

Chapitre 6

Les types scalaires – Les intervalles

Chapitre 6 Les types scalaires – Les intervalles

I Définition du type scalaire

- C'est un ensemble de valeurs ordonnées et discrètes disposant des fonctions :



- `Pred()` : donne la valeur précédente du paramètre
 - `Succ ()` : donne la valeur suivante du paramètre
- Lorsque l'analyse d'un problème fait apparaître une information définie par un ensemble de valeurs connues, et pour mieux contrôler l'utilisation de cette information, on crée un nouveau type qui permet de la définir entièrement.

Chapitre 6 Les types scalaires – Les intervalles

I.1) Exemple

- ❑ On veut manipuler une variable qui permet de décrire les jours de la semaine (uniquement).
- ❑ Prendre un type chaîne pour cette variable autorise à y admettre toutes les valeurs de type chaîne.
- ❑ Pour ne gérer dans cette variable que les noms des jours, on définit un nouveau type qui va décrire l'ensemble des valeurs que peut prendre cette variable :

T_Jour(**type**) = (lundi, mardi, mercredi, jeudi, vendredi, samedi, dimanche)

- ❑ On énumère toutes les valeurs que peut prendre une variable de ce type. Les valeurs énumérées doivent être considérées comme des symboles et non comme des chaînes de caractères. Elles sont discrètes et ordonnées :

lundi < mardi < mercredi < jeudi < vendredi < samedi < dimanche

C'est la **position** dans l'énumération qui est utilisée pour la comparaison.

Chapitre 6 Les types scalaires – Les intervalles

Utilisation :

algo ...	Ord	Lexique
<u>si</u> ouvrable <u>alors</u> ecrire ("jour ouvrable") <u>sinon</u> ecrire ("jour chômé") <u>fsi</u> ouvrable ← (jour ≠ samedi) <u>et</u> (jour ≠ dimanche) Lirejour(jour)		T_Jour (type) = (lundi, ..., dimanche) jour (T_Jour) ouvrable(booléen) : à vrai si jour de semaine, faux sinon Lirejour(module) : permet la saisie d'une valeur de type T_Jour

I.2) Définition

- Un type scalaire est un type permettant de représenter une information définie par un ensemble de valeur ordonnées et discrètes. Il peut être prédéfini ou créé.
- Il peut être utilisé pour préciser l'intervalle de définition d'une table.

Chapitre 6 Les types scalaires – Les intervalles

3) Notion d'intervalle

On peut exercer un plus grand contrôle sur l'utilisation d'une valeur de type scalaire en définissant des sous-ensembles bornés de valeurs : c'est la notion d'**intervalle**.

1.3.1) Exemples

a) 1..Nbemax : ensemble de valeurs entières prises dans le type Entier, borné par 1 et Nbemax.

On peut donner un nom à cet ensemble \Rightarrow on crée le nom d'un nouveau type scalaire
 \Rightarrow T_NbeMax

Ce qui donne : T_NbeMax(**type**) = 1..Nbemax

Utilisation de ce type : Nbe (T_Nbemax) : nbre effectif d'étudiants

NB : Nbemax est une valeur constante connue : Nbemax (**cste**/entier = 1000)

Chapitre 6 Les types scalaires – Les intervalles

b) (lundi, mardi, ... , samedi, dimanche) : ensemble de valeurs possibles pour les jours de la semaine.

On peut donner un nom à cet ensemble et définir un nouveau type \Rightarrow

T_Jour (**type**) = (lundi, mardi, mercredi, jeudi, vendredi, samedi, dimanche) Et créer à partir de ce type un *sous-ensemble plus petit* \Rightarrow

T_We (**type**) = samedi..dimanche

c) Soit l'ensemble des caractères : 'a', 'b', ... , 'A', 'B', ... , 'Z'.

On définit à partir de cet ensemble un sous-ensemble plus petit, borné \Rightarrow

T_Maj (**type**) = 'A'..'Z'

qui permet de manipuler des variables dont le contenu ne peut être qu'une lettre majuscule de l'alphabet.

Chapitre 6 Les types scalaires – Les intervalles

I.3.2) Définition

- Un intervalle est un sous-ensemble d'un type scalaire et représente un sous-type de ce type scalaire.
- Toutes les opérations définies sur le type de base restent valides sur le sous-type \Leftrightarrow le sous-type " hérite " des propriétés du type de base.

Chapitre 6 Les types scalaires – Les intervalles

II Utilisation de types scalaires

Soit les tables suivantes :

Tabnote :

	NoteM	NoteF	NoteA
1			
2			
3			
...			
...			
Nbemax			

Tabcoef :

NoteM	NoteF	NoteA

Tabmoy :

1	2	3	4	5	...	Nbemax-1	Nbemax

Chapitre 6 Les types scalaires – Les intervalles

Pour une meilleure exploitation de ces tables, nous pouvons définir les types/constantes suivants :

- Nbemax (**cste**/entier=1000) *(nombre max d'étudiants traités)*
- T_Etud (**type**) = 1..Nbemax *(type intervalle)*
- T_Mat (**type**) = (NoteM, NoteF, NoteA) *(type énuméré)*
- T_Tabnote (**type**) = **table**[T_Etud, T_Mat] réel *(type table à deux dimensions)*
- T_Tabcoef (**type**) = **table**[T_Mat] réel *(type table à une dimension)*
- T_Tabmoy (**type**) = **table**[T_Etud] réel *(idem)*

L.ZERTAL

9

Chapitre 6 Les types scalaires – Les intervalles

algo	Calcul	Ord	Lexique
	<pre> pour i ← 1 à Nbe faire Tabmoy[i] ← somnote / somcoefficient pour j ← NoteM à NoteA faire somnote ← somnote + Tabnote[i,j]*Tabcoef[j] fpour somnote ← 0 fpour lire (Nbe) pour j ← noteM à noteA faire somcoef ← somcoef + Tabcoef [j] fpour somcoef ← 0 pour i ← 1 à Nbe faire pour j ← NoteM à NoteF faire lire (Tabnote [i,j]) fpour pour j ← noteM à noteF faire lire (Tabcoef [j]) fpour </pre>	<p>6</p> <p>6.3</p> <p>6.2</p> <p>6.1</p> <p>1</p> <p>5</p> <p>4</p> <p>2</p> <p>3</p>	<p>Constante + types (décrits ci-dessus)</p> <p>+</p> <p>i (T_Etud)</p> <p>j (T_Tmat)</p> <p>Tabcoef (T_Tabcoef)</p> <p>Tabnote (T_Tabnote)</p> <p>Tabmoy (T_Tabmoy)</p> <p>somcoef, somnote (réel)</p> <p>Nbe (T_Etud) : nbre effectif d'étudiants</p>

L.ZERTAL

10