

Les enregistrements

Christian Lasou, Nour-Eddine Oussous, Éric Wegrzynowski

Licence ST-A, USTL - API1

12 mars 2007

1 Introduction

- Exemples

2 Enregistrements en PASCAL

- Déclaration
- Expressions
- Accès aux champs
- Restrictions

Objectif

Objectif

- pouvoir manipuler des données ...

Objectif

- pouvoir manipuler des données ...
- ... composées d'autres données ...

Objectif

- pouvoir manipuler des données ...
- ... composées d'autres données ...
- ... de types hétérogènes.

Exemple 1/3

Carte à jouer = structure composée

Exemple 1/3

Carte à jouer = structure composée

1 de la couleur de la carte (♣, ♦, ♥, ♠);

Exemple 1/3

Carte à jouer = structure composée

- 1 de la couleur de la carte (♣, ♦, ♥, ♠);
- 2 de la valeur de la carte (as, 2, 3, ..., roi).

Exemple 1/3

Carte à jouer = structure composée

- 1 de la couleur de la carte (♣, ♦, ♥, ♠);
- 2 de la valeur de la carte (as, 2, 3, ..., roi).

⇒ une carte est un couple (couleur,valeur)

$$\text{Carte} = \text{Couleur} \times \text{Valeur}$$

Exemple 2/3

Date = structure composée

Exemple 2/3

Date = structure composée

1 du nom du jour (lundi, mardi, ...);

Exemple 2/3

Date = structure composée

- 1 du nom du jour (lundi, mardi, ...);
- 2 du quantième dans le mois (1, 2, ... 31);

Exemple 2/3

Date = structure composée

- 1 du nom du jour (lundi, mardi, ...);
- 2 du quantième dans le mois (1, 2, ... 31);
- 3 du mois (janvier, février, ...);

Exemple 2/3

Date = structure composée

- 1 du nom du jour (lundi, mardi, ...);
- 2 du quantième dans le mois (1, 2, ... 31);
- 3 du mois (janvier, février, ...);
- 4 de l'année.

Exemple 2/3

Date = structure composée

- 1 du nom du jour (lundi, mardi, ...);
- 2 du quantième dans le mois (1, 2, ... 31);
- 3 du mois (janvier, février, ...);
- 4 de l'année.

⇒ une date est un quadruplet (j,q,m,a) de l'ensemble

$$Date = Jour \times Quantieme \times Mois \times Annee$$

Exemple 3/3

Étudiant = structure composée

Exemple 3/3

Étudiant = structure composée

1 de son NIP

Exemple 3/3

Étudiant = structure composée

- 1 de son NIP
- 2 de son nom

Exemple 3/3

Étudiant = structure composée

- 1 de son NIP
- 2 de son nom
- 3 de son prénom

Exemple 3/3

Étudiant = structure composée

- 1 de son NIP
- 2 de son nom
- 3 de son prénom
- 4 de sa date de naissance

Exemple 3/3

Étudiant = structure composée

- 1 de son NIP
- 2 de son nom
- 3 de son prénom
- 4 de sa date de naissance
- 5 du diplôme qu'il prépare

Exemple 3/3

Étudiant = structure composée

- 1 de son NIP
- 2 de son nom
- 3 de son prénom
- 4 de sa date de naissance
- 5 du diplôme qu'il prépare

⇒ étudiant = quintuplet (nip,nom,prénom,date,diplôme)

$$Date = NIP \times Nom \times Prenom \times Date \times Diplome$$

Les enregistrements

Les enregistrements

- En PASCAL, représentation des n -uplets avec des enregistrements (ou articles, ou structures).

Les enregistrements

- En PASCAL, représentation des n -uplets avec des enregistrements (ou articles, ou structures).
- Les composantes d'un enregistrement sont nommées champs.

Le type **record**

Enregistrement = **record**.

Syntaxe de la définition d'un type **record** :

```
type  
  <nom_du_type> = record  
    <nom_du_champ 1> : <type_du_champ 1>;  
    <nom_du_champ 2> : <type_du_champ 2>;  
    ...  
end {record};
```

Le type **record**

Enregistrement = **record**.

Syntaxe de la définition d'un type **record** :

```
type
  <nom_du_type> = record
    <nom_du_champ 1> : <type_du_champ 1>;
    <nom_du_champ 2> : <type_du_champ 2>;
    ...
end {record};
```

NB : pas de **begin** correspondant au **end**

Exemple des dates

type

JOUR = (LUNDI, MARDI, MERCREDI, JEUDI,
 VENREDI, SAMEDI, DIMANCHE);

QUANTIEME = 1..31;

MOIS = (JANVIER, FEVRIER, MARS, AVRIL,
 MAI, JUIN, JUILLET, AOUT,
 SEPTEMBRE, OCTOBRE, NOVEMBRE,
 DECEMBRE);

ANNEE = CARDINAL;

DATE = record

 jour : JOUR;
 quantieme : QUANTIEME;
 mois : MOIS;
 annee : ANNEE;

end {DATE};

Déclaration de constante

Déclaration d'une constante typée de type enregistrement :

const

<nom> : <type> = <valeur>;

Déclaration de constante

Déclaration d'une constante typée de type enregistrement :

const

<nom> : <type> = <valeur>;

Example :

const

```
AUJOURDHUI : DATE = (jour      : MARDI;
                      quantieme : 13;
                      mois       : MARS;
                      annee      : 2007);
```

Expressions de type enregistrement

Une expression de type enregistrement peut être

Expressions de type enregistrement

Une expression de type enregistrement peut être

- toute constante ou variable de ce type

```
var
  dateCourante : DATE;
begin
  ...
  dateCourante := AUJOURDHUI;
  ...
end
```

Expressions de type enregistrement

Une expression de type enregistrement peut être

- toute constante ou variable de ce type

```
var
  dateCourante : DATE;
begin
  ...
  dateCourante := AUJOURDHUI;
  ...
end
```

- tout appel à une fonction retournant ce type

```
function lendemain(d : DATE) : DATE;
...
demain := lendemain(AUJOURDHUI);
...
```

Accès aux champs

Accès aux champs d'un enregistrement avec la notation pointée

Exemple :

```
{le jour de la date courante}
```

```
dateCourante.jour
```

```
{le mois de la date courante}
```

```
dateCourante.mois
```

```
{l'année de la date courante}
```

```
dateCourante.annee
```

Modification de la valeur d'un champ

Possibilité de ne modifier qu'un seul champ d'un enregistrement.

Exemple :

```
{dateCourante = (jour : LUNDI; quantieme : 13;  
                 mois : MARS; annee : 2007)}  
dateCourante.jour := MARDI;  
{dateCourante = (jour : MARDI; quantieme : 13;  
                 mois : MARS; annee : 2007)}
```

Lendemain d'une date

Fonction de calcul du lendemain d'une date

```
// lendemain(d) = date du lendemain de d
function lendemain(d : DATE) : DATE;
var
    demain : DATE;
begin
    demain.jour := jourSuivant(d.jour);
    if finAnnee(d) then begin
        demain.quantieme := 1;
        demain.mois := JANVIER;
        demain.annee := anneeSuivante(d.annee);
    end else if fin_mois(d) then begin
        demain.quantieme := 1;
        demain.mois := moisSuivant(d.mois);
        demain.annee := d.annee;
    end else begin
        demain.quantieme := quantiemeSuivant(d);
        demain.mois := d.mois;
        demain.annee := d.annee;
    end {if};
    lendemain := demain;
end {lendemain};
```

Restrictions

Restrictions

- pas d'opérateurs de comparaison prédéfini

Restrictions

- pas d'opérateurs de comparaison prédéfini
⇒ nécessité de définir un prédicat d'égalité pour chaque type enregistrement

Restrictions

- pas d'opérateurs de comparaison prédéfini
⇒ nécessité de définir un prédicat d'égalité pour chaque type enregistrement
- pas de procédures d'entrée/sortie prédéfinie (**read** et **write** ne conviennent pas aux données de type enregistrement)

Restrictions

- pas d'opérateurs de comparaison prédéfini
 - ⇒ nécessité de définir un prédicat d'égalité pour chaque type enregistrement
- pas de procédures d'entrée/sortie prédéfinie (**read** et **write** ne conviennent pas aux données de type enregistrement)
 - ⇒ nécessité de définir de telles procédures pour chaque type enregistrement

Égalité des dates

Prédicat d'égalité de deux dates

```
// datesEgales( $d_1, d_2$ )  $\Leftrightarrow d_1 = d_2$   
// deux dates sont égales ssi les champs sont égaux  
function datesEgales(d1,d2 : DATE) : BOOLEAN;  
begin  
    datesEgales := (d1.jour = d2.jour) and  
                   (d1.quantieme = d2.quantieme) and  
                   (d1.mois = d2.mois) and  
                   (d1.annee = d2.annee);  
end {datesEgales};
```