

# Tableaux en Delphi

**Tableaux statiques et tableaux dynamiques**

**Ricco Rakotomalala**  
**Université Lumière Lyon 2**

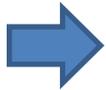
# Tableaux

(1) Collection de données de même type

(2) Accès indicé



Très utilisé pour la manipulation de vecteurs et de matrices de nombres



On peut aller plus loin et manipuler une collection de types structurés (enregistrements) et même d'objets (instances de classes)

OUI

[7 cases, valeurs entières]

12	4	0	5	74	9	47
----	---	---	---	----	---	----

NON

[types non uniformes]

12	'toto'	0	true	74	9.8	47
----	--------	---	------	----	-----	----

# Tableaux statiques

Le nombre d'éléments est défini à la compilation

Définition du type `TYPE Tvecteur = array [1..20] of double;`

Création de  
l'instance

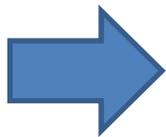
`var v : Tvecteur;`

- (1) Les valeurs du 1<sup>er</sup> et du dernier indices sont définis explicitement.
- (2) On peut aussi utiliser un type énuméré

Utilisation

`v[4] := 11.3 ;`  
...  
`writeln(v[3]);`

Les débordements  
d'indices provoquent une  
erreur.



Du fait que la taille du tableau ne peut pas être défini à l'exécution, ce type de tableau n'est quasiment plus utilisé.

# Tableaux dynamiques

Le nombre d'éléments est défini à l'exécution

Définition du type

```
TYPE TVecDyn = array of double;
```

Création de l'instance

```
var v : TVecDyn;
```

Il n'y a plus de restriction *a priori* sur la taille du tableau.

Initialisation

(un tableau dynamique est en fait un pointeur)

```
SetLength(v, 10);
```

Création d'un vecteur de 10 cases. Le 1<sup>er</sup> indice est 0, le dernier est 9 c.-à-d. (10 - 1).

Affectation  
Utilisation

```
v[4] := 11.3 ;  
writeln(v[3]);
```

Libération  
(destruction)

```
Finalize(v);
```

Il faut libérer la mémoire allouée pour le tableau.

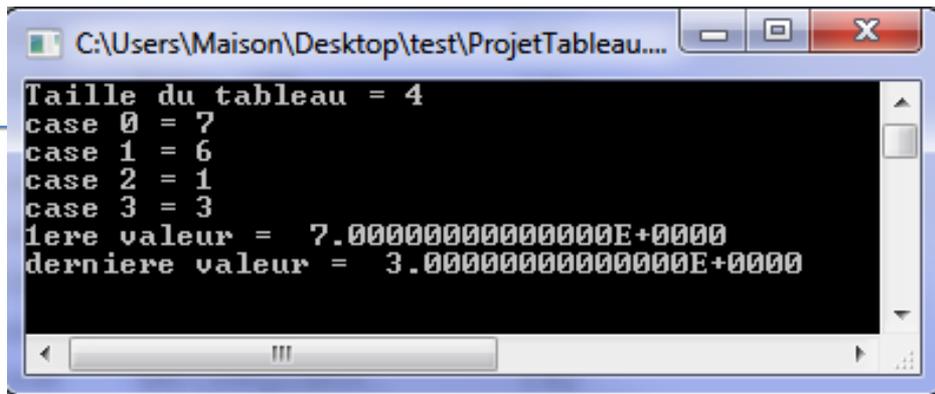
# Un exemple

```
program ProjetTableau;
{$APPTYPE CONSOLE}

uses sysutils;

//définition du type tableau
TYPE TTableau = array of double;

//variables
var n,i: integer;
    t: TTableau;
//prog. principal
begin
  write('Taille du tableau = '); readln(n);
  setLength(t,n); //allouer le tableau avec n cases
  //saisie des données
  for i:= 0 to n-1 do
    begin
      write('case ',i,' = ');
      readln(t[i]);
    end;
  //affichage de qqs valeurs du tableau
  write('1ere valeur = '); writeln(t[0]);
  write('derniere valeur = '); writeln(t[length(t)-1]);
  finalize(t); //désallouer le tableau
  readln;
end.
```



```
Taille du tableau = 4
case 0 = 7
case 1 = 6
case 2 = 1
case 3 = 3
1ere valeur = 7.000000000000000E+0000
derniere valeur = 3.000000000000000E+0000
```

`Length()` renvoie la taille du tableau. Il n'est donc pas nécessaire de trimbalier « n » partout.

```

program ProjetTableau;
{$APPTYPE CONSOLE}

uses sysutils;

//définition du type tableau
TYPE TTableau = array of double;

//variables
var n,i: integer;
    t: TTableau;
//prog. principal
begin
  write('Taille du tableau = '); readln(n);
  setLength(t,n); //allouer le tableau avec n
  //saisie des données
  for i:= 0 to n-1 do
    begin
      write('case ',i, ' = ');
      readln(t[i]);
    end;
  //affichage de qqz valeurs du tableau
  write('1ere valeur = '); writeln(t[0]);
  write('derniere valeur = '); writeln(t[length(t)-1]);
  //rajouter une case et y mettre une valeur
  setlength(t,length(t)+1);
  t[length(t)-1]:= 1664;
  //affichage
  write('last valeur = '); writeln(t[length(t)-1]);
  finalize(t); //désallouer le tableau
  readln;
end.

```

**Tableaux dynamiques : la taille est modifiable durant l'exécution !**

```

C:\Users\Maison\Desktop\test\ProjetTablea...
Taille du tableau = 3
case 0 = 0
case 1 = 1
case 2 = 2
1ere valeur = 0.000000000000000E+0000
derniere valeur = 2.000000000000000E+0000
last valeur = 1.664000000000000E+0003

```

- (1) Si on agrandit le tableau, les nouvelles cases prennent la valeur 0.
- (2) Si on le réduit, les valeurs dans les cases restantes sont préservées.

# Tableaux dynamiques à 2 dimensions

Il n'y a plus de restriction *a priori* sur la taille du tableau.



Définition du type `TYPE TMat = array of array of double;`

Création de l'instance

```
var m : TMat;
```

Initialisation

(un tableau dynamique est en fait un pointeur)

```
SetLength(m, 10, 5);
```

Création d'une matrice de 10 lignes (indices : 0 à 9) et de 5 colonnes (indices : 0 à 4).

Affectation  
Utilisation

```
m[4, 2] := 12.5 ;  
writeln(m[3, 4]);...
```

Hélas, `length()` ne permet pas d'obtenir les tailles sur les 2 dimensions.

Libération  
(destruction)

```
Finalize(m);
```

# FIN...

Les mêmes concepts sont – à peu de choses près – présents dans tous les langages de programmation...