Enregistrements

Dr Khadim DRAME
kdrame@univ-zig.sn

Département d'Informatique
UFR des Sciences et Technologies
Université Assane Seck de Ziguinchor

13 novembre 2021

Plan

Introduction

- Les types présentés dans les chapitres précédents du cours sont **prédéfinis**
  - Types **simples** (entier, réel, caractère, booléen)
  - Types **structurés** (tableaux, chaînes de caractères)
- On peut également définir de **nouveaux types**
  - Type **énuméré** $\rightarrow$ citer explicitement les valeurs que peut prendre une variable.
    Exemple : Type couleur=(vert,jaune,rouge,noir,blanc) ;
  - Type **intervalle** $\rightarrow$ fixer les valeurs dans un intervalle.
    Exemple : Type age=1..150 ;
  - Type **enregistrement** $\rightarrow$ représenter des données de **différents types** dans un même objet.

Notion d'enregistrement

- Un **enregistrement** (appelé aussi **structure**) permet de regrouper des données de **différents types** dans un même objet.
- C'est un type **structuré** qui regroupe un nombre fini de variables (ou **champs**) de types éventuellement différents.
- Exemples
  - Représenter un étudiant avec ses prénoms et nom, son âge, son adresse, son niveau...
  - Représenter une promotion d'étudiants par un tableau d'étudiants.

Plan

Définition de type enregistrement

Syntaxe

Type <nom_enregistrement=**record**
    <champ_1> : <type_1> ;
    <champ_2> : <type_2> ;
    ...
    <champ_n> : <type_n> ;
end ;

- les champs sont des variables qui peuvent être de tout type ;
- chaque champ doit avoir un nom différent.

Définition de type enregistrement

- Exemple

```
1 Type personne=record
2   prenoms: string;
3   nom: string;
4   age: integer;
5   sexe: (masculin,feminin);
6 end;
```

Imbrication d'enregistrements

- Il est possible d'utiliser une variable de type enregistrement comme champ d'un autre enregistrement.
- Exemple

```
1  Type adresse=record
2    numero: integer;
3    rue: string;
4    ville: string;
5  end;
6  Type personne=record
7    prenoms: string;
8    nom: string;
9    age: integer;
10   sexe: (masculin,feminin);
11   adr: adresse; {type enregistrement}
12 end;
```

Tableau comme champs d'un enregistrement

- Il est possible d'utiliser un tableau comme champ d'un enregistrement.
- Exemple

```
1 Type etudiant=record
2   numero: integer;
3   nom: string;
4   age: integer;
5   sexe: (masculin,feminin);
6   notes : array[1..5] of real; {tableau de réels}
7 end;
```

Déclaration de variable de type enregistrement

- Syntaxe

  var <nom_variable> : <nom_enregistrement> ;

- Exemple

```
1  program testEnregistrement;
2  Type personne=record
3    prenoms: string;
4    nom: string;
5    age: integer;
6    sexe: (masculin,feminin);
7  end;
8  var p:personne;
9  begin
10 ...
11 end.
```

Plan

Opérations sur les enregistrements

- L'**affectation** est la seule opération possible sur une variable de type enregistrement sans passer par ses champs.

  p1 :=p2 ;

- Les autres opérations (lecture, écriture, comparaison, etc.) ne sont pas applicables directement sur un type enregistrement.

  read(p) ; write(p1) ;p1<p2 ; pas acceptés

- Pour ces opérations → utiliser les champs de l'enregistrement.

  readln(p.nom) ; write(p1.age)

Accès aux champs d'un enregistrement

- Syntaxe

  <nom_variable>.<nom_champ>

- Exemple

```
1  program testEnregistrement;
2  Type personne=record
3    prenoms: string;
4    nom: string;
5    age: integer;
6    sexe: (masculin,feminin);
7  end;
8  var p, p1, p2:personne;
9  begin
10   p.nom:='Diop';
11   writeln(p.age);
12 end.
```

Accès aux champs d'un sous-enregistrement

- Exemple

```
1  program testEnregistrement;
2  var p:personne;
3  begin
4    p.nom:='Diop';
5    p.age:=45;
6    writeln('L''âge de M. ', p.nom,' est ', p.age);
7    p.adr.numero:=45;{accès au champ numéro}
8    p.adr.ville:='Ziguinchor';
9    {accès au champ ville du type adresse}
10 end.
```

Accès aux champs d'un enregistrement

Structure with...do

- La structure **with** permet d'accéder directement aux champs d'un enregistrement.
- Elle permet d'éviter la répétition d'une variable enregistrement.
- Syntaxe

with <nom_var> do
  begin
    <nom_champ> ;
  end ;

- Exemple

```
1  program testWithDo;
2  var p:personne;
3  begin
4    with p do
5    begin
6      nom:='Bâ'; {p.nom}
7      age:=45; {p.age}
8      sexe:=masculin;
9    end;
10 end.
```

Définition et manipulation d'enregistrement

Exercices d'application

- Exercice 1

  Écrire un programme qui définit un type personne (prénom, nom, âge), demande à l'utilisateur de saisir des données concernant deux personnes, puis affiche de la différence d'âge entre ces deux personnes.

Définition et manipulation d'enregistrement

```
1  program diff_age;{Correction de l'exercice 1}
2  Type personne=record
3    prenoms, nom: string;
4    age: integer;
5  end;
6  var p1, p2:personne;
7      diff: integer;
8  begin
9    writeln('Donner le nom et l''âge de p1 ');
10   readln(p1.nom, p1.age);
11   writeln('Donner le nom et l''âge de p2 ');
12   readln(p2.nom, p2.age);
13   if p1.age>p2.age then
14     diff:=p1.age-p2.age
15   else
16     diff:=p2.age-p1.age;
17   writeln('La différence d''âge est ', diff);
18 end.
```

Tableaux d'enregistrements

- Il est possible de créer un tableau dont les éléments sont de type enregistrement.
- Exemple

```
1  program tabEnregistrement;
2  Type etudiant=record
3    prenoms: string;
4    nom: string;
5    sexe: (masculin,feminin);
6    moy: real;
7  end;
8  var tab_etud: array[1..100] of etudiant;
9  begin
10   tab_etud[1].nom:='Ndiaye';
11   tab_etud[1].age:=25;
12   tab_etud[1].sexe:=masculin;
13   tab_etud[1].moy:=14.5;
14   tab_etud[2].nom:='Diallo';
15 end.
```

Tableaux d'enregistrements

Exercices d'application

- Exercice 2

  Écrire un programme qui définit un type personne (prénom, nom, age), demande à l'utilisateur de saisir des données de cinq personnes, puis affiche les informations de ces personnes (utiliser un tableau d'enregistrements).

Tableaux d'enregistrements

Exercices d'application

- Correction de l'exercice 2

```
1  program saisie_enreg;
2  var
3    i : integer;
4    t : array[1..5] of personne;
5  begin
6    for i:=1 to 5 do {saisie des infos}
7      with t[i] do
8        begin
9          writeln('Donner les prénom, nom et âge ');
10         readln(prenom, nom, age);
11       end;
12   for i:=1 to 5 do {affichage des infos}
13     with t[i] do
14       writeln(prenom,' ', nom, ' ', age);
15 end.
```

Plan

Enregistrements et sous-programmes

- Un enregistrement, comme les autres types de données, peut être utilisée comme paramètre d'une fonction ou d'une procédure.
- Exemple

```
1 function difference(p1, p2: personne): integer;
2 begin
3   if p1.age>p2.age then
4     difference:=p1.age-p2.age
5   else
6     difference:=p2.age-p1.age;
7 end.
```

- Exercice 3

  Un médecin souhaite enregistrer sur un ordinateur les fiches de ses patients. Une fiche est constituée d'un nom (chaîne de 30 caractères maximum), d'un numéro (entier), d'un numéro de téléphone (10 caractères maximum) et du sexe du patient (homme ou femme).

  1. définir un type permettant de représenter les données des patients ;
  2. écrire une procédure qui permet de saisir les informations d'un patient ;
  3. écrire une procédure qui permet d'afficher les informations d'un patient ;
  4. écrire un programme qui fait appel à ces procédures pour saisir et afficher les infos d'un patient ;

Enregistrements et sous-programmes

Exercices d'application

```
1  {Programme principal}
2  program fiche_patient;
3  {Définition du type patient}
4  Type patient = Record
5    nom : string[30];
6    num : integer;
7    tel : string[10];
8    sexe : (homme, femme);
9  end;
10 var  pat : patient;
11 {Définition des procédures ici}
12 begin
13   saisir(pat);
14   afficher(pat);
15 end.
```

Enregistrements et sous-programmes

Exercices d'application

```
1  {Définition de la procédure saisir}
2  procedure saisir(var p: patient) ;
3  begin
4    with p do
5    begin
6      write('Entrer le nom du patient : ');
7      readln(nom);
8      write('Entrer son  numéro : ');
9      readln(num);
10     write('Entrer son numéro de téléphone : ');
11     readln(tel);
12     write('Entrer son sexe  : ');
13     readln(sexe);
14   end;
15 end;
```

Enregistrements et sous-programmes

Exercices d'application

```
1  {Définition de la procédure afficher}
2  procedure afficher(p: patient) ;
3  begin
4    with p do
5    begin
6      writeLn('Nom du patient : ', nom);
7      writeLn('Numéro du patient : ', num);
8      writeLn('Numéro de téléphone : ', tel);
9      writeLn('Sexe du patient : ', sexe);
10   end;
11 end;
```

Complément : le type ensemble

- Syntaxe

  Type <nom_ensemble> = **set** of <type> ;

  var <nom_var> : <nom_ensemble> ;

- Exemple

  Type chiffres = 1..9 ;

  ensemble_chiffres = set of chiffres ;

  var ens1, ens2 : ensemble_chiffres ;

  ens1 := [1,3,7] ; ens2 := [2,4,6] ;

- Opérations possibles
  - opérations classiques sur les ensembles : union (+), intersection (*), différence (-), appartenance (elt **in** ens1) ;
  - opérations de comparaison.

Synthèse

- Un **enregistrement** est un type structuré qui permet de regrouper un ensemble de données (dites **champs**).
- Les champs d'un enregistrement peuvent être de types **différents**.
- Les champs d'un enregistrement peuvent être de tout type.
- La structure **with** permet d'éviter la répétition d'un nom de variable de type enregistrement lors de l'accès à ses champs.
- L'utilisation de sous-programmes simplifie la manipulation des enregistrements.