

# Sesiunea 4

---

## Agenda

- Recapitulare concepte din sesiunile anterioare si/sau predate la scoala
- Recapitulare exercitii de antrenament din sesiunile anterioare
- Oleaca de teorie pentru structuri
- Exerciții cu structuri de incalzire
- Exerciții cu structuri pb-info
- Exerciții cu structuri extrase din variante de bacalaureat
- Exerciții de antrenament

## Recapitulare teorie structuri de date

O structură de date este o colecție de informații grupate sub un singur nume. Aceste informații, cunoscute drept membri, pot avea tipuri de date diferite și lungimi diferite. Structurile de date pot fi definite în C++ folosind următoarea sintaxă:

```
struct nume_tip {  
    tip_data_1 id_membru_1;  
    tip_data_2 id_membru_2;  
    tip_data_3 id_membru_3;  
    .  
    .  
} denumiri_obiecte;
```

- `nume_tip` este o denumire pentru tipul structură, efectiv cum vrem noi sa botezam aceasta grupare de date. De exemplu, daca am grupa informatii precum nume, prenume, varsta, adresa, email, putem sa numim aceasta grupare `persoana`
- `denumiri_obiecte` poate fi un set de identificatori valizi de obiecte care sunt de acel tip structură. Sau, mai pe limba noastra mai lenesa, e ca si cum ti-ai declara deja cateva variabile de tipul `nume_tip`
- Între acolade `{}` avem o listă cu informațiile membre, fiecare dintre ele fiind precizată cu tip de dată și un identificator. Efectiv toate informațiile pe care noi vrem sa le grupam si la care vrem sa le dam sens, atunci cand le folosim impreuna.
- De exemplu:

```
struct produs {  
    int greutate;  
    double pret;  
};
```

```
produs mar;  
produs banana, pepene;
```

Avem o definiție de tip structură, denumit **produs**, care are doi membri: **greutate** și **pret**, fiecare de un alt tip de dată fundamental. Această definiție creează un nou tip (**produs**), care este apoi folosit pentru a declara trei obiecte (**variabile**) de acest tip: **mar**, **banana** și **pepene**. Să observăm că, odată definit, tipul **produs** poate fi folosit la fel ca orice alt tip de dată fundamental (gen **int**, **double**, **float**, etc.).

Chiar la sfârșitul definiției struct, înainte de simbolul (**;**), câmpul opțional **denumiri\_obiecte** poate fi folosit pentru a declara mai direct obiecte de acel tip structură. De exemplu, obiectele structură **mar**, **banana** și **pepene** pot fi declarate simultan cu definiția tipului de dată structură:

```
struct produs {  
    int greutate;  
    double pret;  
} mar, banana, pepene;
```

În acest caz, când precizăm **denumiri\_obiecte**, numele tipului de dată (**produs**) devine opțional:

- **NOTA:** **struct** necesită fie un **nume\_tip** fie cel puțin o **denumire\_obiect**, dar nu neapărat amândouă simultan.
- **NOTA 2:** Este foarte important să înțelegem diferența dintre numele tipului de dată (**produs**) și un obiect de acest tip (**mar**, **banana** și **pepene**). Putem declara mai multe obiecte (ca **mar**, **banana** și **pepene**) cu un singur tip structură (**produs**).

O dată declarate cele trei obiecte de un anumit tip (**mar**, **banana** și **pepene**), putem accesa informațiile membru ale lor direct. Sintaxa pentru aceasta presupune inserarea simbolului punct (**.**) între denumirea obiectului și identificatorul membrului. De exemplu, am putea opera cu oricare dintre aceste elemente ca și cum ele ar fi variabile obișnuite declarate cu propriul lor tip de dată:

```
mar.greutate  
mar.pret  
banana.greutate  
banana.pret  
pepene.greutate  
pepene.pret
```

Fiecare dintre ele are tipul de dată corespunzător membrului la care se referă: \* **mar.greutate**, **banana.greutate** și **pepene.greutate** sunt de tip **int**, \* în timp ce **mar.pret**, **banana.pret** și **pepene.pret** sunt de tip **double**.

Avem aici un exemplu real de folosire a tipului structură în care salvăm date despre filmul nostru favorit. Putem observa cum putem initializa un membru, direct dar și de la tastatură:

```
#include <iostream>
#include <string.h>
using namespace std;

struct FilmFavorit{
    char titlu[51];
    int anLansare;
} al_meu, al_tau;

void afisareFilm (FilmFavorit film);

int main ()
{
    strcpy(al_meu.titlu, "Lord of the Rings: Return of the King\0");
    al_meu.anLansare = 2003;

    cout << "Tasteaza titlul filmului tau favorit: ";
    cin.getline(al_tau.titlu, 51);
    cout << "Tasteaza anul in care a fost lansat: ";
    cin >> al_tau.anLansare;

    cout << "Filmul meu favorit: " << endl;
    afisareFilm(al_meu);
    cout << "Filmul tau favorit: " << endl;
    afisareFilm(al_tau);
    return 0;
}

void afisareFilm (FilmFavorit film)
{
    cout << "Titlu: " << film.titlu << " si a fost lansat in anul: "
    << film.anLansare << endl;
}
```

Pentru final, o sa vedem un exemplu in care citim de la tastatura un vector in care fiecare element, reprezinta o variabila de tipul FilmFavorit:

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

struct FilmFavorit{
    char titlu[51];
    int anLansare;
};

void afisareFilm (FilmFavorit film);

int main ()
{
    int n;
```

```

        cout << "Cate filme doriti sa introduceti: ";
        cin >> n;
        FilmFavorit filmeFavorite[n];

        for (int i = 0; i < n; i++) {
            cout << "Introduceti titlul pentru filmul #" << i+1 << " (max
51 caractere):" << endl;
            cin.ignore();
            cin.getline(filmeFavorite[i].titlu, 51);
            cout << "Introduceti anul in care filmul a fost lansat: " <<
endl;
            cin >> filmeFavorite[i].anLansare;
        }

        cout << "Ati introdus urmatoarele filme: " << endl;
        for (int i = 0; i < n; i++) {
            cout << "#" << i+1<<": " << endl;
            afisareFilm(filmeFavorite[i]);
            cout << endl;
        }
        return 0;
    }

    void afisareFilm (FilmFavorit film)
    {
        cout << "Titlu: '" << film.titlu << "' lansat in anul: " <<
film.anLansare << endl;
    }
}

```

- NOTA: Daca s-a inteles tot ce s-a scris mai sus, problemele de la bacalaureat, pe acest topic, nu au cum sa puna vreo dificultate.

## Exercitii cu structuri de incalzire

1. Enunt: Creați o structură numită **Elev** care să conțină următoarele câmpuri: **Nume** (șir de caractere) **Nota1** (float) **Nota2** (float) **Nota3** (float) Declarați două variabile de tip **Elev** pentru doi elevi diferiți și citiți datele lor (nume, nota1, nota2 și nota3) de la tastatură.

Calculați media notelor pentru fiecare elev și afișați rezultatele în următorul format: Pentru elevul 1: Nume - [NumeElev1], Medie - [MediaElev1] Pentru elevul 2: Nume - [NumeElev2], Medie - [MediaElev2] Asigurați-vă că valorile notelor și numele elevilor sunt corect citite și media este calculată corect în programul dvs.

- Date de intrare:
  - Andrei 7.2 8.3 8.9
  - Marius 5.67 8.13 9.43
- Date de iesire:
  - Nume - Andrei, Medie: 8.13
  - Nume - Marius, Medie: 7.743

## Exercitii cu structuri pb-info

1. Se dau 4 numere naturale a b c d, reprezentând fracțiile a/b și c/d. Determinați și afișați cea mai mare dintre cele două fracții.

- Sursa: <https://www.pbinfo.ro/probleme/923/fractiimax>

- Date de intrare: 7 14 24 18

- Date de iesire: 4 3

- Solutie:

```
#include <iostream>
#include <cstring>

using namespace std;
int gcd(int a, int b);

struct fractie {
    int numarator;
    int numitor;
};

int main()
{
    int a, b, c, d;
    cin >> a >> b >> c >> d;
    struct fractie f1, f2;
    f1.numarator = a;
    f1.numitor = b;
    f2.numarator = c;
    f2.numitor = d;

    int rezultat1 = f1.numarator/f1.numitor;
    int rezultat2 = f2.numarator/f2.numitor;
    if (rezultat1 > rezultat2) {
        int cmmdc = gcd(f1.numarator, f1.numitor);
        f1.numarator = f1.numarator / cmmdc;
        f1.numitor = f1.numitor / cmmdc;
        cout << f1.numarator << " " << f1.numitor;
    } else {
        int cmmdc = gcd(f2.numarator, f2.numitor);
        f2.numarator = f2.numarator / cmmdc;
        f2.numitor = f2.numitor / cmmdc;
        cout << f2.numarator << " " << f2.numitor;
    }

    return 0;
}

int gcd(int a, int b) {
    if (b == 0) {
        return a;
    }
}
```

```
    }  
    int rest = a % b;  
    return gcd(b, rest);  
}
```

2. Se dau coordonatele carteziene a  $n$  puncte în plan. Să se determine distanța maximă dintre un punct dat și originea sistemului de coordonate și numărul de puncte situate la acea distanță față de origine. Programul citește de la tastatură numărul  $n$ , iar apoi  $n$  perechi de puncte  $x$   $y$ , reprezentând coordonatele punctelor.

- Sursa: <https://www.pbinfo.ro/probleme/922/puncte>

- Date de intrare:

```
7  
2 -1  
-2 4  
2 3  
3 -1  
-2 -4  
2 -2  
-4 2
```

- Date de iesire: 4.472 3

- Solutie:

```
#include <iostream>  
#include <cmath>  
  
using namespace std;  
  
struct Point{  
    int x;  
    int y;  
};  
  
int main() {  
  
    struct Point points[100];  
    int n;  
    cin >> n;  
    for(int i = 0; i < n; i++) {  
        cin >> points[i].x;  
        cin >> points[i].y;  
    }  
  
    float distance = -1;
```

```

        int numberOfPoints = 0;

        for(int i =0; i < n; i++) {
            float currentDistance = sqrt((points[i].x -0)*
(points[i].x -0) + (points[i].y -0)*(points[i].y -0));
            if(currentDistance > distance) {
                distance = currentDistance;
                numberOfPoints = 1;
            } else if(currentDistance == distance) {
                numberOfPoints++;
            }
        }

        cout << distance << " " << numberOfPoints;
        return 0;
    }

```

3. Se dau datele de naștere a  $n$  persoane, numerotate de la 1 la  $n$ , în forma an luna zi. Să se determine numărul de ordine al celei mai tinere și al celei mai în vârstă persoană dintre cele date. Programul citește de la tastatură numărul  $n$ , iar apoi  $n$  triplete de numere naturale a l z, reprezentând anul, luna și ziua de naștere a fiecărei persoane. Programul va afișa pe ecran două numere p q, reprezentând numărul de ordine al celei mai tinere, respectiv al celei mai în vârstă persoană.

- Sursa: <https://www.pbinfo.ro/probleme/1013/aniversari>
- Date de intrare:

```

5
1998 5 26
1987 12 18
1987 9 25
2015 1 16
2015 1 8

```

- Date de iesire: 4 3
- Solutie:

## Exercitii cu structuri extrase din variante de bacalaureat

1. Se consideră declararea alăturată, în care variabila FIG memorează coordonatele, în planul xOy, ale centrului unui cerc, precum și lungimea razei acestuia. O expresie C/C++ care are valoarea 1 dacă și numai dacă centrul cercului se află pe prima bisectoare a sistemului de coordonate al planului xOy este:

```
struct cerc {  
    float x, y;  
    float raza;  
} FIG;
```

- Sursa: BAC 2010 Iunie, Subiectul II, exercitiul 1.
- Solutie:

- Variante de raspuns:
  - a. centru(x)==centru(y)
  - b. FIG.cerc.x==FIG.cerc.y
  - c. FIG(x)==FIG(y)
  - d. FIG.x==FIG.y

2. Variabilele e1 și e2, declarate alăturat, memorează numele și data nașterii pentru câte un elev. Știind că cei doi elevi sunt născuți în ani diferiți, scrieti secventa de instructiuni care afișează pe ecran numele elevului de vârstă mai mare

```
struct data_n {  
    int an, luna, zi;  
};  
struct elev{  
    char nume[20];  
    data_n d;  
} e1, e2;
```

- Sursa: BAC 2010 August, Subiectul II - exercitiul 4.
- Solutie:

3. Structura cu numele CARTE permite memorarea unui număr real, reprezentând valoarea exprimată în euro a unei cărți, a unui număr întreg, reprezentând numărul de pagini ale cărții, și a unui șir de caractere reprezentând titlul cărții, format din cel mult 50 de caractere. Scrieti în limbajul C/C++ o definiție a structurii mentionate și declarati o variabilă x de acest tip. Denumiti sugestiv câmpurile acestei structuri.

- Sursa: BAC 2011 August, Subiectul II - Exercițiul 3.
- Solutie:



4. Se consideră declararea alăturată, în care variabilele t1 și t2 memorează lungimile laturilor a două triunghiuri. Scrieți o expresie C/C++ care are valoarea 1 dacă și numai dacă triunghiul corespunzător variabilei t1 are perimetrul strict mai mare decât perimetrul triunghiului corespunzător variabilei t2.

```
struct triunghi {  
    int a,b,c;  
} t1, t2;
```

- **Sursa:** BAC 2012 August, Subiectul II - Exercițiul 3
- **Soluție:**

5. Se consideră declararea alăturată. Scrieți o secvență de instrucțiuni în urma executării căreia câmpul paritate al variabilei x să memoreze litera p, dacă numărul memorat în câmpul v al acesteia este par, sau litera i dacă numărul memorat în câmpul v al acesteia este impar.

```
struct numar  
{ int v;  
  char paritate;  
} x;
```

- **Sursa:** BAC 2013 Iunie, Subiectul II - Exercițiul 3
- **Soluție:**

6. În declarațiile alăturate: • variabila p memorează coordonatele unui punct în sistemul de coordonate xOy; • variabila d memorează în câmpurile A și B coordonatele, în același sistem de coordonate, ale vârfurilor din stânga – sus, respectiv din dreapta – jos ale unui dreptunghi cu laturile paralele cu axele sistemului de coordonate. Scrieți o expresie C/C++ care să aibă valoarea 1 dacă punctul corespunzător variabilei p se află în interiorul dreptunghiului corespunzător variabilei d (dar nu pe laturile acestuia) sau valoarea 0 în caz contrar.

```
struct punct {  
    int x,y;  
} p;  
  
struct dreptunghi {  
    punct A, B;  
} d;
```

- **Sursa:** BAC 2013 August, Subiectul II - Exercițiul 4

- **Soluție**

7. Se consideră declararea alăturată. Scrieți o secvență de instrucțiuni în urma executării căreia să se afișeze pe ecran mesajul **acceptat**, dacă momentul de timp corespunzător variabilei **start** precede momentul de timp din aceeași oră, corespunzător variabilei **stop**, sau mesajul **respins** în caz contrar.

```
struct timp
{   int minut;
    int secunda;
} start, stop;
```

- **Sursa:** BAC 2014 Iunie, Subiectul II - Exercițiul 3

- **Soluție:**

8. Variabila c, declarată alăturat, memorează titlul și prețul unei cărți. Expresia C/C++ a cărei valoare reprezintă prețul cărții respective majorat cu 50% este:

```
struct carte
{   char titlu[21];
    float pret;
}c;
```

- **Sursa:** BAC 2015 Iunie, Subiectul II - Exercițiul 1

- **Soluție:**

- **Variante:**

- a.  $c.pret * 3/2$
- b.  $pret.c * 3/2$
- c.  $c(pret) * 3/2$
- d.  $pret[c] * 3/2$

9. Variabila z, declarată alăturat, memorează partea reală și partea imaginară a unui număr complex. Expresia care are valoarea egală cu pătratul modulului acestui număr (suma dintre pătratul părții reale și pătratul părții imaginare) este:

```
struct complex
{ float re;
  float im;
}z;
```

- **Sursa:** BAC 2015 August:

- **Solutie:**

- **Variante:**

- a. `complex(re)*complex(re)+complex(im)*complex(im)`
- b. `complex.z(re)*complex.z(re)+complex.z(im)*complex.z(im)`
- c. `re.zre.z+im.zim.z`
- d. `z.rez.re+z.imz.im`

10. În declarațiile alăturate, variabilele A și B memorează coordonatele câte unui punct în sistemul de coordonate xOy. Indicați expresia care are valoarea 1 dacă și numai dacă cele două puncte coincid.

```
struct punct
{ int x,y;
}A,B;
```

- **Sursa:** BAC 2015 Speciala, Subiectul II - Exercițiul 1.

- **Solutie:**

- **Variante:**

- a. ☒ `&& A[y]==B[y]`
- b. `A.x == B.x && A.y==B.y`
- c. `x[A]==x[B] && y[A]==y[B]`
- d. `x.A==x.B && y.A==y.B`

## Exercitii de antrenament:

- Exerciții de sintaxa similare cu cele de la bacalaureat:
  - <https://www.pbinfo.ro/?pagina=itemi-evaluare-lista&disciplina=0&clasa=10&tag=89>
  - Rezolva cate poti din ele, incearca sa faci cat mai multe
  - Sesiunea viitoare o sa o dedicam in totalitate exercitiilor cu structuri, posibil pe urmatoarele 2, 3 chiar deoarece este un capitol destul de simplu si poti ajunge sa il stapanesti super ok, intr-un timp scurt.

