

Les types non standards

Les types non standards

Le type énuméré:

Une variable énumérée possède un nombre fixe de valeurs prédéfinies; on ne peut alors lui affecter aucune autre valeur que celles prévues dans l'énumération.

On définit alors un nouveau type de données correspondant à cette énumération et on lui attribut un nom aussi.

Il devient possible alors de déclarer des variables de ce type et seulement de ce type.

Les types non standards

Syntaxe:

Type <nom énumération> = (élt1, élt2, ...);

Variable <nom variable> : < nom énumération>;

Exemple:

Type couleur = (vert, noir, blanc);

Variable C1, C2 : couleur;

Les constantes énumérées sont : vert, noir et blancs.

Les types non standards

Remarque:

On ne peut pas lire ou écrire une variable énumérée, mais on peut manipuler la suite d'éléments de différentes manières:

Des affectations:

C1 := vert;

C1 := C2;

Des fonctions prédéfinis : Pred (précédent), SUCC (suivant). ORD (numéro d'ordre dans la déclaration : de 0 à n-1.

Exemple: écrire (ORD (noir)); ici, c'est le 1.

Les types non standards

Des comparaisons:

Exemple: vert < noir suivant l'ordre donné dans la déclaration.

Des boucles:

Pour C1 := vert à noir faire

Des sélections:

Choix C1 de

vert :

noir :

blanc :

Fin;

Les types non standards

```
Programme type_enum;  
Type jour = (samedi, dimanche, lundi, mardi, mercredi, jeudi,  
vendredi);  
Variable no : entier;  
        J : jour;  
  
Debut  
    écrire ('entrez un numéro no jour entre 0 et 6');  
    lire (no);  
    j := jour (no);  
    si (J = vendredi) alors  
        écrire (' c'' est un jour férié')  
    sinon  
        écrire (' c'' est un jour de semaine');  
  
Fin.
```

Les types non standards

Le type intervalle:

Il est possible de définir des variables de type intervalle. Elle peut être entière, réelle, etc. mais ne peut prendre que des valeurs appartenant à un intervalle donné.

Syntaxe:

On définit un intervalle en deux phases.

```
Type majuscule = ' A ' .. ' Z ' ;  
    note = 0..20;
```

```
Variable lettre : majuscule;  
    x : note;
```

Remarque: à la différence des énumérations, chaque variable définie correspond à un type classique (entier,...etc.).

Mais ne peut contenir des valeurs appartenant aux bornes fixés dans le type.

Les types non standards

Le type ensemble:

Un ensemble est une collection d'objets comportant moins de 256 valeurs de même type. C'est un ensemble au sens mathématique sur lequel on peut réaliser les opérations classiques de réunions, d'intersection, de complémentation, d'inclusion, d'égalité et d'appartenance.

La syntaxe:

Variable ensemble : ensemble de <type de base>

Le type de base est:

- Un type scalaire énuméré;
- Un intervalle d'entiers ou de caractères;
- Le type caractère;
- Le type booléen.

Les types non standards

Exemple:

Type chiffre = ensemble de 1..10;

Variable A, B : chiffre;

Exemple:

Type couleur = (blanc, vert, rouge, bleu, jaune);

dessin = ensemble de couleur;

Variable D : dessin;

Les types non standards

L'affectation:

L'affectation de données à une variable déclarée de type ensemble se fait en utilisant des crochets « $[]$ ».

L'écriture $[]$ indique l'ensemble vide.

Les opérations sur les ensembles sont:

Réunion (+): $A = [1,2,3]$ $B = [3,4,5]$ $A+B = [1,2,3,4,5]$

Différence (-): $A = [1,2,3]$ $B = [3,4,5]$ $A-B = [1,2]$

Intersection (*): $A = [1,2,3]$ $B = [3,4,5]$ $A*B = [3]$

Inclusion (\leq): $A = [1,2]$ $B = [1,2,3]$ $A \leq B$ ou $B \geq A$

Egalité (=): $A = [1,2]$ $B = [1,2]$ $A = B$

Appartenance (dans): $A = [1,2]$ 1 dans A

Les types non standards

Exemple:

Programme ensemble;

Type majus ensemble de 'A'..'Z';

 minus = ensemble 'a'..'z';

Variable maj : majus;

 min : minus;

 c : char;

Debut

 écrire ('tapez une lettre');

 lire (c);

si c in maj **alors**

 écrire (' c''est une majuscule')

sinon

 écrire ('c''est minuscule')

sinon

 écrire('ce n''est pas une lettre');

Fin.

Les types non standards

Les enregistrements:

La notion **d'enregistrement** permet de regrouper ensemble, des données de type différents correspondant à un même objet.

Un **enregistrement** est un nouveau type défini par un identificateur possédant, en son sein, un certain nombre de variables. Une fois ce type défini, on peut déclarer des variables de ce type. On accède alors à une information en précisant le nom de la variable de **type enregistrement** suivi d'une variable membre généralement séparés par un point.

Le type enregistrement est défini à l'aide du mot clé « **enregistrement** » derrière lequel on déclare la liste des variables contenues dans l'enregistrement selon la syntaxe suivante:

Les types non standards

Syntaxe:

Type < enregistrement > = enregistrement // record

variable1 : type1;

variable2 : type2;

.....

variableN : typeN;

Fin;

Variable E : enregistrement;

Les types non standards

Exemple:

Type personne = enregistrement

Nom : chaine de caractères [25];

prénom : chaine de caractère[25];

sit_famille : (marié, celibataire, divorcé);

telephone : chaine de caractère [15];

Fin;

Variable employe : personne;

Remarque:

Une fois définie, une variable de type enregistrement s'utilise comme une variable simple.

Les types non standards

Employe . nom := ' M''hand ';

Écrire (employe . telephone);

Remarque:

La variable « **employe** » définie à l'aide du type enregistrement appelée « **personne** » ne peut contenir qu'un seul nom, une seule situation familiale, un seul numéro de téléphone et un seul prénom.

Il ne s'agit pas d'un tableau d'employés, mais d'une seule variable.

Les types non standards

Tableaux d'enregistrements:

Une fois le type enregistrement défini comme précédemment, on déclare l'existence d'un tableau dont les éléments sont de ce type.

On déclare ensuite le tableau selon la syntaxe habituelle en précisant que les éléments sont du type de l'enregistrement.

Exemple:

```
Type personne = enregistrement
    nom : chaîne de caractère [20];
    prenom : chaîne de caractères [20];
    sit_famille : (marié, celibataire, divorcé);
    telephone : chaîne de caractère [15];
Fin;
Variable employe : tableau [1 .. 30 ] de personne;
```


Les types non standards

On accède à chaque case du tableau contenant un enregistrement en respectant la syntaxe suivante:

< nom tableau > [indice] . < nom de la variable membre >

Les types non standards

Exemple:

Programme tab_enreg;

Type personne = enregistrement

nom : chaîne [20];

prenom : chaîne [15];

Fin;

Variable employe : tableau [1..10] de personne;

i : entier;

nomper : chaîne;

Debut

// saisie du tableau employé

Pour i := 1 à 10 faire

debut

écrire ('donnez le nom et le prénom');

lire (employe[i].nom , employe [i] . Prenom)

fin

// recherche d'un enregistrement par nom de personnes

Ecrire ('Donnez le nom de la personne à rechercher');

Lire(nomper);

Pour i := 1 à 30 **faire**

debut

si nomper = employe[i].nom **alors**

debut

ecrire ('nom =', employe[i].nom);

ecrire ('prenom =', employe[i].prenom)

fin

Fin

Les types non standards

Exemple: RECHERCHE D'UN ENREGISTREMENT DANS UN TABLEAU;

Programme recherche_enreg ;

Var nomper : chaine;

begin

Ecrire ('Donnez le nom de la personne à rechercher');

Lire(nomper);

Pour i := 1 à 30 faire

 debut

 si nomper = employe[i].nom alors

 debut

 ecrire ('nom =', employe[i].nom);

 ecrire ('prenom =', employe[i].prenom)

 fin

 fin

Fin.

Les types non standards

Accès aux champs d'un enregistrement :

Alors que les éléments d'un tableau sont accessibles au travers de leur indice, les champs d'un enregistrement sont accessibles à travers leur nom, grâce à l'opérateur '.'

Les types non standards

Exemple :

Programme de saisie des données concernant les personnes pers1 et pers2, puis affichage de la différence d'âge entre ces deux personnes

Programme Exemple

Type personne = enregistrement

nom : chaîne;
prénom : chaîne;
âge : entier;

Fin;

Var pers1, pers2 : tpersonne

Début

```
    écrire ("Entrez le nom puis l'age de la personne 1« );  
    lire ( pers1.nom, pers1.age ); // il est impossible d'écrire Saisir pers1  
    écrire ( "Entrez le nom puis l'âge de la personne 2« );  
    lire ( pers2.nom, pers2.age);  
    écrire ( "La différence d'âge entre ", pers1.nom, " et ", pers2.nom, " est de “: ”);  
    Si pers1.age > pers2.age Alors  
        écrire (pers1.age – pers2.age, " ans ");  
    Sinon  
        écrire ( pers2.age – pers1.age, " ans “)
```

FinSi

Fin

Les types non standards

Passage d'un enregistrement en paramètre d'un sous-programme:

Il est possible de **passer tout un enregistrement en paramètre d'une fonction ou d'une procédure** (on est pas obligé de passer tous les champs un à un, ce qui permet de diminuer le nombre de paramètres à passer), exactement comme pour les tableaux.

Exemple :

Voilà une fonction qui renvoie la différence d'age entre deux personnes:

Les types non standards

Fonction différence (p1, p2 : tpersonne)

Début

Si pers1.age > pers2.age Alors

Retourne (pers1.age – pers2.age)

Sinon

Retourne (pers2.age – pers1.age)

Fin;

Question :

- Définir la structure d'enregistrement de cette fonction.
- Définir le programme entier et l'appel de fonction.

Les types non standards

D'éclarer des types qui permettent de stocker :

1. Un joueur de basket caractérisé par son nom, sa date de naissance, sa nationalité, et son sexe

```
type t_sexe = (masculin, feminin);
```

```
type t_joueur = enregistrement
```

```
    nom : chaine [20];
```

```
    nation : chaine [20];
```

```
    sexe : t_sexe;
```

```
END;
```


Les types non standards

2. Une association de joueurs de basket

```
type t_assoc = tableau[1..N] de t_joueur;
```

Remarque : Pour permettre à l'association de croître, c'est-à-dire avoir plus de N joueurs, il faudra allouer un tableau dynamique ou une liste...

3. Une équipe de basket avec son nom, ses joueurs, ainsi que les points gagnés et les basket marqués et encaissés dans la saison courante.

```
type t_equipe = RECORD
    nom : string;
    joueurs : array[1..N] of t_joueur;
    points : integer;
    basketmarque : integer;
    basketencaisse : integer;
END;
```

Remarque : Pour les joueurs, il vaudrait mieux allouer un tableau dynamique ou une liste...

4. Un tableau de 18 équipes de basket

```
type t_tableau = array[1..18] of t_equipe;
```

Les types non standards

Exercice:

Je constate que la somme des n premiers nombres impairs est égale à n^2 , c'est à dire que $1 + 3 + 5 + \dots + (2n - 1) = n^2$.

1. Ecrivez la fonction `sommeImpairs(n : integer) : integer;` qui calcule la somme des n nombres impairs.
2. Ecrivez une fonction qui calcule la puissance carré d'un nombre quelconque X . Telque : $x := x * x$.

Les types non standards

```
function sommImpairs(n : integer) : integer;  
var i, somme : integer;  
begin  
    somme := 0;  
    for i := 1 to n do  
        begin  
            somme := somme + (2*i)-1;  
        end;  
        result := somme;  
    end;
```