

Bazele programării I

Structuri de date neomogene

Tipul articol. Noțiune



Tipul articol este o structură neomogenă, care cuprinde un număr fix sau variabil de componente, care **pot fi de tipuri diferite**.



Componentele articolului se numesc **câmpuri**.



Un câmp al articolului este specificat prin 2 caracteristici:

- **Nume** (identificator);
- **Tip de date**.

Tipuri de articole


Tipul de date **articol** poate fi de **2 tipuri**:

- **Articole fixe.** Numărul componentelor este fix;
- **Articole cu variante.** Numărul componentelor este variabil în ceea ce privește numărul componentelor sau tipul acestora.

Articole fixe

Type

```
<NumeArticol> = Record  
    <NumeCâmp1> : < TipDate>  
    <NumeCâmp2> : < TipDate>  
    ...  
    <NumeCâmpn> : < TipDate>  
End
```



NumeCâmp – reprezintă identificatoare. În cadrul unui articol toate identificatoarele trebuie să fie diferite.

TipDate – reprezintă un tip de date elementar sau structurat.

Article fixe. Exem plu

Type

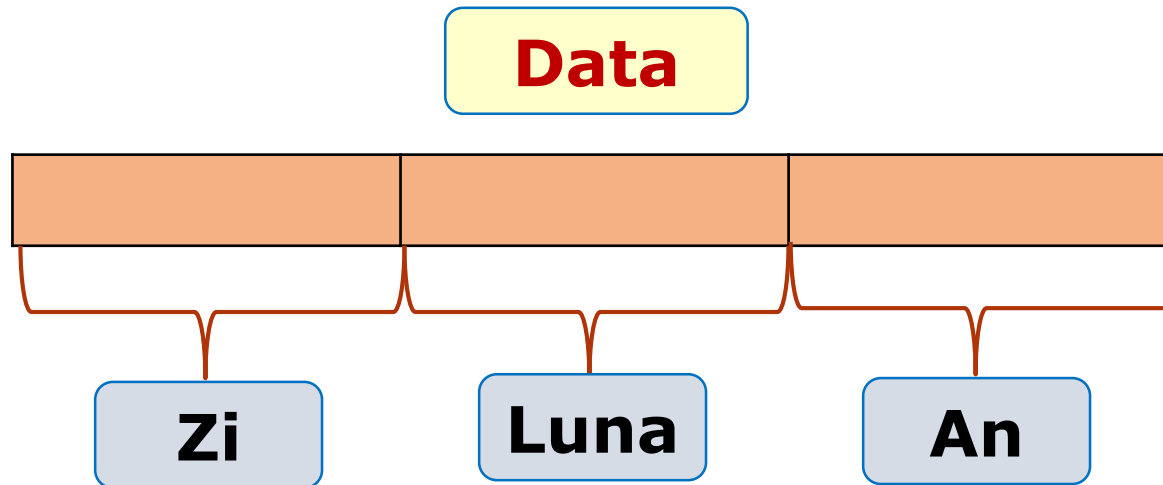
Data = Record

Zi: 1..31

Luna: 1..12

An: Natural

End



Articole fixe. Exemplu

Type

Data = Record

Zi: 1..31

Luna: 1..12

An: Natural

End

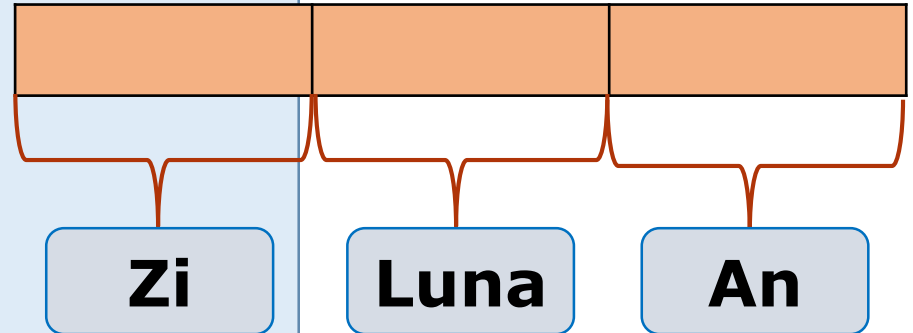
Persoana = Record

Nume: String[10]

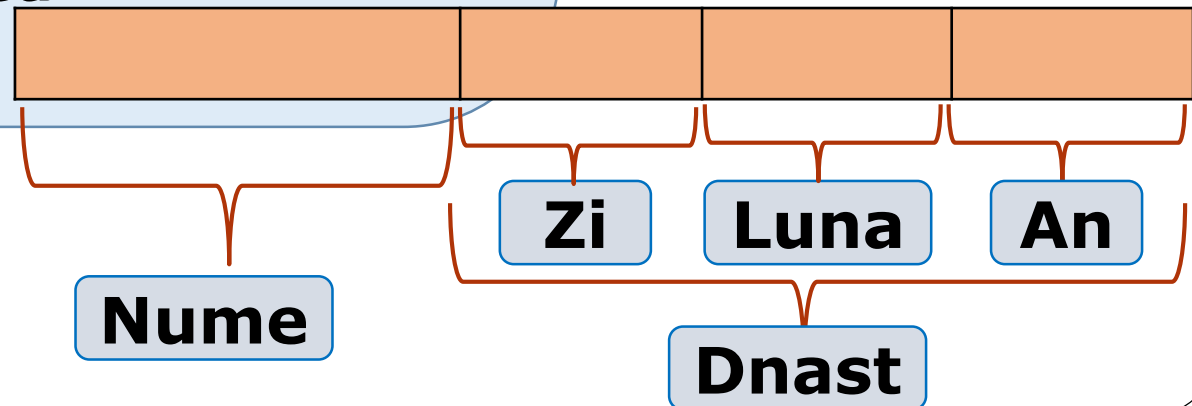
Dnast: Data

End

Data



Persoana



Article fixe. Exemplan

Type

Nota = 1..10

Examene = Array[1..6] Of Nota

Data = Record

Zi: 1..31

Luna: 1..12

An: Natural

End

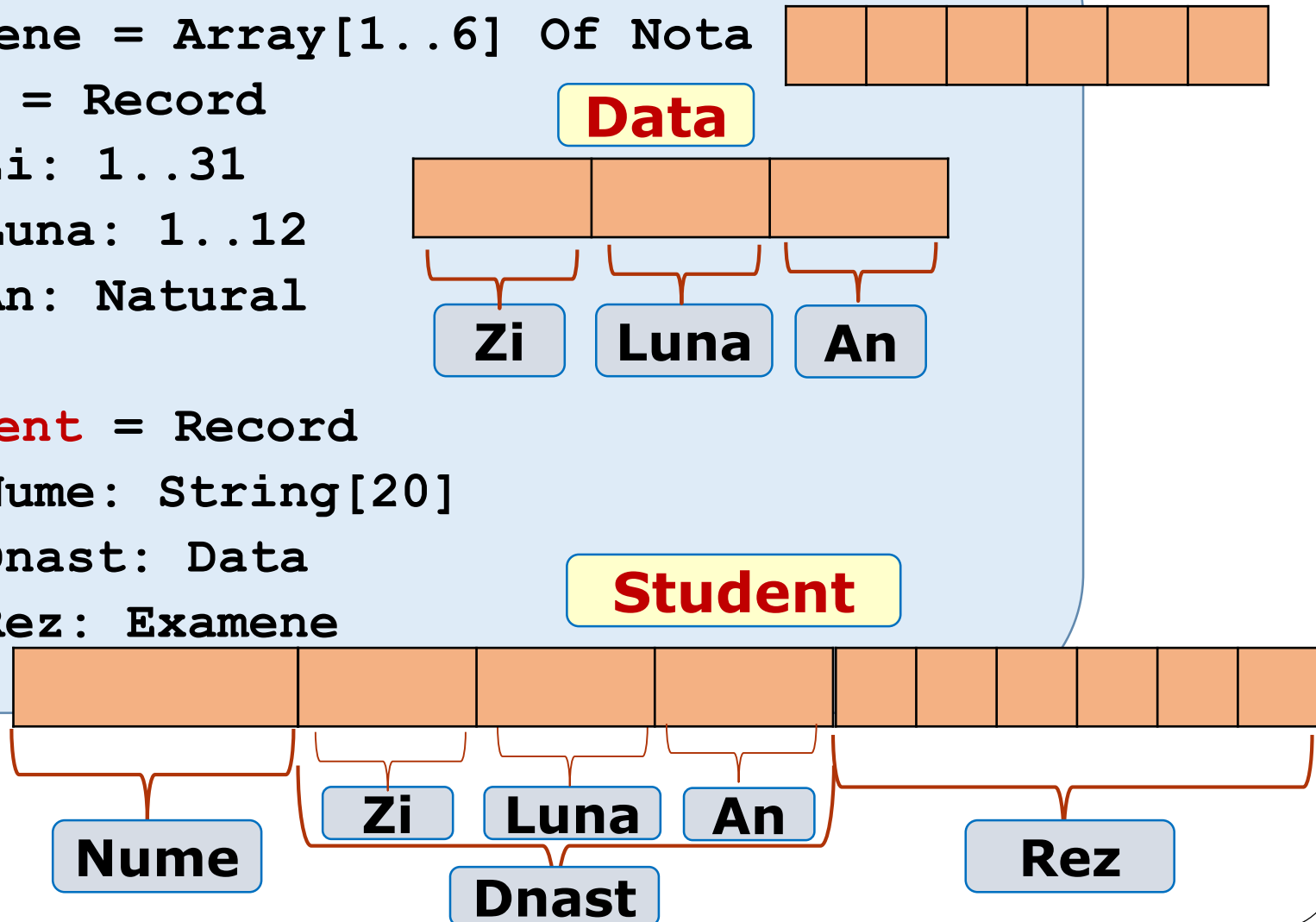
Student = Record

Nume: String[20]

Dnast: Data

Rez: Examene

End



Accesarea componentelor articolului

Var

D1: Data

D2: Data

A: Persoana

B: Persoana

C: Student

Accesarea unei componente articolului

Nume_variabila.Câmp

D1.Zi

D2.Luna

A.DNast.Luna

C.Rez[4]

Exemplu

Type

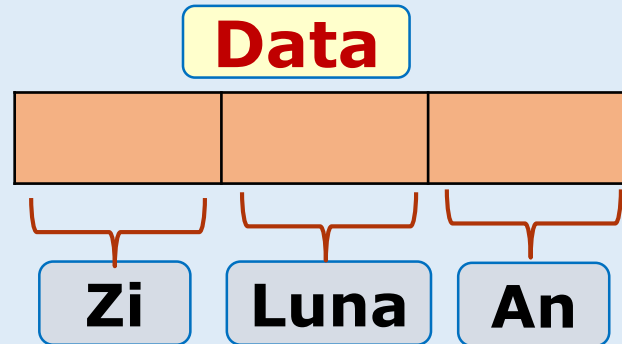
```
Data = Record  
  Zi: 1..31  
  Luna: 1..12  
  An: Natural
```

```
End
```

Var

```
D1: Data
```

```
D2: Data
```



| | | |
|----|----|------|
| 18 | 12 | 2015 |
| 9 | 5 | 1945 |

```
D1.Zi := 18  
D1.Luna := 12  
D1.An := 2015
```

```
D2.Zi := 9  
D2.Luna := 5  
D2.An := 1945
```

Exemplu

Type

Data = Record

Zi: 1..31

Luna: 1..12

An: Natural

End

Persoana = Record

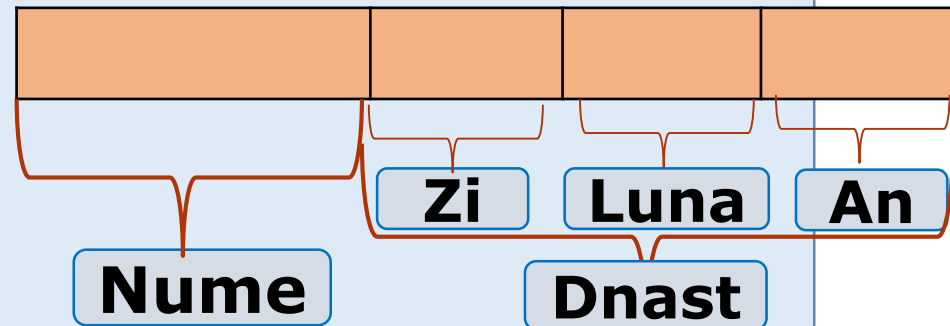
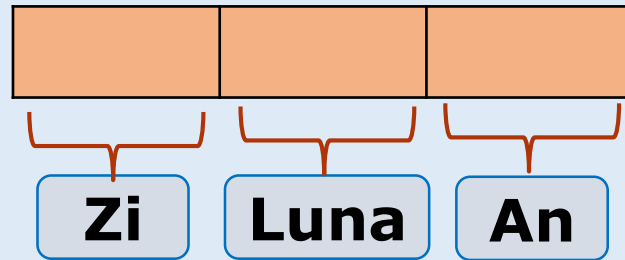
Nume: String[20]

Dnast: Data

End

Var

P1: **Persoana**



| | | | |
|--------|----|---|------|
| Ivanov | 18 | 5 | 1997 |
|--------|----|---|------|

```
P1.Nume := 'Ivanov'  
P1.Dnast.Zi := 18  
P1.Dnast.Luna := 5  
P1.Dnast.An := 1997
```

Exemplu

Type

Nota = 1..10

Examene = Array[1..4] Of Nota

Data = Record

Zi: 1..31

Luna: 1..12

An: Natural

End

Student = Record

Nume: Stri

Dnast: Data

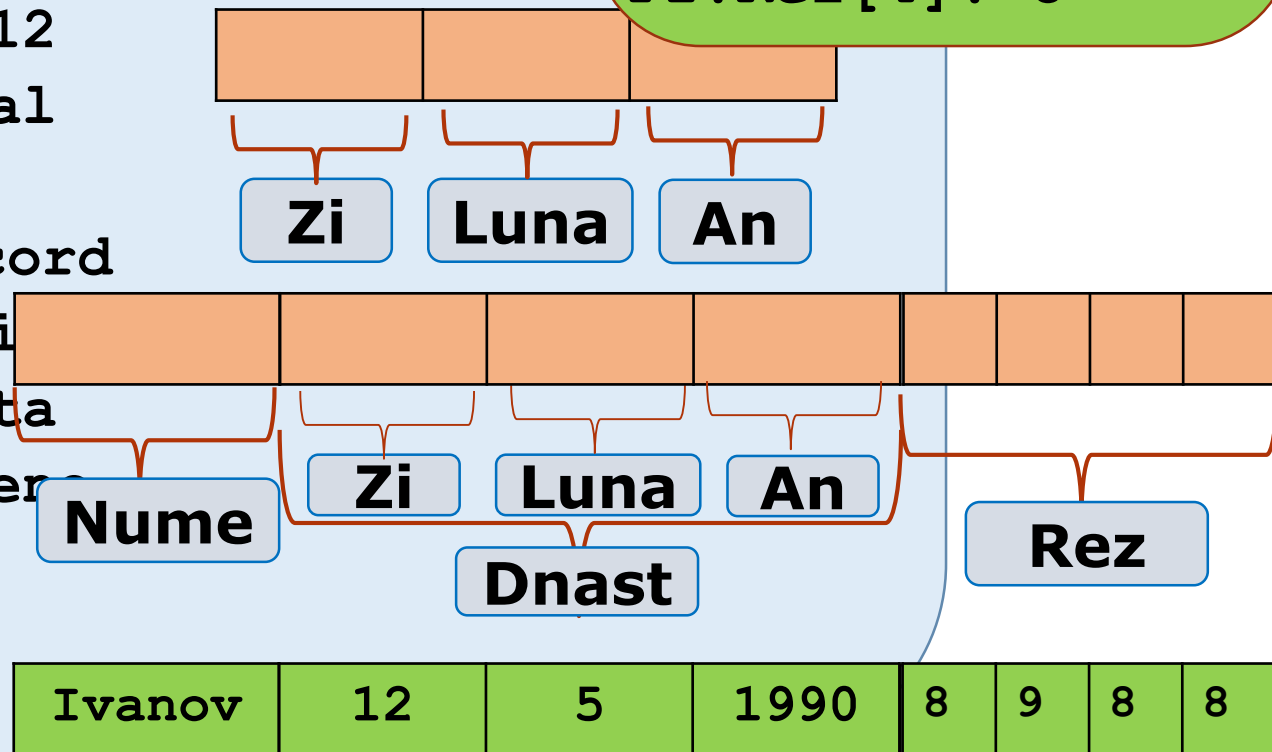
Rez: Examene

End

Var

P1: **Student**

P1.Nume := 'Ivanov'
P1.Dnast.Zi := 12
P1.Dnast.Luna := 5
P1.Dnast.An := 1990
P1.Rez[1] := 8
P1.Rez[2] := 9
P1.Rez[3] := 8
P1.Rez[4] := 8



Exemplu

Se consideră 2 cărți de joc K1 și K2. Să se elaboreze o funcție logică care determină dacă cartea K1 bate cartea K2, presupunând cozul Coz.

Cartea de joc are 2 caracteristici:

- **Culoare;**
- **Valoare.**

Culorile pot fi: *Verde, Rosu, Doba, Cruce.*

Valorile pot fi: *6,7,8,9,10, Valet, Dama, Crai, As.*

Exemplu

Type

Culoare = (Verde, Rosu, Doba, Cruce)

Valoare = (Sase, Sapte, Opt, Noua, Zece,
Valet, Dama, Crai, As)

Carte = Record

C: Culoare

V: Valoare

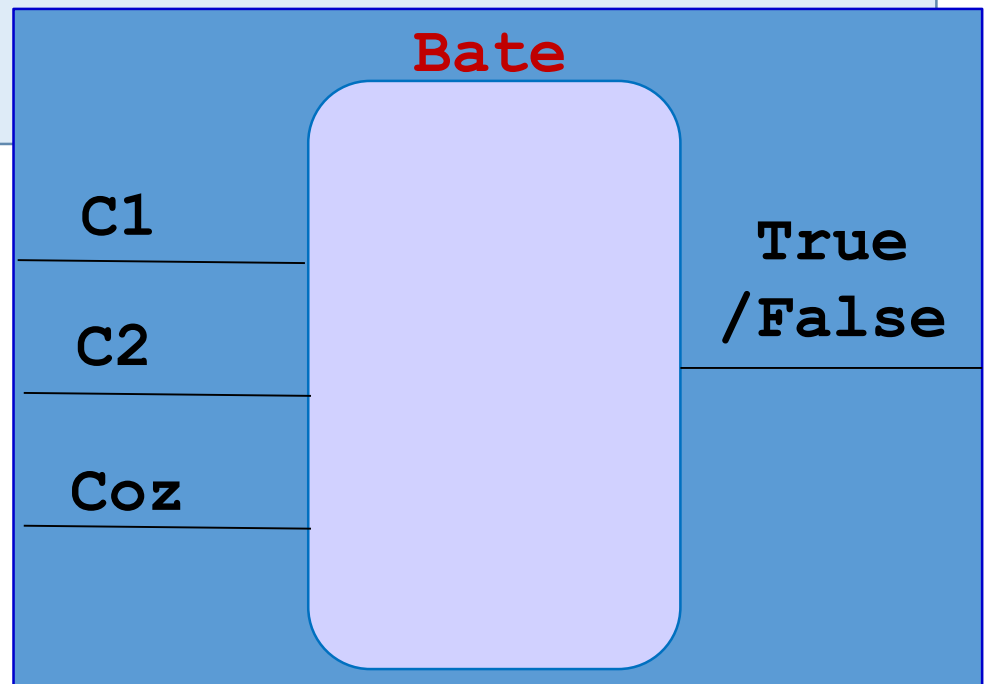
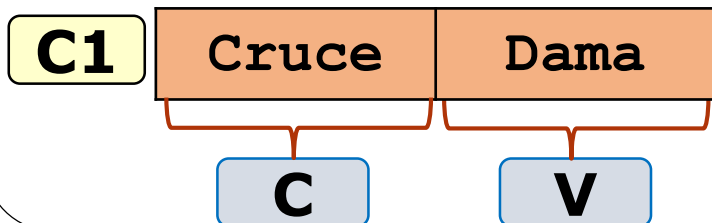
End

Var

C1, C2: **Carte**

C1.C := Cruce

C1.V := Dama



Exemplu

```
Function Bate (C1:Carte, C2:Carte, Coz:Culoare) :  
    Boolean  
Begin  
    If C1.C = C2.C Then  
        If C1.V > C2.V Then  
            Return True  
        Else  
            Return False  
        End  
    Else  
        If C1.C = Coz Then  
            Return True  
        Else  
            Return False  
        End  
    End  
End  
End
```

cărțile au aceeași culoare

cartea C1 este de coz

Tablouri formate din articole

Type

Nota = 1..10

Examene = Array[1..4] Of Nota

Data = Record

 Zi: 1..31

 Luna: 1..12

 An: Natural

End

Persoana = Record

 Nume: String[10]

 Dnast: Data

 Rez: Examene

End

Vector = Array[1..20] Of Persoana

Var

Grupa: Vector

Tablouri formate din articole

```
Grupa[I].Nume  
Grupa[I].Dnast.Zi  
Grupa[I].Dnast.Luna  
Grupa[I].Dnast.An  
Grupa[I].Rez[1]  
Grupa[I].Rez[2]  
Grupa[I].Rez[3]  
Grupa[I].Rez[4]
```

```
For I:=1 To 20 Step 1  
    ReadString(Grupa[I].Nume)  
    ReadNat(Grupa[I].DNast.Zi)  
    ReadNat(Grupa[I].DNast.Luna)  
    ReadNat(Grupa[I].DNast.An)  
    For J:=1 To 6 Step 1  
        Grupa[I].Rez[J]  
    End  
End
```


Exemple

Problema 1. Să se afișeze pe ecran informații despre persoanele grupei.

```
For I:=1 To 20 Step 1
    WriteString(Grupa[I].Nume)
    WriteNat(Grupa[I].DNast.Zi)
    WriteNat(Grupa[I].DNast.Luna)
    WriteNat(Grupa[I].DNast.An)
    For J:=1 To 4 Step 1
        WriteNat(Grupa[I].Rez[J])
    End
End
```

Problema 2. Să se afișeze numele persoanelor născute în anul 2003.

```
For I:=1 to 20 step 1
    If Grupa[I].DNast.An = 2003 Then
        WriteString(Grupa[I].Nume)
    End
End
```

Example

Problema 3. Să se determine, dacă există persoane, născute în anul 2005.

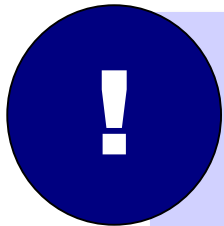
```
I:=1
While (I<=20) And
    (Not(Grupa[I].DNast.An = 2005)) Do
    I:=I+1
End
If I<=20 Then
    WriteString ('Există')
Else
    WriteString ('Nu există')
End
```

Exemple

Problema 4. Să se afișeze pe ecran lista persoanelor, indicând numele și vârsta persoanei.

```
An_curent:=2017  Luna_curent:= 11  Zi_curent:=13
For I:=1 to 20 step 1
  WriteString(Grupa[I].Nume)
  WriteString(' - ')
  Virsta := An_curent - Grupa[I].Dnast.An
  if Luna_curent < Grupa[I].DNast.Luna then
    Virsta := Virsta - 1
  else
    if (Luna_curent = Grupa[I].Dnast.Luna) and
      (Zi_curent < Grupa[I].Dnast.Zi) then
      Virsta := Virsta - 1
    end end
  WriteInt(Virsta)
End
```

Aritcole cu variante



Tipul **articol cu variante** reprezintă o structură care cuprinde o parte variantă, adică o parte structură cărei diferă de la un caz la altul.

Type

```
Nume = Record  
    Descrierea partea fixa  
    Descrierea partea variabila  
End
```

Aritcole cu variante. Exemplu

Type

Fel=(punct,triung,dreptun,cerc)

Figura = record

Nume:string[15]

Case F: Fel of

Punct:(x,y: Real)

Triung:(a,b,c: Natural)

Patrulat:(a,b,c,d: Natural)

Cerc:(x,y: Integer, r: Natural)

End

End

Exemplu

Var

F1: Figura

F2: Figura

F3: Figura

F4: Figura

F1.Nume:='Punctul_1'

F1.F:=Punct

F1.X:=3

F1.Y:=7

F2.Nume:='Triung'

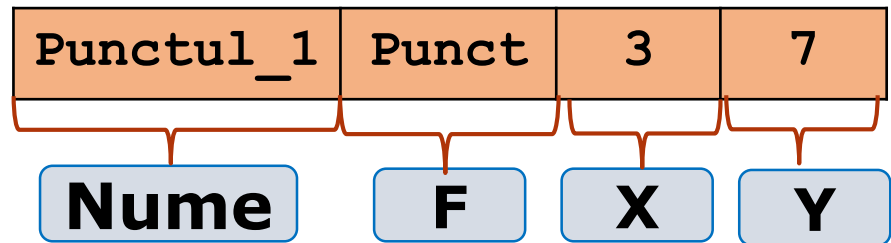
F2.F:=triung

F2.A:=3

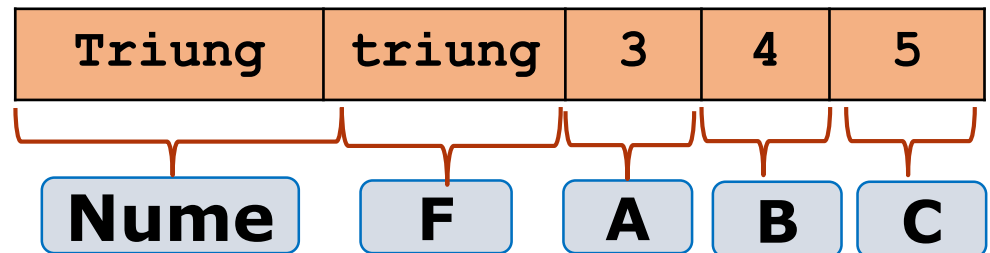
F2.B:=4

F2.C:=5

F1



F2



Exemplu

F3.Nume:='Patrulater'

F3.F:=patrulat

F3.A:=3

F3.B:=4

F3.C:=5

F3.D:=6

F3

Patrulater

patrulat

3

4

5

6

Nume

F

A

B

C

D

F4.Nume:='Cerc_1'

F4.F:=cerc

F4.X:=0

F4.Y:=0

F4.R:=8

F4

Cerc_1

cerc

0

0

8

Nume

F

X

Y

R