

21-22-AMALIY MASHG'ULOT. STRUKTURALARNI E'LON QILISH VA INITSIALIZATSIYALASH. STRUKTURALARDAN FOYDALANIB DASTURLAR TUZISH

STRUKTURALAR MASSIVI. STRUKTURAGA KO'RSATKICH. DINAMIK STRUKTURALARDAN FOYDALANIB DASTURLAR TUZISH

C++ tilida bir yoki har xil turdagi berilganlarni jamlanmasi **struktura** deb nomlanadi.

Struktura e'loni:

```
struct <struktura nomi>
{
    <tur1> <nom1>;
    <tur2> <nom2>;
    <tur3> <nom3>;
    .....
    <turn> <nomn>;
}
```

Masalan:

```
struct Klub
{
    int Tartib;
    char nom;
    char shahar;
    unsigned short yil;
    unsigned short pozitsiyal
};
```

Programmada strukturalardan foydalanish, shu turdagi o'zgaruvchilar e'lon qilish va ularni qayta ishlash orqali amalga oshiriladi.

Masalan:

Klub klublar;

Struktura maydoniga murojaat qilish '.' (nuqta) orqali amalga oshiriladi.

Masalan:

```
Klub klublar;

klublar.Tartib = 1;
strcpy_s(klublar.nom, "Chelsi");
strcpy_s(klublar.shahar, "London");
klublar.yil = 1923;
klublar.pozitsiya = 1;
```

struktura turidagi obyektning xotiradan qancha joy egallaganini bilish uchun sizeof funksiyasidan foydalanamiz:

```
int i = sizeof(Klub);
```

Ayrim hollarda struktura maydonlari o'lchamini bitlarda aniqlash orqali egallanadigan xotirani kamaytirish mumkin. Buning uchun struktura maydoni quyidagicha e'lon qilinadi.

<maydon nomi>:<o'zgarmas ifoda>

Masalan:

Agar

```
struct Sana_Vaqt
{
    unsigned short Yil;
    unsigned short Oy;
    unsigned short Kun;
    unsigned short Soat;
    unsigned short Minut;
    unsigned short Sekund;
}
```

Ko'rinishida bersak, 6 maydon har bir ikki baytdan jami 12 bayt bo'ladi. Agar ushbu strukturani quyidagicha e'lon xotiradan 5 bayt joy egallaydi:

```
struct Sana_Vaqt
{
    unsigned short Yil : 7;
    unsigned short Oy : 4;
    unsigned short Kun : 7;
    unsigned short Soat : 7;
    unsigned short Minut : 7;
    unsigned short Sekund : 7;
}
```

Struktura funksiya argumenti sifatida

Buning uchun funksiya prototipida struktura nomi ko'rsatilishi kerak.

Masalan:

```
Void Klublar_haqida(Klub);
```

Funksiyaga strukturani parametr sifatida uzatishga misol sifatida quyidagini keltirilaylik:

```
#include <iostream>
#include <string>
using namespace std;
struct Klub
{
    int Tartib;
```

```

        char nom[10];
        char shahar[10];
        unsigned short yil;
        unsigned short pozitsiya;
    };
    void Klublar_haqida(Klub K)
    {
        cout << "Klubning shahri:" << K.shahar;
    }
    int main()
    {
        Klub klublar;

        klublar.Tartib = 1;
        strcpy_s(klublar.nom, "Chelsi");
        strcpy_s(klublar.shahar, "London");
        Klublar_haqida(klublar);
        klublar.yil = 1923;
        klublar.pozitsiya = 1;

        return 0;
    }

```

Strukturalar massivi

Strukturalar massivini e'lon qilish xuddi standart massivlarni e'lon qilishdek, farqi massiv turi o'rnida foydalanuvchi tomonidan aniqlangan struktura turning nomi yoziladi.

Masalan:

```

#include <iostream>
#include <string>
using namespace std;
struct Klub
{
    int Tartib;
    string nom;
    char shahar[10];
    unsigned short yil;
    unsigned short pozitsiya;
};
void Klublar_haqida(Klub K)
{
    cout << "Klubning shahri:" << K.shahar;
}
int main()
{
    Klub klublar[6];
    for (int i = 1; i <= 6; i++)
    {
        cout << "T="; cin >> klublar[i].Tartib;
        cout << "N="; cin >> klublar[i].nom;
        cout << "Y="; cin >> klublar[i].yil;
    }
}

```

```

    }
    cout << "Klublar va ular haqida ma'lumot" << endl;
    for (int i = 1; i <= 6; i++)
    {
        cout << klublar[i].Tartib << " ";
        cout << klublar[i].nom << " ";
        cout << klublar[i].yil << endl;
    }
}

```

Strukturalarga ko'rsatkich

Struktura elementlariga ko'rsatkichlar orqali murojaat qilish mumkin. Buning uchun strukturaga ko'rsatkich o'zgaruvchisi e'lon qilinishi kerak. Masalan Klub strukturasi ko'rsatkich quyidagicha e'lon qilinadi

Klub *k_klub;

Ko'rsatkich orqali aniqlangan struktura elementlariga murojaat nuqta bilan emas, balki “->” vositasida amalga oshiriladi.

k_klub->nom

Dinamik strukturalar.

Birlashmalar va ular ustida amallar.

Birlashma xotiraning bitta sohasida (bitt adres bo'yicha) har xil turdagi bir nechta aberilganlarni saqlash imkonii beradi. Birlashma e'lonida *union* kalit so'zi, undan keyin identifikator va blok ichida har xil turdagi elementlar blokidan iborat bo'ladi.

Masalan:

```

union Birlashma
{
    int n;
    unsigned N;
    char Satr[10];
};

```

Birlashma maydonlariga xuddi struktura lementlriga murojaat qilgandek murojaat qilinadi. Birlashma elementi sifatida stukturalar kelishi mumkin va ular odatda berilganni “bo'laklarga” ajratish yoki “bo'laklardan” yaxlit berilganni hosil qilish uchun xizmat qiladi. Misol uchun so'zni baytlarga, baytlarni tetradalarga ajratish va qaytadan birlashtirish mumkin.

Foydalanuvchi tomonidan aniqlangan berilganlar turi.

Foydalanuvchi tomonidan aniqlanadigan tur typedef kalit so'zi bilan boshlanadai, undan keyin mavjud tur ko'rsatiladiva identifikator yoziladi.

Oxirida yozilgan identifikator – yangi yaratilgan turning nomi hisoblanadi. Masalan,

```
typedef unsigned int tur;
```

ifodasi tur deb nomlanuvchi yanggi turni yaratadi va o'z mazmunga ko'ra unsigned char turi bilan bilan ekvivalent bo'ladi.

Massiv ko'rinishidagi foydalanuvchi tomonidan aniqlanuvchi tur e'loni quyidagicha bo'ladi:

```
typedef char Ism[30];  
Ism ism;
```

Ism turidagi ism o'zgaruvchisi e'loni – bu 30 belgidan iborat massiv e'lonidir.

Odatda yechilayotgan masalaning predmet sohasi terminlarida ishlash uchun qayta nomlanadi. Natijada murakka tuzilishga ega bo'lgan va zarur xususiyatlarni o'zida jamlagan yanggi turlarni yaratishfa muvofiq bo'linadi. Masalan kompleks son haqida ma'lumotlarni o'z ichiga oluvchi Complex turi quyidagicha aniqlanadi:

```
typedef struct  
{  
    double re;  
    double im;  
} Complex;
```

Endi kompleks son e'lonini

Complex KSon;

Yozish mumkin va uning maydonlarigamurojaat qilish mumkin:

```
KSon.re = 5.64;  
KSon.im = 2.3;
```