Structuri, uniuni și enumerări

CURS 4- 23.03.2021

Titular: Șef. Lucr. Dr. Mat. Cărbureanu Mădălina

Universitatea Petrol-Gaze din Ploiești

Obiective:

- **Noțiunile de structură, uniune și enumerare;**
- **Declarația unei structuri, uniuni și enumerări;**
- **Accesul la câmpurile unei structuri, uniuni și enumerări;**
- **Diferența dintre structuri și uniuni;**
- **Observații;**
- **Exemplu aplicație:**
  - **Declarație *struct stud*** ⇨ **set cerințe;**
- **Aplicații propuse.**

Noțiunile de structură, uniune și enumerare

Alte tipuri de date structurate?

- Structuri+uniuni+enumerări ⟹ tipuri de date structurate;
- Structură ⟹ colecție de date neomogene grupate sub același nume;
  ⟹ câmpuri diferite ocupă zone de memorie diferite!!!;
- Structuri

Utilitate
Avantaj

Noţiunile de structură, uniune şi enumerare

- Uniune ⟺ reuniune;
- <u>Uniune</u> ⟹ tip de date structurat similar cu structura;

  ⟹ câmpuri diferite ocupă aceiaşi zonă de memorie!!!;

- Uniuni

Utilitate
Avantaj

Noțiunile de structură, uniune și enumerare

Alternativă la?

- <u>Enumerări</u> ⟹ liste de constante întregi cu nume;

  ⟹ tipuri unice cu valori ce aparțin unei mulțimi de constante (enumeratori);

  ⟹ liste de elemente cu valori unice.

- Enumerări

  Utilitate

Declaraţia unei structuri, uniuni şi enumerări

- **<u>Declaraţia unei structuri:</u>**

struct nume_structură

{ *tip nume_câmp_1;*

tip nume_câmp_2;

…………..

tip nume_câmp_n;

} *[<listă_variabile_structură>];*

Declaraţia unei structuri, uniuni şi enumerări

- Exemple de structuri:

struct student {char nume[20],pren[20]; int varsta; char facultate[20], spec[20]; int an; } stud1, stud2, *stud3;	struct persoana { struct student stud; float salariu; char adresa[20]; }pers1;	struct noname {int a; int b; }s; int noname; float a;	typedef struct lista {int inf; struct lista*leg; }stiva;

Declaraţia unei structuri, uniuni şi enumerări

- **<u>Declaraţia unei uniuni:</u>**

union nume_uniune

{ *tip nume_câmp_1;*

tip nume_câmp_2;

…………..

tip nume_câmp_n;

} *[<listă_variabile_uniune>];*

Declarația unei structuri, uniuni și enumerări

- Exemplu de uniune:

```
union elev
{ char nume[10], pren[10];
  int varsta;
  float nota_bac;
}e;
```

Declarația unei structuri, uniuni și enumerări

- <u>Declarația unei enumerări:</u>

  enum nume_enumerare {lista_enum} lista_de_variabile;

- Exemplu de enumerare:

enum lunile_anului {ianuarie, februarie, martie, aprilie, mai, iunie, iulie, august, septembrie, octombrie, noiembrie, decembrie} luna;

Accesul la câmpurile unei structuri, uniuni și enumerări

- Variabilele din *<listă_variabile_structură>* declarate prin nume $\Rightarrow$ *dot* (.):

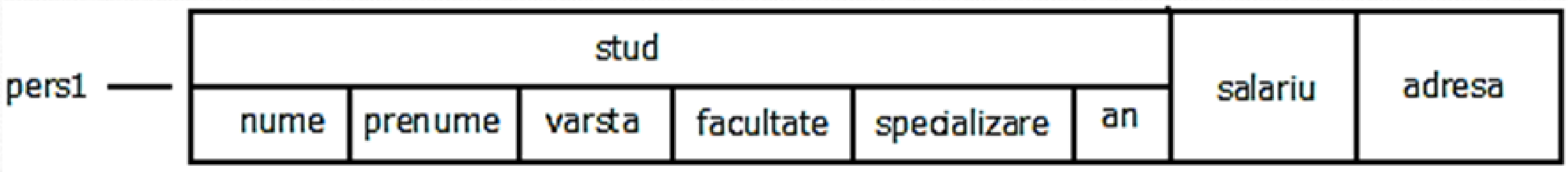
  nume_variabilă_structură.nume_câmp;

- Variabilele din *<listă_variabile_structură>* sunt pointer către structură $\Rightarrow$ →:

  nume_pointer_structură→ nume_câmp;

Accesul la câmpurile unei structuri, uniuni și enumerări

- Accesul la câmpurile unei structuri imbricate:



- Accesul la câmpurile unei uniuni:

luna=martie; sau luna=3;

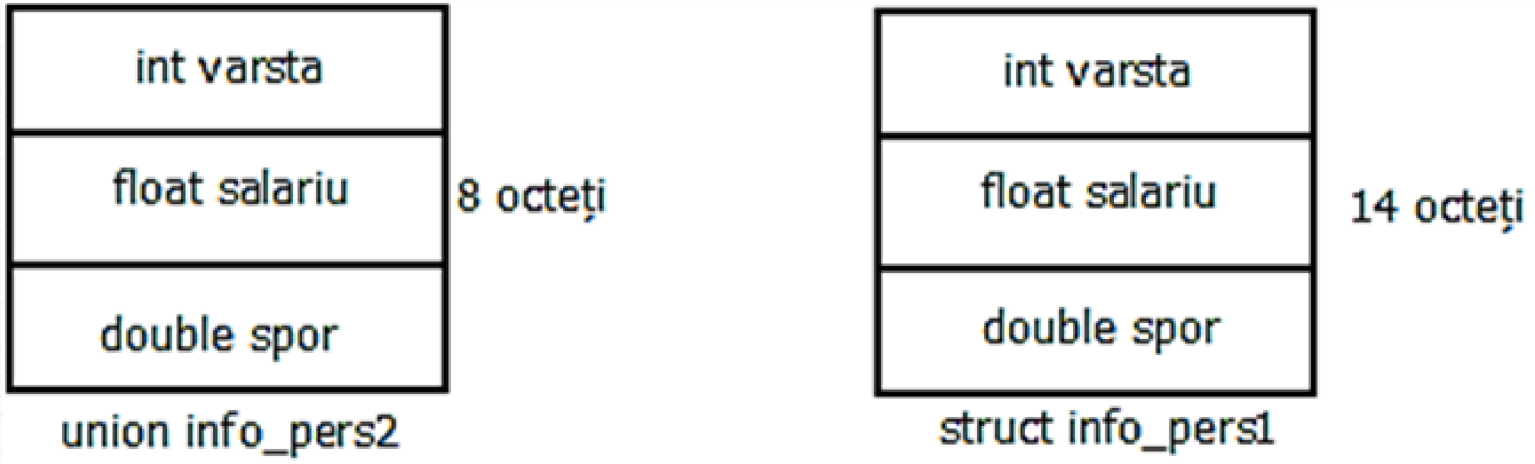
Diferența dintre structuri și uniuni

- **Modul de utilizare a memoriei de către câmpuri:**
  - Structuri ⇨ zonele de memorie rezervate câmpurilor sunt diferite pentru câmpuri diferite;
  - Uniuni ⇨ toate câmpurile din uniune împart aceeași zonă de memorie;
- Calcul spațiu de memorie:
  - sizeof (struct nume_structură);
  - sizeof (union nume_uniune);

Diferența dintre structuri și uniuni

union info_pers1 { int varsta; float salariu; double spor; };	struct info_pers2 { int varsta; float salariu; double spor; };

Diferența dintre structuri și uniuni



Observații

- **inițializare variabilă de tip structură:**

  nume_structură nume_var={listă_valori};

- **exemplu:**

  student stud1={“Horia”, “Maria”, 22, “IME”, “AIA”, 3};

- **declararea unui șir de caractere:**

  char nume_șir[dim];

  char* nume_sir;

Observaţii

- **Funcţii pentru prelucrarea şirurilor de caractere:**
  - scanf() şi printf(), cu descriptorul de format %s;
  - gets(şir_destinaţie);
  - puts (şir);
  - strcpy(sir_destinaţie, sir_sursă);
  - int strcmp(şir1, şir 2);
  - strlen(şir);
  - char *strcat(şir1, şir2).

Exemplu aplicație

```
#include<iostream.h>
#include<string.h>
#include<conio.h>
#include<stdio.h>
struct stud
{char nume[10], pren[10];
 long int cnp;
 int nrex;
 int nlab, notad, nex;
}a[10],temp,temp1,aux1;
 void main()
{clrscr();
 int n,i,j, max,nrrest,aux;
long int x;
 //citire informatii studenti
```

```
cout<<"Nr. studenti=";cin>>n;
 for(i=1;i<=n;i++)
 {cout<<"Nume stud"<<i<<"este:"; cin>>a[i].nume;
  cout<<"Prenume stud"<<i<<"este:";cin>>a[i].pren;
  cout<<"Cnp stud"<<i<<"este:";cin>>a[i].cnp; cout<<"Nr. examene stud"<<i<<"este:";cin>>a[i].nrex;
  cout<<"Nota laborator stud"<<i<<"este:"; cin>>a[i].nlab;
  cout<<"Nota examen stud"<<i<< "este:“ ; cin>>a[i].nex;
  //calcul nota finala pt. studenti
  a[i].notad=(a[i].nlab+a[i].nex)/2;
  cout<<"Nota finala-nota catalog student"<<i<<" este:"<<a[i].notad; }
//afisare informatii studenti
for(i=1;i<=n;i++)
{cout<<"\nNume stud"<<i<<"este:"<<a[i].nume; cout<<"\nPrenume stud"<<i<<"este:"<<a[i].pren;
 cout<<"\nCnp stud"<<i<<"este:"<<a[i].cnp; cout<<"\nNr. examene stud"<<i<<"este:"<<a[i].nrex;
 cout<<"\nNota laborator stud"<<i<<"este:"<<a[i].nlab;
 cout<<"\nNota examen stud"<<i<<"este:"<<a[i].nex;
 cout<<"\nNota finala-nota catalog stud"<<i<<"este:"<<a[i].notad;}
```

```
//numele si prenumele celui mai bun student
max=a[1].notad;
for(i=2;i<=n;i++)
if(a[i].notad>max) max=a[i].notad;
cout<<"\nNota cea mai mare la examen este:"<<max;
for(i=1;i<=n;i++)
if(a[i].notad==max) cout<<"\nNumele celui mai bun student este"<<a[i].nume<<"iar prenumele este" <<
a[i].pren;
//numarul de studenti restanti la examen
nrrest=0;
for(i=1;i<=n;i++)
{if(a[i].notad<5) nrrest++; }
cout<<"\nNr. de studenti restanti este:"<<nrrest;
//sortare descrescatoare nota examen disciplina
for(i=1;i<=n-1;i++)
for(j=i+1;j<=n;j++)
{if(a[i].notad<a[j].notad) {int aux=a[i].notad;  a[i].notad=a[j].notad; a[j].notad=aux; }}
```

```
cout<<"\nNote examen in ordine descrescatoare sunt:";
for(i=1;i<=n;i++) cout<<a[i].notad;
// sortarea crescatoare a studentilor dupa nume
 for(i=1;i<=n-1;i++)
 for(j=i+1;j<=n;j++)
 {if((strcmp(a[i].nume,a[j].nume)>0))
    {temp1=a[i]; a[i]=a[j];  a[j]=temp1;}}
cout<<"\nAranjare crescatoare a studentilor dupa nume-lexicografic:";
for(i=1;i<=n;i++)
cout<<"\n"<<a[i].nume<<" "<<a[i].pren<<" "<<endl;
cout<<"\nIdentificare student cu un anumit CNP";
cout<<"\nCNP-ul cautat este:"; cin>>x;
for(i=1;i<=n;i++)
{if(a[i].cnp==x)  cout<<"Studentul cu CNP-ul cautat este:"<<a[i].nume<<a[i].pren; }
```

```
//Afisare nume si prenume student pentru care numarul de examene este >4
cout<<"\nNume si pren student cu nrex>4:";
for(i=1;i<=n;i++)
{if(a[i].nrex>4)  cout<<a[i].nume<<" "<<a[i].pren; }
//ordonarea crescatoarea a studentilor cu acelasi nume dupa prenume
cout<<"\nOrdonarea descrescatoare a studentilor cu acelasi nume dupa prenume:";
 for(i=1;i<=n-1;i++)
 for(j=i+1;j<=n;j++)
 {if(strcmp(a[i].nume,a[j].nume)==0 && strcmp(a[i].pren,a[j].pren)<0)
    {aux1=a[i]; a[i]=a[j]; a[j]=aux1; } }
for(i=1;i<=n;i++)
cout<<a[i].nume<<" "<<a[i].pren;
 getch();
}
```

Aplicaţii propuse

- **Testarea aplicaţiilor 4.2.1, 4.2.2 şi 4.2.3, pag. 63-70;**
- **Rezolvarea aplicaţiilor nr. 1, 2 şi 3, pag. 70-72;**
- **Obs: "*Elemente de proiectarea algoritmilor. Ghid teoretic şi practic*", Şef lucr. dr. mat. Cărbureanu Mădălina, Editura Universităţii Petrol-Gaze din Ploieşti, 2021.**

Să vă fie de folos și spor la lucru!