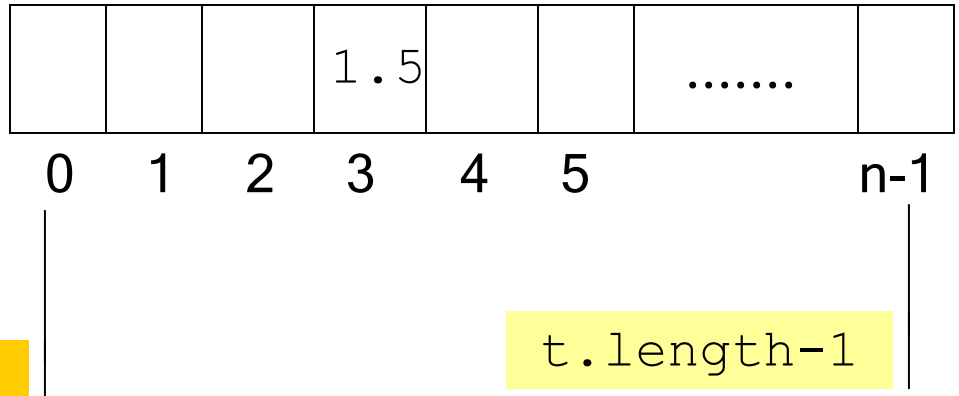


Tableaux Java (1/2)

- Un tableau est une structure de données qui réunit des valeurs (données) d'un même type (le type `int` dans l'exemple ci-dessus).
- On peut le voir comme une suite de cases contiguës repérées (indicées) par un entier (`int`). Le premier indice ayant pour valeur `0`.
- Un tableau constitue une nouvelle valeur. Or toute valeur doit appartenir à un type. Il est donc nécessaire de définir un nouveau type auquel ces éléments (ces valeurs) appartiendront.
- Dans l'exemple, on définit le type `int[]` comme un ensemble de tableaux contenant des entiers (de type `int`). Les indices de `0` à `6` correspondront respectivement aux jours de la semaine de `lundi` à `dimanche`.
- Il est alors possible de déclarer une variable (`uneSemaine`) de ce nouveau type, d'enregistrer des valeurs dans les cases de ce tableau, de sélectionner une case connaissant son indice (`uneSemaine[i]`).

Tableau Java (2/2)

```
float[] t = new float[n]
```




```
// si t[3] = 1.5f
t[3] = t[3] + 2.5f
// alors t[3] = 4.f
```

Un indice en dehors des bornes du tableau (`0..length-1`) provoque une erreur à l'exécution.
l' exception :
`ArrayIndexOutOfBoundsException`
est levée

Tableaux : création (1/2)

- Déclaration

```
boolean[] T;
```

allocation d'un mot mémoire

- Création

```
new boolean[5];
```

allocation en mémoire de 5 composantes de type booléen (1 octet * 5)

- Déclaration avec initialisation

```
boolean[] T = {true,false,false};
```

ou

```
boolean[] T = new boolean[3];
T[0]=true;T[1]=false;T[2]=false;
```

Tableaux : création (2/2)

ATTENTION

```
int[] source;
source = {6,78,9,0}; // impossible
```

MAIS constructions possibles :

```
int[] source = new int[]{6,78,9,0};
```

et ensuite

```
source = new int[]{12,42,9,76,0,0,65};
```

Exemple

```
public class AfficherTableau{
  public static void main(String[] args){
    int[] tab = {10,20,30,40};
    for( int i=0;i<=tab.length-1;i++)
      System.out.println("tab["+i+"] "+tab[i]);
  }
}
```

résultat affiché :

```
tab[0]=10
tab[1]=20
tab[2]=30
tab[3]=40
```

résultat affiché si erreur sur la condition de sortie de boucle : `i<=tab.length` :

```
ArrayIndexOutOfBoundsException : 4
```

Exemple

Dans l'exemple qui suit, on implante l'algorithme :

- saisie dans un tableau du nombre d'heures d'ensoleillement de chaque jour d'une semaine.

- calcul de la journée la plus ensoleillée

- affichage à l'écran du résultat

```java
-- nombre d'heures d'ensoleillement journalier d'une semaine donnée
import java.util.Scanner;
public class Soleil{
  final int LUNDI = 0;
  final int MARDI = 1;
  final int MERCREDI = 2;
  final int JEUDI = 3;
  final int VENDREDI = 4;
  final int SAMEDI = 5;
  final int DIMANCHE = 6;
  int[] uneSemaine = new int[7];
  Scanner input = new Scanner( System.in );
  -- construction du tableau uneSemaine
  for( int i=0;i<7;i++){
   System.out.print( "taper le nb d'heures de soleil" );
   System.out.print( " pour la journée " );
   System.out.print(i + " : " );
   uneSemaine[i] = input.nextInt();
   System.out.println();
  }
```

```
-- calcul du jour le plus ensoleille
 int nbHeures = uneSemaine[LUNDI];
 int unJour   = LUNDI;
 for( int x=MARDI;x<=DIMANCHE;x++){
   if (uneSemaine[x]>nbHeures){
       nbHeures = uneSemaine[x];
       unJour   = x;
   }
 }
-- affichage du résultat
 System.out.print( "le n°du jour le plus ensoleille fut : " );
 System.out.print( unJour + " avec " );
 System.out.print( uneSemaine[unJour] );
 System.out.println( " heures d'ensoleillement" );
}
```

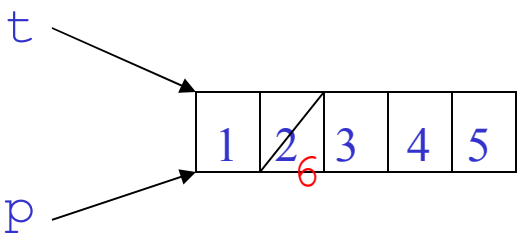
Affectation entre tableaux

```
int[] t ;
int[] p = {1,2,3,4,5};
t = p;
System.out.println("t[1]="+t[1]);
System.out.println("p[1]="+p[1]);
t[1] = 6;
System.out.println("nouveau t[1]="+t[1]);
System.out.println("nouveau p[1]="+p[1]);
```

`t` et `p` désignent le même tableau

résultat

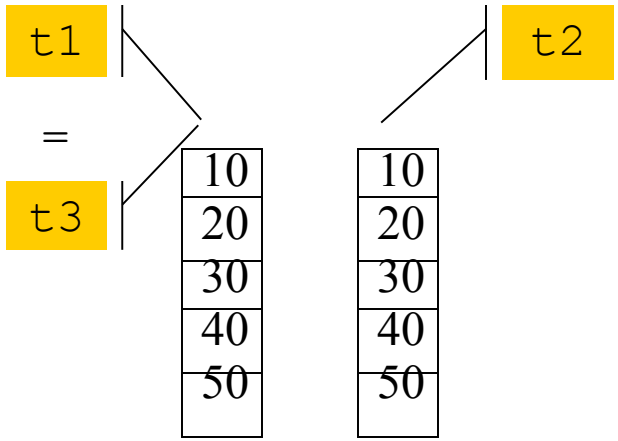
```
t[1]=2
p[1]=2
nouveau t[1]=6
nouveau p[1]=6
```



Egalité entre tableaux

La comparaison entre 2 tableaux porte sur leur référence et non sur leur contenu.

```
int[] t1 = new int[5]{10,20,30,40,50};
int[] t2 = new int[5]{10,20,30,40,50};
if(t1==t2)
  System.out.println(t1=t2);
else
  System.out.println(t1 t2);
int[] t3 = t1;
if(t3==t1)
  System.out.println(t3=t1);
else
  System.out.println(t3 t1);
```



Tableaux multidimensionnels

- Un tableau multidimensionnel est vu comme un tableau de tableau.

```
double[][] precipitation = new double[13][32];
```

- tableau de 13 composants, chacun d'entre eux étant un tableau de 32 composants. Il peut, par exemple, représenter la précipitation de pluie pour chaque jour de chaque mois d'une année.

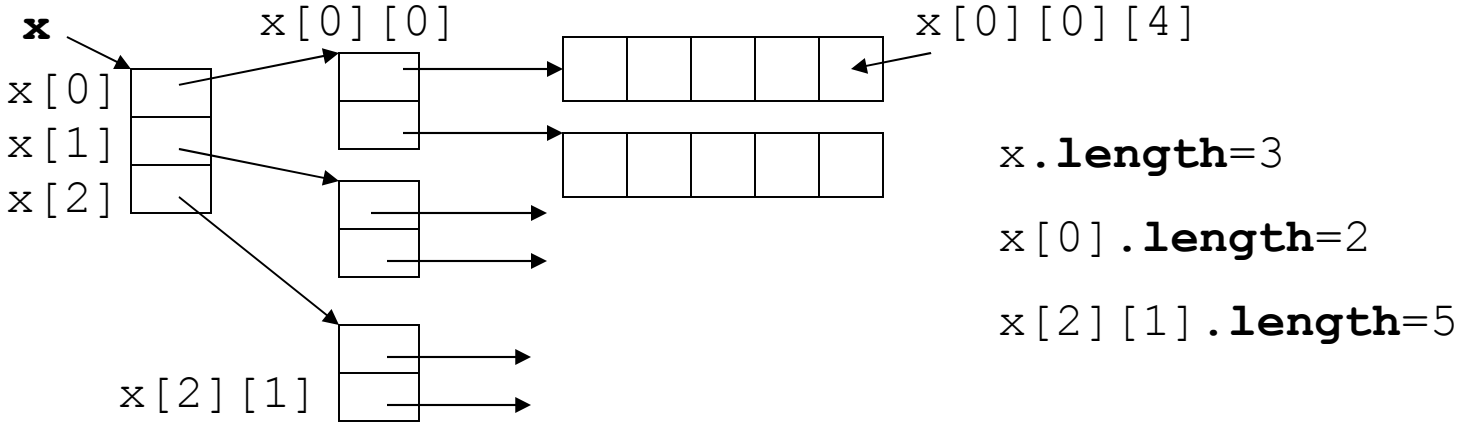
- Le niveau de précipitation du 23 mars sera représenté par :
  `precipitation[3][23]`
  à condition que les lignes et colonnes d'indice `0` soient initialisées à `0`

- La longueur du tableau représentant le mois d'avril sera connue par :
  `precipitation[4].length`

Longueur des tableaux multi-dimensionnels

- Un tableau à 2 dimensions est un tableau à 1 dimension dont chaque élément est lui-même un tableau à une dimension
- Un tableau à 3 dimensions est un tableau à 1 dimension dont chaque élément est lui-même un tableau à 2 dimensions
- Exemple :

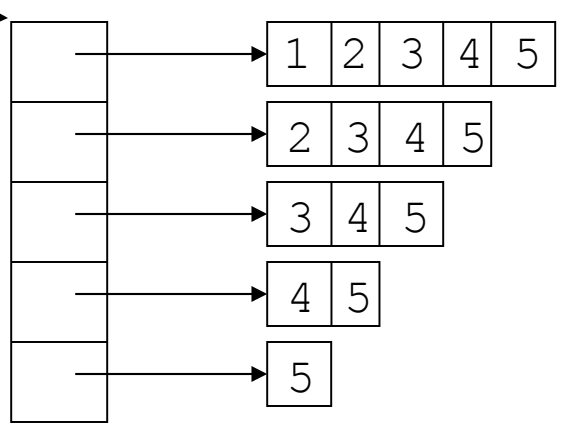
  Soit le tableau `x = new int[3][2][5];`



Tableaux asymétriques

On peut aussi construire un tableau par initialisation de ses éléments

```
int[][] x = {
   {1,2,3,4,5},
   {2,3,4,5},
   {3,4,5},
   {4,5},
   {5}
};
```



Ou encore, si l'on ne connaît pas les valeurs initiales du tableau `x` :

```
int[][] x = new int[5][];
x[0]= new int[5];
x[1]= new int[4];
x[2]= new int[3];
x[3]= new int[2];
x[4]= new int[1];
```

Exemple : plateau du Tic Tac Toe (1/3)

Un plateau de Tic Tac Toe est représenté par une matrice 3x3.

Algorithme :

```
 - remplir aléatoirement la matrice avec l'un
des deux entiers : 1 ou 0
 - afficher le plateau
 - découvrir les lignes, colonnes ou diagonales
exclusivement composées de 1 ou 0
 - afficher le résultat
```

Exemple : plateau du Tic Tac Toe (2/3)

```
import java.util.*;
public class TicTacToe{
    public static void main( String[] args ){
        Random alea = new Random();
        int[][] plateau = new int[3][3];
        for (int i=0;i<3;i++)
          for (int j=0;j<3;j++)
            plateau[i][j] = alea.nextInt(2);
        // affichage du plateau
        for (int i=0;i<3;i++){
          for (int j=0;j<3;j++)
           System.out.print( plateau[i][j] );
          System.out.println();
        }
```

```
110
010
111
```

Exemple : plateau du Tic Tac Toe (3/4)

```
// test lignes
   int[] ligne = null;
   for (int i=0;i<3;i++){
     ligne = plateau[i];
     if( ligne[0]==ligne[1] && ligne[0]==ligne[2] )
     System.out.println
       ("ligne["+i+"]="+ligne[0]+ligne[1]+ligne[2]);
   }
// test colonnes
    for (int j=0;j<3;j++){
     if( plateau[0][j]==plateau[1][j]
         &&
         plateau[0][j]==plateau[2][j] )
      System.out.println
             ("colonne["+j+"]="+
               plateau[0][j]+plateau[1][j]+plateau[2][j]
             );
   }
```

```
ligne[2]=111
colonne[1]=111
```

Exemple : plateau du Tic Tac Toe (4/4)

```
// test diagonales
   if( plateau[0][0]==plateau[1][1]
        &&
        plateau[0][0]==plateau[2][2]
      )
    System.out.println
    ("diagonale 1 ="+plateau[0][0]+plateau[1][1]+plateau[2][2]);
   if(
      plateau[0][2] == plateau[1][1]
      &&
      plateau[1][1] == plateau[2][0]
      )
    System.out.println
    ("diagonale 2 ="+plateau[0][2]+plateau[1][1]+plateau[2][0]);
  }
}
```

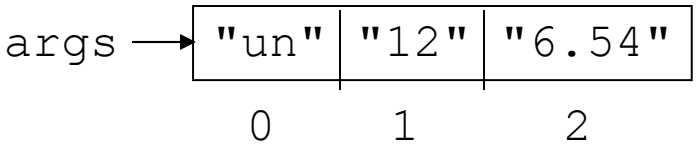
diagonale 1 =111

Retour sur la méthode `main`

La méthode `main` prend un paramètre de type `String[],` tableau de chaîne de caractères.

Ce paramètre permet de transférer des données entre la ligne de commande et le programme java

```
public class Transfert{
   public static void main(String[] args){
        for(int i=0;i<args.length;i++)
              System.out.println(args[i]);
   }
}
>java Transfert un 12 6.54
un
12
6.54
```



Retour sur la méthode `main`