STRUCTURI DE DATE NEOMOGENE

TIPUL DE DATE ÎNREGISTRARE

Clasa a XI-a MI

Colegiul Național Pedagogic „Regina Maria” Deva - Prof. Berceanu Camelia Florentina

TIPUL DE DATE ÎNREGISTRARE

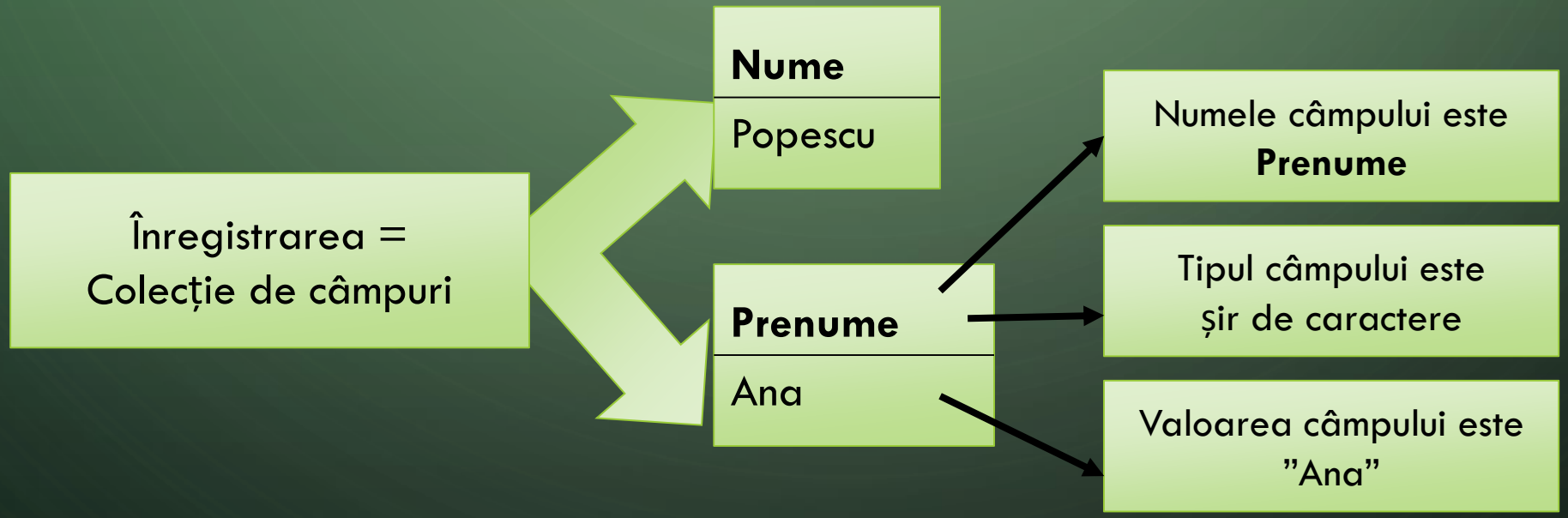
- Înregistrarea este structura de date formată dintr-un ansamblu de date neomogene (date elementare sau structuri de date) între care există o legătură de conținut. Elementele structurii se numesc câmpuri și pot fi identificate după nume.
- Înregistrarea se folosește atunci când dorim să păstrăm și să prelucrăm informații legate între ele din punct de vedere al conținutului, informații care sunt memorate în variabile de memorie, având tipuri diferite. Astfel, pentru a obține informații despre un obiect, întreaga colecție de date trebuie prelucrată împreună.

CÂMPUL

- Câmpul este reprezentarea unui atribut din lista de atribute care descriu obiectul. El conține o dată elementară sau o structură de date care are legătură logică cu datele din celelalte câmpuri ale înregistrării, cu care formează împreună o entitate de informație.
- Fiecare câmp se identifică în listă printr-un nume și este caracterizat de tipul datei și de valoarea datei.
- Accesul la elementele structurii se face prin numele câmpului.

IMPLEMENTAREA UNEI ÎNREGISTRĂRI

- Înregistrarea este o structură de date secvențială sau liniară care grupează date corelate, care sunt așezate în locații succesive în memorie.
- Înregistrarea este o zonă continuă de memorie căreia i se atribuie un nume și care permite memorarea mai multor date de tipuri diferite ce pot fi tratate ca un tot unitar sau ca date elementare independente.



IMPLEMENTAREA ÎNREGISTRĂRII ÎN LIMBAJUL C++

- În limbajul C++, pentru a putea crea și manipula o structură de date de tip înregistrare, trebuie definit tipul de date numit structură, având următoarea formă generală:

```
struct [< nume structură >]
        {   <tip dată 1> <nume v11>, <nume v12>, …, <nume v1n>;
            <tip dată 2> <nume v21>, <nume v22>, …, <nume v2n>;
            ……
            <tip dată m> <nume vm1>, <nume vm2>, …, <nume vmn>;      } ;
```

- Unde < nume structură > este numele tipului de dată care reunește mai multe variabile de memorie de tipul <tip dată i>, identificate prin

<nume vi1>, <nume vi2>, …, <nume vin>; Numele structurii este opțional.

OBSERVAȚII

- Declararea unei structuri de termină cu caracterul ; deoarece este o instrucțiune.
- Se pot folosi trei variante de declarare a tipului înregistrare:

V1: struct < nume structură >
{< descriere câmpuri înregistrare>} ;
< nume structură > < nume variabilă > ;

```
struct elev
{
    char nume[21], prenume[21];
    int varsta;
};
elev x;
```

V2: struct < nume structură >
{< descriere câmpuri înregistrare>} < nume variabilă > ;

```
struct elev
{
    char nume[21], prenume[21];
    int varsta;
} x;
```

V3: struct
{< descriere câmpuri înregistrare>} < nume variabilă >;

```
struct
{
    char nume[21], prenume[21];
    int varsta;
} x;
```

ACCESUL LA CÂMPURILE ÎNREGISTRĂRII

- Pentru a avea acces la un câmp al unei înregistrări se folosește **operatorul punct „ . ”**

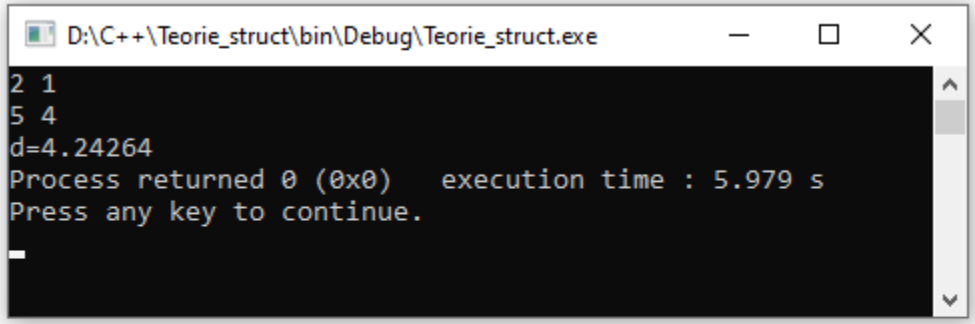
  <nume_variabilă_înregistrare>.<nume_câmp>

- Operatorul punct este operator de selecție a câmpului unei structuri și leagă numele structurii de numele câmpului. Rezultatul furnizat de expresie este valoarea câmpului selectat. Punctul este operator binar și are prioritate maximă.

APLICAȚII PRACTICE

- Se citesc de la tastatură coordonatele a două puncte din plan – A și B. Să se afișeze distanța dintre cele două puncte.

```
#include <iostream>
#include <cmath>
using namespace std;
struct punct
{
    int x,y;
} A,B;
int main()
{
    cin>>A.x>>A.y;
    cin>>B.x>>B.y;
    cout<<"d="<<sqrt(pow(A.x-B.x,2)+pow(A.y-B.y,2));
    return 0;
}
```



- Se citesc de la tastatură numărătorul și numitorul a două fracții pozitive. Să se calculeze și să se afișeze suma și produsul celor două fracții.