

## Tipul struct

Forma generală pentru declararea variabilelor de tip struct:

```
1. struct nume_structură
    { tip1 nume câmp [,nume câmp, ...];
      [tip2 nume câmp [,nume câmp , ...];
      .....
      [tipn nume câmp [,nume câmp , ...];
    }listă variabile;
```

unde

nume\_structură - desemnează numele sub care este cunoscut tipul structurat definit în program

tip1...tipn - reprezintă tipurile de date din care se compune structura

nume câmp - desemnează identificatorii datelor care compun structura (membrii structurii)

**Exemplu:**

```
struct elev
    {char nume[20], prenume[20];
      float medie;
      int clasa;
    }x;
```

Variabila x este structurată în câmpurile: **x.nume**, **x.prenume**, **x.medie** și **x.clasă**

```
2. struct nume_structură
    { tip1 nume câmp [,nume câmp, ...];
      [tip2 nume câmp [,nume câmp, ...];
      .....
      [tipn nume câmp [,nume câmp, ...];
    };
```

```
struct nume_structură listă variabile;
```

//sau

```
nume_structură listă variabile;
```

**Exemplu:**

```
struct elev
    {char nume[20],prenume[20];
      float medie;
      int clasa;
    };
```

```
struct elev x;
```

//sau

```
elev x;
```

```
3. struct
    { tip1 nume câmp ,nume câmp, ...;
      [tip2 nume câmp ,nume câmp , ...;
      .....
      tipn nume câmp ,nume câmp , ...;
    } listă variabile;
```

**Exemplu:**

```
struct
    {char nume[20],prenume[20];
      float medie;
      int clasa;
    }x;
```

**Observație 1:**

În acest caz numele structurii lipsește, spunem că avem o **structură anonimă** și din acest motiv variabilele trebuie declarate odată cu structura, precizându-le după }.

## Observație 2:

**Tipul *struct* descrie doar modul cum sunt organizate datele**, asemenea tipurilor predefinite, nu rezervă o zonă de memorie în care datele se păstrează. Pentru a memora datele **este obligatorie declararea unei variabile** de tip `nume_structură`.

**Declararea variabilelor de tipul structurii definite** se poate realiza în două moduri:

- După cum putem observa din forma generală, variabilele se pot declara după definirea structurii. În acest caz structura poate fi și anonimă, adică `<nume structură>` poate să lipsească.
- Oriunde în program, în același mod cu declararea variabilelor pentru tipurile predefinite (dacă structura nu este anonimă):

`<nume structură> <nume var1>, <nume var2>, ..., <nume varn>;`

În acest caz `<nume structură>` este obligatoriu, la definirea structurii.

## Exemplu:

```
struct elev e, v[30];  
elev e, v[30];
```

**Observație:** Putem declara variabile de tip tablou cu elemente de tip structurat, sau câmpurile unei structuri pot fi de tip tablou.

## 4. Cu definire de tip utilizator

### `typedef struct`

```
{ tip1 nume câmp ,nume câmp, ...;  
  [tip2 nume câmp ,nume câmp , ...;  
  .....  
  tipn nume câmp ,nume câmp , ...;  
} nume_structura;
```

`nume_structura` listă variabile;

exemplu:

### `typedef struct`

```
{ char marca[10];  
  int pret;  
} masina;
```

`masina m;`

## Modul de adresare

Pentru a putea accesa un anumit câmp din interiorul unei structuri vom folosi **operatorul de selecție directă "."**, acest operator are **prioritate maximă**.

## Exemplu:

### 1. `struct elev`

```
{char nume[20],prenume[20];  
  float medie;  
  int clasa;  
} x[30];
```

Reprezintă declararea unei variabile `x`, de tip tablou unidimensional, cu maxim 30 de elemente: `x[0]`, ..., `x[29]`. Fiecare element are câmpurile `nume`, `prenume`, `medie`, `clasa`. Pentru a le parcurge vom scrie:

```
for(int i=0;i<30;i++)
```

```
{//prelucrează x[i].nume, x[i].prenume,x[i].medie,x[i].clasa;  
}
```

### 2. `struct elev`

```
{char nume[20];
```

```
int note[5];
} x;
```

Reprezintă declararea unei variabile x, care are câmpurile nume și note, dar câmpul note este de tip tablou unidimensional cu 5 elemente indexate de la 0 la 4. Pentru a le parcurge vom scrie:

```
//prelucrează x.nume,
for(int i=0;i<5;i++)
    // prelucrează x.note[i];
```

### 3. struct elev

```
{char nume[20];
int note[5];
}x[30];
```

Reprezintă declararea unei variabile x, de tip tablou unidimensional, cu maxim 30 de elemente: x[0], ..., x[29]. Fiecare element are câmpurile nume, și note, în care câmpul note este la rândul lui un tablou cu 5 elemente indexate de la 0 la 4. Pentru a le parcurge vom scrie:

```
for(int i=0;i<30;i++)
{
    //prelucrează x[i].nume,
    for(int j=0;j<5;j++)
        //prelucrează x[i].note[j];
}
```

Definirea unui **tip** structură poate fi plasată oriunde în program, acest tip va fi cunoscut din momentul definirii lui până în momentul încheierii structurii în interiorul careia a fost definit.

#### typedef struct

```
{ tip1 <nume câmp> ,<nume câmp>, ...;
tip2 <nume câmp > ,<nume câmp >, ...;
.....
tipn <nume câmp > [,<nume câmp >, ...;
} <nume structură>;
```

Pentru declararea variabilelor vom scrie:

```
<nume structură>[listă variabile];
```

#### Modul de adresare

Pentru exemplul dat și declarația de variabile **elev articol** ne vom adresa unui câmp din variabila articol astfel:

```
articol.nume;          articol.prenume;          articol.medie;          articol.clasa;
```

O variabilă de tip structurat poate fi **copiată** într-o altă variabilă de același tip cu ajutorul instrucțiunii de atribuire.

**Exemplu:** v[1]=v[2];e=v[i];

#### Observație: tablouri bidimensionale cu date de tip struct

struct element

```
{int x,y,z;} a[10][10];
```

Va memora în tabloul a maxim  $10 \cdot 10 \cdot 3 = 300$  numere întregi

În fiecare elemet a[i][j] avem câmpurile a[i][j].x, a[i][j].y, a[i][j].z

#### Structuri imbricate

Este posibilă definirea unei structuri ale cărei componente sunt de un alt tip structură, definite anterior. În acest caz câmpurile de tip structură se numesc structuri imbricate (incluse).

Se pot realiza mai multe nivele de imbricări.

Se va define mai întâi structura elementară, putând declara sau nu variabile corespunzătoare acesteia și apoi structura compusă care conține câmpuri de tipul structurii elementare

**Exemplu:**

```
#include <iostream>
using namespace std;
```

```
struct data
{int zi,luna,an;
};
```

**struct elev**

```
{ char nume[10],prenume[20];
  data datan;           //datan –câmp de dată de tip structurat elementar cu numele data
  float medie;
  int clasa;
};
elev a;
```

```
/* sau
```

**struct elev**

```
{ char nume[10],prenume[20];
  struct data
  {int zi,luna,an;
  } datan;           //datan –câmp de dată de tip structurat elementar cu numele data
  float medie;
  int clasa;
};
*/
```

```
int main()
```

```
{ cout<<"nume:";cin>>a.nume;
  cout<<"prenume:";cin>>a.prenume;
  cout<<"zi:";cin>>a.datan.zi;           //modul de adresare al datelor imbricate
  cout<<"luna:";cin>>a.datan.luna;
  cout<<"an:";cin>>a.datan.an;
  cout<<"medie:";cin>>a.medie;
  cout<<"clasa:";cin>>a.clasa;
  cout<< a.nume<<" "<<a.prenume<<" ";
  cout<<a.datan.zi<<"/"<<a.datan.luna<<"/"<<a.datan.an;
  cout<<" "<<a.medie<<" "<<a.clasa;
  return 0; }
```

```
sau
```

**structura elementară poate fi declarată direct în structura compusă**

**struct elev**

```
{ char nume[10],prenume[20];
  struct data
  {int zi,luna,an;
  } datan;           //datan –câmp de dată de tip structurat elementar cu numele data
  float medie;
  int clasa;
};
```

**Exemplu vectori cu elemente de tip struct**

Fiecare element al vectorului are câmpuri corespunzătoare structurii

Exemplu:

```
struct elev{ char nume[20];
             float media;} v[30];
```

```
int n;
```

citirea datelor corespunzătoare vectorului:

```
cin>>n; cin.get();
for(int i=1;i<=n;i++)
{ cin.get(v[i].nume,20);
```

```
cin.get();
cin>>v[i].media;
cin.get();
}
```

în același mod se parcurge vectorul analizând câmpurile fiecărui `v[i]`

### Exemplu de structură care are câmpuri de tip vector

```
struct alocatie
{
    char nume[30];
    long long cnp;
    int aloc[12];
} copil;
```

Câmpurile sunt:

```
copil.nume
copil.cnp
copil.aloc[0], copil.aloc[1]....copil.aloc[11];
```

### Probleme

1. Definiți un tip înregistrare punct pentru a memora coordonatele unui punct din plan. Definiți un tip înregistrare triunghi pentru a memora coordonatele celor 3 varfuri ale unui triunghi. Cititi 6 valori care reprezintă coordonatele celor 3 varfuri ale unui triunghi. Calculati si afisati perimetrul triunghiului citit.  
Exemplu: 0 0 1 0 0 1  
perimetrul este 3.4142
2. Declarati un tip înregistrare pentru memorarea unei date calendaristice.Cititi doua date calendaristice. Afisati-o pe cea mai mare in ordine cronologica.
3. Definiti un tip de date pentru memorarea lungimii si latimii unui dreptunghi. Cititi un numar natural  $n$  si apoi  $n$  perechi de numere intregi reprezentand  $n$  dreptunghiuri.
  - a) Afisati laturile dreptunghiului cu arie maxima.
  - b) Afisati laturile dreptunghiului cu perimetrul minim.
4. Se citesc urmatoarele informatii despre fiecare elev dintr-o clasa: numele(char nume[20]), prenumele (char prenume[30]) si media(float media).Sa se afiseze elevii din clasa ordonati descrescator dupa medie.
5. Definiți structura fracție ce memorează numărătorul și numitorul unei fracții. Dacă se citesc  $n$  fracții se cere afișarea acestora în formă ireductibilă.