

32 - ARTICLES À PARTIES VARIANTES

Programmation Concurrente - LI330
Université P. & M. Curie - année scolaire 2013/2014

PrC



ARTICLE: RAPPEL

- **Objet constitué de composants hétérogènes**
 - De sous-types différents
- **La valeur d'un objet article est constituée de la valeur de ses composants**
- **Deux types d'articles**
 - Articles à format constant
 - Articles à format variable





ARTICLES À PARTIES VARIANTES

decl_art **type** type_id [(discriminant{discriminant})] **is record**
 {un_champ} partie_var {partie_var} {un_champ}
end record;

partie_var **case** identificateur **is**
 when choix => un_champ
 { **when** choix => un_champ }
end case;

choix littéral du type considéré (valeur immédiate)
 littéral1 .. littéral2 (intervalle discret)
 identificateur d'un sous-type
 valeurs | valeurs

discriminant liste_ident : type_id [:= expression];

un_champ liste_ident : type_id [:= expression];
null;

liste_ident identificateur
 identificateur liste_ident

type_id identificateur



UNIVERSITÉS



EXEMPLE (1)

-- Appelé type "polymorphe" en Ada

```
type T_Sexe is (Homme, Femme);
```

```
type Personne (Sexe : T_Sexe) is record
```

```
  Nom, Prenom : String (1 .. 20) := (others => ' ');
```

```
  case Sexe is
```

```
    when Homme =>
```

```
      null;
```

```
    when Femme =>
```

```
      Epouse : String (1 .. 20) := (others => ' ');
```

```
  end case;
```

```
end record;
```



EXEMPLE (2)

-- Appelé type "polymorphe" en Ada

```
type Ent_Flot (Est_Entier : Boolean) is record  
  case Est_Entier is  
    when True =>  
      Entier : Integer;  
    when False =>  
      Reel   : Float;  
  end case;  
end record;
```

Exemple 1

```
C1, C2, C3 : Client;
```

```
...
```

```
C1 := C2;
```

```
C3.Nom := C1.Nom;
```

Exemple 2

```
H1, H2 : Personne (Homme);
```

```
F : Personne (Femme);
```

```
...
```

```
H1 := H2; -- Possible
```

```
H1 := F; -- Impossible
```

```
F.Nom := H2.Nom; -- Possible
```

```
F.Nom := H2.Epouse; -- Impossible
```

- La notion de valeur immédiate pour un article doit être étendue
 - C'est un ensemble de valeurs immédiates
 - Tous les champs
 - Y compris le(s) discriminant(s)
- Il est intéressant de pouvoir représenter des valeurs immédiates composées



-- Le type

```
type Client is  
  record
```

```
    Nom, Prenom : String (1 .. 20) := (others => \ \');
```

```
    Num_Client  : Positive;
```

```
  end record;
```

-- Utilisation d'un agrégat article

```
Jacques : Client := ("Paul      ",  
                    "Jacques   ",  
                    125);
```




OPÉRATIONS PRÉDÉFINIES POUR LES TYPES ARTICLES

Affectation, comparaison d'égalité

 := et =, s'appliquent à l'article complet ou à un champ de l'article

