UNIVERSITÉ PRIVÉE DE MARRAKECH | UP | جامعة ابن طفيل مراكش
UNIVERSITÉ RECONNUE PAR L'ÉTAT
جامعة معترف بها من طرف الدولة

Programmation Impérative
Les tableaux

INFO 2

Année universitaire 2022-2023

Contenu

I. Introduction
II. Tableaux fixes
    1. Utilisation des tableaux
    2. Déclaration d'un tableau
    3. Utilisation des éléments d'un tableau
    4. Tableaux multidimensionnels
III. Tableaux dynamiques
    1. Étapes de création d'un tableau dynamique

Introduction

Introduction: Insuffisance de la notion de variable

- Supposons que dans un programme, nous avons besoin de conserver les notes de 30 étudiants.
  - Solution : On peut utiliser une seule variable qui prend successivement la valeur des différentes notes.
- Si nous avons besoin de disposer des 30 notes simultanément alors :
  - Solution : On utilise 30 variables différentes.

Introduction: Inconvénients des variables

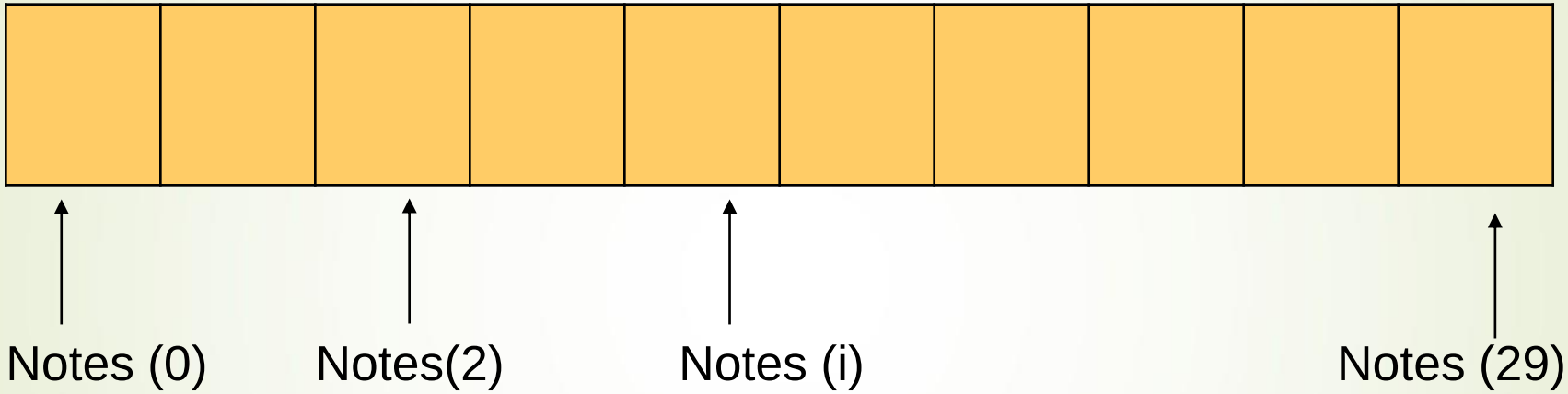
- Il faut trouver un nom de variable par note. Que ce passe t-il quand on veut avoir cent, mille notes, …
- De plus, il n'existe aucun lien entre ces différentes valeurs (notes). Parce que dans certains cas, on est amené à appliquer un même traitement (somme, moyenne, variance,…) à l'ensemble ou à une partie de ces valeurs.

Introduction: La solution
Le tableau

- Les tableaux sont certainement les variables structurées les plus populaires.
- Les tableaux sont disponibles dans tous les langages de programmation et servent à résoudre une multitude de problèmes.
- La notion de tableau consiste :
  - À attribuer un seul nom à l'ensemble des 30 valeurs, par exemple NOTES
  - À repérer chaque note par ce nom suivi entre parenthèses d'un numéro compris entre 0 et 29

Les tableaux fixes

Tableaux fixes: Description d'un tableau



Notes(0) : la première note du tableau
Notes(2) : la troisième note du tableau
…..
Notes(i) : La ième note du tableau

i est l'indice du tableau

- Un tableau à une dimension est une variable structurée formée d'un nombre entier N de variables simples de même type, qui sont appelées les composantes du tableau.
- Le nombre de composantes N est alors la dimension (taille) du tableau.

Tableaux fixes

- Le tableau permet de stocker de grandes quantités d'informations pendant l'exécution d'un programme.
- Il permet d'invoquer un groupe de valeurs en utilisant un seul nom et en traitant ces valeurs séparément ou collectivement.

Tableaux fixe: Déclaration d'un tableau

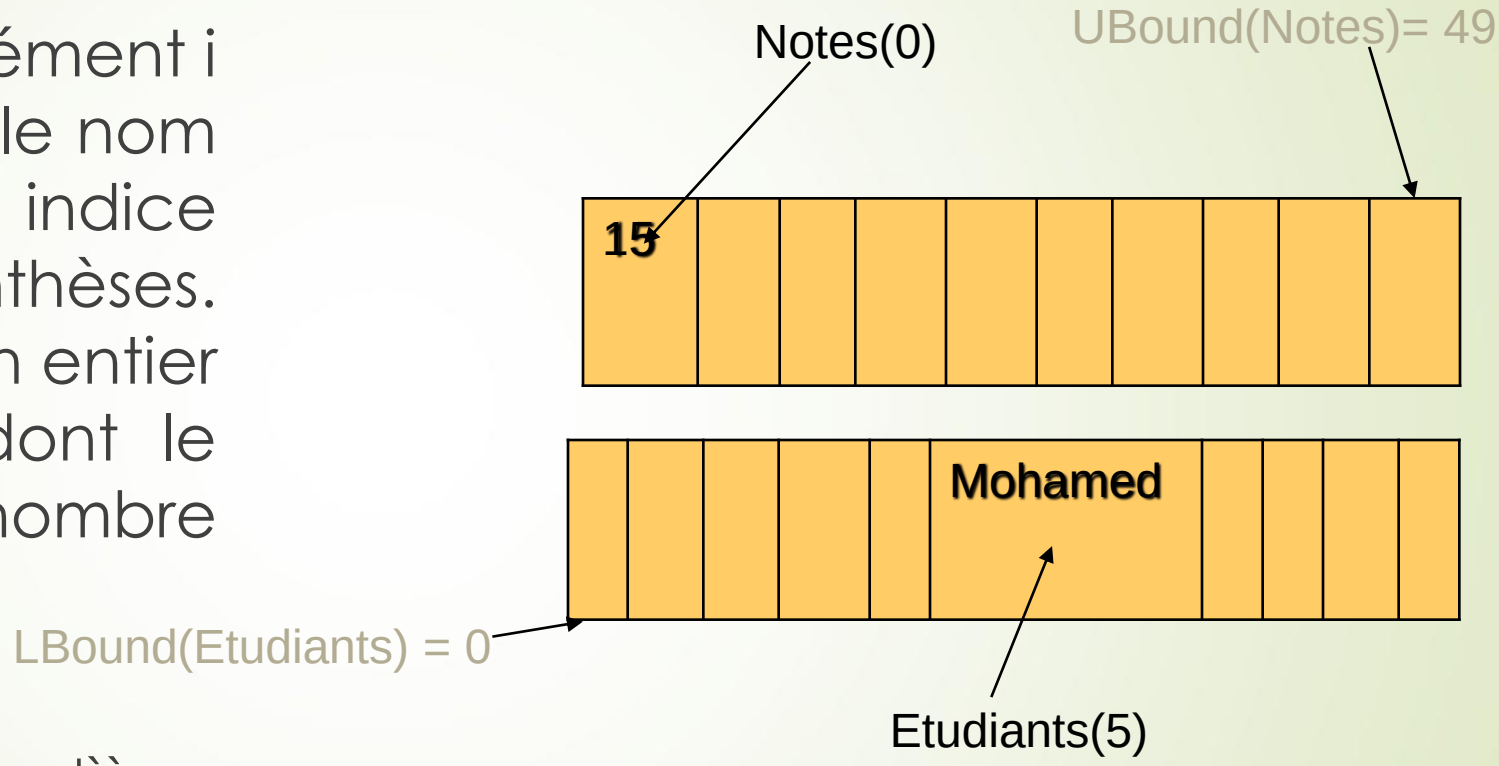
- La syntaxe de base d'un tableau est :
  - Dim NomTableau(IndiceDimension) As Type
  - NomTableau : est le nom de variable (identificateur) du Tableau qui sera utilisé dans le programme
  - IndiceDimension : est la limite supérieure de la dimension du tableau, qui correspond au nombre total d'éléments de cette dimension -1
  - Type : est le type des données qui seront stockées dans le tableau

Exemples de déclarations

- Dim Etudiants(9) As String
  - Tableau (vecteur) de chaînes de caractères nommé **Etudiants** qui peut contenir dix noms d'étudiants (numérotés de 0 à 9).
- Dim Notes(49) As Byte
  - Tableau d'entiers positifs nommé **Notes** qui peut contenir 50 notes d'étudiants (numérotés de 0 à 49).
- Par défaut, le premier élément d'un tableau est identifié par l'indice 0.
- Quand on déclare un tableau, VB.NET lui réserve l'espace mémoire nécessaire.

Utilisation des éléments d'un tableau

- Pour accéder à un élément i du tableau, on utilise le nom du tableau suivi d'un indice placé entre parenthèses. Cet indice doit être un entier ou une expression dont le résultat est un nombre entier.
  - Notes(0) = 15
  - Etudiants(5) = ``Mohamed``



Affichage et affectation

- La structure **for** se prête particulièrement bien au travail avec les tableaux.
- La majorité des applications se laissent implémenter par simple modification des exemples-types de l'affichage et de l'affectation.

Exemple 1 : Saisie des données et stockage dans un tableau

```
Dim Prompt, Title As String
Dim Temperatures(6) As Single
Dim i As Short
Message = " Entrer la température maximale du jour."
For i = 0 To UBound(Temperatures)
    Titre = "Jour   " & (i + 1)
    Temperatures(i) = InputBox(Message, Titre)
Next
```

Consultation des éléments d'un tableau

```
Dim Result As String
Dim i As Short
Dim Total As Single = 0

Result = "La température Max de la semaine :" & vbCrLf & vbCrLf
// Consultation du tableau
For i = 0 To 6
        Result = Result & " Jour " & (i + 1) & vbTab & Temperatures(i) & vbCrLf
        Total = Total + Temperatures(i)
 Next

Result = Result & vbCrLf & _
" Moyenne température:  " & _ format(Total / 7, "0.0")

TextBox1.Text = Result
```

La température Max de la semaine :

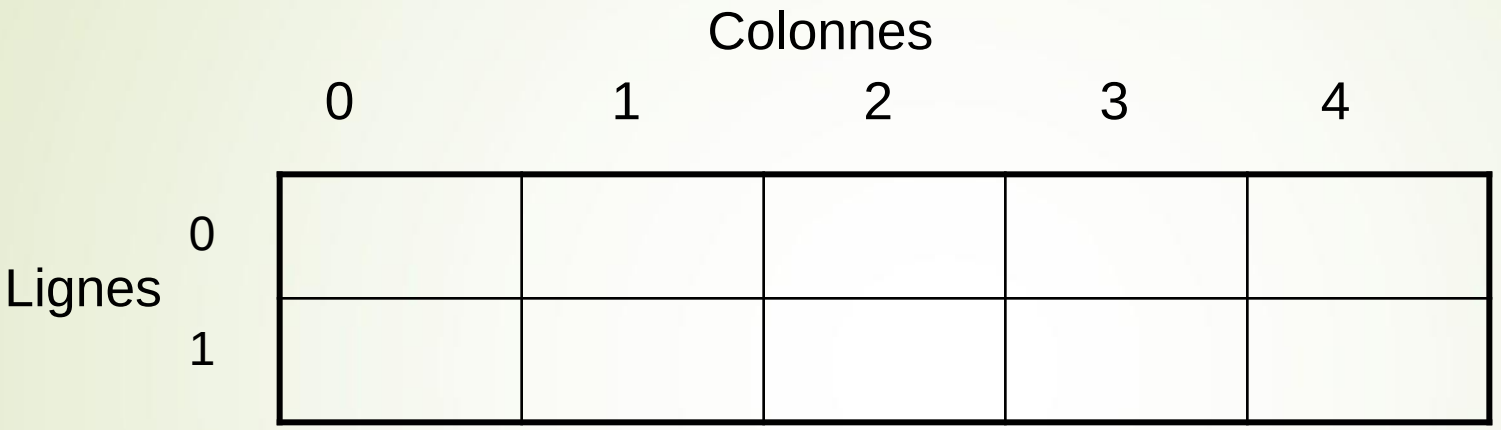
Jour 1 20
Jour 2 21
Jour 3 22
Jour 4 23
Jour 5 24
Jour 6 25
Jour 7 26

Moyenne température: 23,0

Tableaux multidimensionnels

- La syntaxe de base d'un tableau de taille fixe est :
- Dim NomTableau(IndiceDimension1, IndiceDimension2, …) As Type
- Les arguments sont importants :
  - IndiceDimension1 : limite supérieure de la première dimension du tableau, qui correspond au nombre total d'éléments de cette dimension -1
  - IndiceDimension2 : limite supérieure de la deuxième dimension du tableau, qui correspond au nombre total d'éléments de cette dimension -1
- Exemple
  - Dim Matrice (1,4) As Short
  - Déclaration d'un tableau bidimensionnel nommé **Matrice** qui permet de stocker 2 lignes et 5 colonnes de données entières de type Short.

Tableaux à 2 dimensions



- Si L est le nombre de lignes du tableau et C le nombre de colonnes du tableau. L et C sont alors les deux dimensions du tableau.
- Un tableau a deux dimensions; il contient donc L*C composantes.
- On dit qu'un tableau à 2 dimensions est carré, si L est égal à C.
- En faisant le rapprochement avec les mathématiques, on peut dire que le tableau à deux dimensions est une matrice.

Exemple

- Soit un tableau TEMPS à une dimension pour mémoriser les 7 températures maximales de la semaine : Dim TEMPS(6) As Short
- Pour mémoriser les températures des 4 semaines d'un mois, nous pouvons rassembler plusieurs de ces tableaux à une dimension dans un tableau TEMPM à deux dimensions :
- Dim TEMPM(3,6) As Short
- Dans une ligne nous retrouvons les températures de la semaine. Dans une colonne, nous retrouvons toutes les températures d'une journée.
- Si la température de la 4ème journée de la 1ère semaine est 15 alors : TEMPM(0,3) = 15

Colonnes : Températures max de chaque jour

	0	1	2	3	4	5	6
0	20	23	19	15	17	20	23
1	19	18	16	19	20	20	21
2	22	23	23	22	20	19	18
3	20	21	22	23	24	25	25

Accès aux composants

Colonnes j

Lignes i

A(0,0)	A(0,1)	A(0,2)
A(1,0)	A(1,1)	A(1,2)

- Dim A(1,2) As Short
- L'élément (i,j) du tableau A est :
  - A (<Ligne i>,<Colonne j>)

Exemple : Remplissage d'une matrice

```
Dim Matrice(1, 1) As Short
Dim I, J, Valeur As Short

    For I = 0 To 1
        For J = 0 To 1
            Valeur = InputBox("Entrer un nombre positif compris entre 0 et 255")
            Matrice(I, J) = Valeur
        Next
    Next
```

Exemple : Affichage du contenu d'une matrice

```
Dim Matrice(1, 1) As Short
Dim I, J, Valeur As Short
For I = 0 To 1
    For J = 0 To 1
       MsgBox("L‘element de la ligne" & CStr(I) & " et la  colonne " & CStr(J) & " est : " & CStr(Matrice(I, J)))
    Next
 Next
```

Les tableaux dynamiques

Tableaux dynamiques: Création d'un tableau dynamique

- Jusqu'à présent, nous avons vu des tableaux dont la taille a été fixée avant l'exécution (automatique).
- Comment faire si l'utilisateur détermine lui-même la taille de son tableau lors de l'exécution
  - Le nombre de températures maximal entré par l'utilisateur lors de l'exécution
- Visual Basic gère ce problème avec un conteneur élastique spécial : Tableau dynamique.

Les étapes de création d'un tableau dynamique

1. Spécifier le nom et le type du tableau pendant la conception sans donner le nombre d'éléments
   - Dim Temperatures() As Single
2. Ajouter du code pour déterminer le nombre d'éléments qui devront figurer dans le tableau lors de l'exécution.
   - Dim Jours As Short
   - Jours = InputBox (``Entrer le nombre de jours``)
3. Utiliser cette variable dans une instruction ReDim pour dimensionner (définir la taille) le tableau
   - ReDim Temperatures(Jours – 1)
4. Utiliser la fonction UBound pour déterminer la limite supérieure dans une boucle FOR…Next

```
For I = 0 To UBound(Temperatures)
        Temperatures(i) = InputBox(``Entrer la température``)
Next
```

Exemple

```
Dim Temperatures() As Single
Dim Jours As Integer
Dim Message, Titre, Result As String
Dim i As Short
Dim Total As Single = 0
Prompt = "Enter la temperature Max du jour."
Jours = InputBox("Combien de jours?", "création tableau")
If Jours > 0 Then ReDim Temperatures(Jours - 1)
For i = 0 To UBound(Temperatures)
    Titre = "Jour " & CStr(i + 1)
    Temperatures(i) = InputBox(Message, Titre)
Next
```

Redimensionnement d'une matrice

```
Dim Matrice(,) As Short
Dim i, J, Dim1, Dim2, Valeur As Short

Dim1 = InputBox("Entrer le nombre de lignes")
Dim2 = InputBox("Entrer le nombre de colonnes")

ReDim Matrice(Dim1 - 1, Dim2 - 1)
For i = 0 To UBound(Matrice, 1)
    For J = 0 To UBound(Matrice, 2)
        Valeur = InputBox("Entrer un nombre positif compris entre 0 et 255")
        Matrice(i, J) = Valeur
    Next
Next
```

Préservation du contenu d'un tableau avec ReDim Preserve

- Si on redimensionne un tableau qui contient déjà des données, ces informations seront irrémédiablement perdues.
- Après l'exécution de l'instruction ReDim, le contenu d'un tableau dynamique est défini à sa valeur par défaut.
- Le mot-clé Preserve, sauvegarde les données d'un tableau pendant son redimensionnement.
- ReDim Preserve NomTableau(dim1, dim2,…)
- Seule la dernière dimension d'un tableau peut être redimensionnée.

Exemple avec un tableau à une dimension

```
Dim Philosophes() As String
ReDim Philosophes(200)
Philosophes(200) = "Steve Harrison"
ReDim Preserve Philosophes(300)
```

Exemple avec un tableau tridimensionnel

```
Dim monCube(,,) As Single
//on re-dimensionne le tableau
ReDim monCube(25,25,25)
monCube(10,1,1) = 150
//on augmente la taille de la 3ème //dimension en préservant son contenu
ReDim Preserve monCube(25,25,50)
```

FIN