

Les tableaux

Nicolas Delestre et Michel Mainguenaud

`{Nicolas.Delestre,Michel.Mainguenaud}@insa-rouen.fr` Adapté

pour l'ENSICAEN par

Luc Brun

`luc.brun@ensicaen.fr`



Plan...

- Pourquoi les tableaux ?
- Les tableaux à une dimension
- Les tableaux à deux dimensions
- Les tableaux à n dimensions

Pourquoi les tableaux ?

- Imaginons que l'on veuille calculer la moyenne des notes d'une promotion, quel algorithme allons nous utiliser ?
- Pour l'instant on pourrait avoir l'algorithme suivant :

Nom: moyenne

Role: Affichage de la moyenne des notes d'une promo saisies par le prof

Entrée: -

Sortie: -

Déclaration: somme, nbEleves, uneNote, i : Naturel

début

 somme \leftarrow 0.0

écrire(Nombre d'élèves:)

lire(nbEleves)

pour i \leftarrow 0 à nbEleves **faire**

écrire("Note de l'élève numéro",i,":")

lire(note)

 somme \leftarrow somme+note

finpour

fin

écrire("Moyenne des notes :",somme/nbEleves)

Pourquoi les tableaux ?

- Imaginons que l'on veuille toujours calculer la moyenne des notes d'une promotion mais en gardant en mémoire toutes les notes des étudiants (pour par exemple faire d'autres calculs tels que l'écart type, la note minimale, la note maximale, etc.)
- Il faudrait alors déclarer autant de variables qu'il y a d'étudiants, par exemple en supposant qu'il y ait 3 étudiants, on aurait l'algorithme suivant :

procédure moyenne ()

Déclaration somme, note1, note2, note3 : Naturel

début

écrire("Les notes des trois étudiants :")

lire(note1) ;

lire(note2) ;

lire(note3) ;

 somme \leftarrow note1+note2+note3

écrire("La moyenne est de :",somme/3)

Pourquoi les tableaux ?

- Le problème est que cet algorithme ne fonctionne que pour 3 étudiants
 - Si on en a 10, il faut déclarer 10 variables
 - Si on en a n , il faut déclarer n variables
...ce n 'est pas réaliste
- Il faudrait pouvoir par l'intermédiaire d'une seule variable
...c'est le rôle des tableaux

Les tableaux à une dimension...

- C'est ce que l'on nomme un type complexe (en opposition aux types simples vus précédemment)
- Le type défini par un tableau est fonction :
 - `taille` maximal que peut contenir le tableau
 - `type` des éléments que peut contenir le tableau
- Par exemple un tableau d'entiers de taille 10 et un tableau d'entiers de taille 20 sont deux types différents
- On peut utiliser directement des variables de type tableau, ou définir de nouveau type à partir du type tableau
- On utilise un type tableau via la syntaxe suivante :
 - **Tableau***[intervalle]* *de type des éléments stockés par le tableau*
où `intervalle` est un intervalle sur un type simple **dénombrable** avec des bornes **constantes**

Les tableaux à une dimension...

- Par exemple :



- **Type Notes = Tableau[1..26] de Naturel**

- défini un nouveau type appelé Notes, qui est un tableau de 26 naturels

- **a : Notes**

- déclare une variable de type Notes

- **b : Tableau[1..26] de Naturel**

- déclare une variable de type tableau de 26 Naturels

- a et b sont de même type

- **c : Tableau['a'..'z'] d'Entier**

- déclare une variable de type tableau de 26 entiers

- a et c sont de types différents

Les tableaux à une dimension...

- Ainsi l'extrait suivant d'algorithme :

tab : **Tableau**['a'..'c'] **de** Réel

tab['a'] ← 2.5

tab['b'] ← -3.0

tab['c'] ← 4.2

- ... peut être présentée graphiquement par :

Les tableaux à une dimension...

- On accède (en lecture ou en écriture) à la $i^{\text{ème}}$ valeur d'un tableau en utilisant la syntaxe suivante :
 - *nom de la variable*[*indice*]
- Par exemple si *tab* est un tableau de 10 entiers (tab : **Tableau**[1..10] d'Entier)
 - `tab[2] ← -5`
 - met la valeur -5 dans la 2^{ème} case du tableau
 - En considérant le cas où *a* est une variable de type Entier, `a ← tab[2]`
 - met la valeur de la 2^{ème} case du tableau tab dans a, c'est-à-dire 5
 -
 - **lire**(tab[1])
 - met l'entier saisi par l'utilisateur dans la première case du tableau
 -

Exemple...

Nom: moyenne

Role: Affichage de la moyenne des notes d'une promo saisies par le prof

Entrée: -

Sortie: -

Déclaration: somme, nbEleves, i : Naturel, lesNotes : **Tableau**[1..100] **de** Naturel
début

 somme \leftarrow 0

répéter

écrire("Nombre d'eleves (maximum 100) :")

lire(nbEleves)

jusqu'à ce que nbEleves > 0 et nbEleves \leq 100

pour i \leftarrow 1 à nbEleves **faire**

écrire("Note de l'eleve numero ",i," :")

lire(lesNotes[i])

finpour

pour i \leftarrow 1 à nbEleves **faire**

 somme \leftarrow somme + lesNotes[i]

finpour

écrire("La moyenne est de :",somme/nbEleves)

fin

Remarques...

- Un tableau possède un nombre maximal d'éléments défini lors de l'écriture de l'algorithme (les bornes sont des constantes explicites, par exemple 10, ou implicites, par exemple MAX)
 - ce nombre d'éléments ne peut être fonction d'une variable
- Par défaut si aucune initialisation n'a été effectuée les cases d'un tableau possèdent
- Le nombre d'éléments maximal d'un tableau est différent du nombre d'éléments significatifs dans un tableau
 - Dans l'exemple précédent le nombre maximal d'éléments est de 100 mais le nombre significatif d'éléments est référencé par la variable nbEleves
- L'accès aux éléments d'un tableau est direct (temps d'accès constant)
- Il n'y a pas conservation de l'information d'une exécution du programme à une autre

Les tableaux à deux dimensions...

- On peut aussi avoir des tableaux à deux dimensions (permettant ainsi de représenter par exemple des matrices à deux dimensions)
- On déclare une matrice à deux dimensions de la façon suivante :
 - **Tableau***[intervallePremièreDimension][intervalleDeuxièmeDimension]* **de** *type des éléments*
- On accède (en lecture ou en écriture) à la $i^{\text{ème}}$, $j^{\text{ème}}$ valeur d'un tableau en utilisant la syntaxe suivante :
 -

Les tableaux à deux dimensions...

- Par exemple si *tab* est défini par **tab : Tableau[1..3][1..2] de Réel**)
 - $\text{tab}[2][1] \leftarrow -1.2$
 - met la valeur -1.2 dans la case 2,1 du tableau
 - En considérant le cas où *a* est une variable de type Réel, $a \leftarrow \text{tab}[2][1]$
 - met -1.2 dans *a*

	1	2
1	7.2	5.4
2	-1.2	2
3	4	-8.5

Les tableaux à deux dimensions...

- Attention, le sens que vous donnez à chaque dimension est important et il ne faut pas en changer lors de l'utilisation du tableau

- Par exemple, le tableau `tab` défini de la façon suivante :

`tab : Tableau[1..3][1..2] de Réel`

`tab[1][1]←2.0;tab[2][1]←-1.2;tab[3][1]←3.4`

`tab[1][2]←2.6; tab[2][2]←-2.9; tab[3][2]←0.5`

... peut permettre de représenter l'une des deux matrices suivantes :

$$\begin{pmatrix} 2.0 & & 3.4 \\ 2.6 & -2.9 & 0.5 \end{pmatrix} \quad \begin{pmatrix} 2.0 & 2.6 \\ & -2.9 \\ 3.4 & 0.5 \end{pmatrix}$$

Les tableaux à n dimensions...

- Par extension, on peut aussi utiliser des tableaux à plus grande dimension
- Leur déclaration est à l'image des tableaux à deux dimensions, c'est-à-dire :
 - **tableau** [*intervalle1*][*intervalle2*]. . . [*intervallen*] **de type des valeurs**
 - Par exemple :
 - **tab : tableau**[1..10][0..9]['a'..'e'] **d'Entier**
- Ainsi que leur utilisation :
 - **tab**[2][1]['b'] ← 10
 - **a** ← **tab**[2][1]['b']