

TIPURI DE DATE DEFINITE DE UTILIZATOR

Limbajele de programare de nivel înalt oferă utilizatorului facilități de a prelucra atât datele singulare (izolate), cât și pe cele grupate (tablourile). Datele predefinite și tablourile nu sunt însă suficiente. Informația prelucrată în programe este organizată, în general în ansambluri de date, de diferite tipuri. Pentru a putea descrie aceste ansambluri (structuri) de date, limbajele de programare de nivel înalt permit programatorului să-și definească *propriile tipuri de date*.

Aceste tipuri de date sunt structuri de date neomogene care ne permit următoarele gruparea unor obiecte (date) de tipuri diferite, referite printr-un nume comun.

Exemplificând, dacă dorim să prelucrăm date referitoare la mai mulți elevi, pentru fiecare elev trebuie să cunoaștem:

- *Numele* – *char[20]*
- *Prenumele* – *char[20]*
- *Nota matematică* - *float*
- *Nota informatică* – *float*

În C++ există un tip de date, numit **struct**, care permite ca fiecărui elev să-i corespundă o singură înregistrare.

Forma generală este:

```
struct [nume structura]
{
    [<tip> <nume variabila [, nume variabila, .....]>];
    [<tip> <nume variabila [, nume variabila, .....]>];
    ...
} [lista de variabile];
```

Structura o denumim **elev** și este:

```
struct elev
{ char nume[20], prenume[20];
  float nota_mate, nota_info;
  int varsta;
};
```

Există două posibilități de declarare a variabilelor care alcătuiesc structura:

1. Scriind la sfârșit numele variabilelor:

```
struct elev
{ char nume[20], prenume[20];
  float nota_mate, nota_info;
  int varsta;
} inr1, inr2;
```

2. Declarând variabilele așa cum suntem obișnuiți:

```
elev inr1, inr2;
```

Definiția structurii poate fi făcută:

- În cadrul funcției main()
- Înaintea funcției main() (caz recomandat)

Pentru accesul la câmpurile unei variabile de tip struct se folosește operatorul de selecție directă, notat cu '.', operator cu prioritate maximă.

Dacă *inr* este o variabilă de tipul elev. Atunci:

- *inr.nume* – reprezintă șirul nume al variabilei *inr*;
- *inr.nume[0]* - reprezintă primul caracter al șirului nume;
- *inr.nota_mate* – reprezintă câmpul *nota_mate* al variabilei *inr*.

Între două variabile de același tip struct se poate folosi atribuirea.

Dacă *inr1*, *inr2* sunt două variabile de tip elev, prin atribuirea *inr1=inr2*, variabila *inr1* ia aceeași valoare ca variabila *inr2*.

O astfel de atribuire se mai numește copiere bit cu bit.

Înregistrările imbricate

Putem avea un tip structurat ce conține în interiorul său un alt tip structurat.

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

Dacă *inr* este o variabilă de tipul elev. Atunci accesarea elementelor situate în interiorul structurii se face astfel:

- *inr.situatie.clasa* – se accesează câmpul *clasa* al substructurii
- *inr.situatie.note[0]* – se accesează prima notă a vectorului inclus în substructură

Dacă avem

```
struct elev1
{ char nume[20], prenume[20];
  struct
  { int clasa;
    float note[20];
  } situatie_1, situatie_2;
  int varsta;
};
```

Tipul structurat *elev1*, subordonează structurile *situatie_1* și *situatie_2*.

Dacă una sau mai multe structuri sunt incluse una în alta, acestea se numesc structuri **imbricate**.

Cele două structuri sunt echivalente:

```
struct elev1
{
    char nume[20],
    prenume[20];

    struct
    {
        int clasa;

        float note[20];

    } situatie_1, situatie_2;

    int varsta;
};
```

```
struct elev1
{
    char nume[20], prenume[20];

    struct
    {
        int clasa;

        float note[20];

    } situatie_1;

    struct
    {
        int clasa;

        float note[20];

    } situatie_2;

    int varsta;
};
```

Exemplu: O firma are un numar de angajati, pentru fiecare pastrandu-se urmatoarele date:

- nume
- prenume
- nr copii
- locul nasterii

Folosind structuri introduceti datele apoi afisati datele introduce.

Rezolvare:

```
#include<iostream.h>
#include<conio.h>
struct persoana{
    char nume[20], prenume[20];
    int nr_copii;
    double salariu;
    char loc_nastere[20];
}angajati[10];

void main()
{
    int n;
    cout<<"Nr. angajati=";
    cin>>n;
    for(int i=1;i<=n;i++)
```

```

    {
        cout<<"Introduceti datele pentru angajatul "<<i<<"\n";
        cout<<"Numele :"; cin>>angajati[i].nume;
        cout<<"Prenumele :"; cin>>angajati[i].prenume;
        cout<<"Nr. copii:"; cin>> angajati[i].nr_copii;
        cout<<"Locul nasterii:"; cin>> angajati[i].loc_nastere;
    }
    cout<<"Datele introduse au fost:"<<"\n";
    for(int i=1;i<=n;i++)
    {
        cout<<"Numele "<<i<<angajati[i].nume<<"\n";
        cout<<"Prenumele: "<<i<<angajati[i].prenume<<"\n";
        cout<<"Nr. copii: "<<angajati[i].nr_copii<<"\n";
        cout<<"Locul nasterii: "<<angajati[i].loc_nastere;
    }
    getch();
}

```

Laborator:

Se introduce de la tastatura n elevi, memorandu-se pentru fiecare din acestia urmatoarele:

- nume
- prenume
- clasa
- nota romana
- nota mate
- nota info.

Sa se asiseze urmatoarele:

- a. numele tuturor elevilor cititi, clasa din care fac parte si media
- b. sa se afiseze elevii din clasa X (X fiind una din clasele citite)