

## 22-MA'RUZA. STRUKTURALAR MASSIVI. STRUKTURAGA KO'RSATKICH. DINAMIK STRUKTURALAR

### 1. Strukturalar massivi. Strukturaga ko'rsatkich

Har qanday o'rnatilgan ma'lumotlar turidan (int, char, ...) massivlar yaratishimiz mumkin bo'lgani kabi, strukturalar massivlarini yaratishimiz mumkin.

Talabalar haqidagi ma'lumotlarni saqlovchi quyidagicha jadval mavjud bo'lsin.

T/r	Talabaning F.I.SH	Tug'ilgan yili	Tug'ilgan joyi	Yo'nalish	Guruh
1	Asrorov Sunnat	2000	Samarqand	Dasturiy injiniring	109
2	Ziyodov Ne'mat	2001	Samarqand	Amaliy matematika	110
3	Suhrobova Nigina	2000	Buxoro	Amaliy matematika	102
4	Amonova Zuhra	2002	Toshkent	Amaliy matematika	103
5	Norboyev Zokir	2003	Farg'ona	Amaliy matematika	102
6	Sheraliyev Feruz	2001	Samarqand	Amaliy matematika	103

Yuqoridagi jadvalni struktura ko'rinishida quyidagicha e'lon qilamiz:

```
#include<iostream>
using namespace std;
struct Student
{
    char fullName[64]; //F.I.SH
    int year; // Tug'ilgan yili
    char placeOfBirth[20]; //Tug'ilgan joyi
    char direction[25]; //Yo'nalish
    int group; //Guruhi
};

void showData(const Student Obj[], int countStudent)
{
```

```

        cout << "№\t" << "F.I.SH\t" << "Tug'ilgan yili\t" << "Tug'ilgan joyi\t"
<< "Yo'nalish\t" << "Guruh" << endl;
        cout <<
"=====
===== " << endl;
        for (int i = 0; i < countStudent; i++)
        {
            cout << i + 1 << '\t' << Obj[i].fullName << '\t' << Obj[i].year
<< '\t' << Obj[i].placeOfBirth << '\t' << Obj[i].direction << '\t'
<< Obj[i].group << endl;
        }
    }

int main()
{
    const int countStudent = 6;
    Student Talaba[countStudent] = {};

    for (int i = 0; i < countStudent; i++)
    {
        cout << "Yo'nalishi: ";
        cin.getline(Talaba[i].direction, 20);
        cout << "Ismi: ";
        cin.getline(Talaba[i].fullName, 32);
        cout << "Tug'ilgan yili: ";
        cin >> Talaba[i].year;
        cout << "Guruh: ";
        cin >> Talaba[i].group;
        cout << "Tug'ilgan joyi: ";
        cin.getline(Talaba[i].placeOfBirth, 15);

        cin.get(); //
        cout << endl;
    }
    showData(Talaba, countStudent);

    return 0;
}

```

Student strukturalari uchun massiv 30-satrdan e'lon qilinadi va shu satrda ushbu massivdagi barcha obyekt elementlari nol-initsializatsiya qilinadi. Ko'rib turganingizdek, o'rnatilgan tip o'rniga biz struktura tavsiflovchisini yozganimizdan tashqari, ushbu e'lon qilishda biz uchun yangi narsa yo'q. Massivga nom berildi, elementlar (obyektlar) sonini ko'rsatdik, nol bilan initsializatsiya qilindi. Shunday qilib, strukturalar massivini yaratdik.

Muayyan struktura obyektining elementiga qanday murojaat qilish kerak? Nuqta orqali odatiy murojaat qilish endi ishlamaydi:

**cin>>Talaba.year; //Bu xatolik**

Bu holatda, 6 ta obyektidan qaysi biri to'ldirish kerakligi aniq belgilanmagan. Har bir obyekt endi strukturali massiv yacheykasi bo'lganligi sababli, uning elementlariga murojaat qilish uchun avval siz obyektning indeksini to'rtburchak qavsda ko'rsatishingiz kerak. Shundan keyingina elementning a'zosi va nomi keltiriladi:

**cin >> Talaba[i].year //Bu murojaat to'g'ri**

Qolgan qismi esa oddiy bir o'lchamli massivning qiymatlarini to'ldirish jarayoni kabi bajariladi.

**Strukturalarga ko'rsatkich.** Ko'rsatkichlarni strukturalarda ham boshqa turdagi obyektlarda ham aniqlash mumkin. Masalan, person strukturasiga ko'rsatkich:

**struct shaxs \*p;**

Strukturalarga ko'rsatkichlar nomlanmagan struktura turlari uchun ham yaratilishi mumkin:

```
struct
{
    int age;
    char name[20];
} *p1, *p2;
```

Qiymat sifatida bunday ko'rsatkichga bir xil turdagi strukturaning obyekti beriladi:

**struct shaxs Anvar= {31, "Anvar"};**

```
struct shaxs *p_Anvar= &Anvar;
```

Strukturaga ko'rsatkich yordamida uning a'zolariga kirishingiz mumkin. Buning ikki yo'li mavjud. Birinchi usul - ajratish amalidan foydalanish:

```
(*strukturaga_ko'rsatkich).element_nomi
```

Ikkinchi usul -> amalidan (yo'nalish amali) foydalanishni o'z ichiga oladi:

```
strukturaga_ko'rsatkich->element_nomi
```

Struktura elementlariga murojaat qilish uchun ushbu ikkala usuldan ham foydalanish mumkin:

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

```
struct shaxs  
{  
    int yosh;  
    char ism[20];  
};
```

```
int main()  
{  
    struct shaxs Anvar = {31, "Anvar"};  
    struct shaxs * p_anvar= &Anvar;
```

```
    char * ism = p_anvar->ism;  
    int yosh = (*p_anvar).yosh;
```

```
    cout<<"Ism: "<<ism<<endl;  
    cout<<"Yosh: "<<yosh<<endl;
```

```
    // Strukturadagi yosh maydonini o'zgartirish  
    p_anvar->yosh = 32;  
    cout<<Anvar.ism<<endl;  
    cout<<Anvar.yosh;  
    return 0;
```

}

Bu p\_anvar ko'rsatkichini Anvar o'zgaruvchisiga belgilaydi va ko'rsatkich yordamida struktura elementlarining qiymatlarini olishimiz yoki o'zgartirishimiz mumkin.

## **2. Dinamik strukturalar**