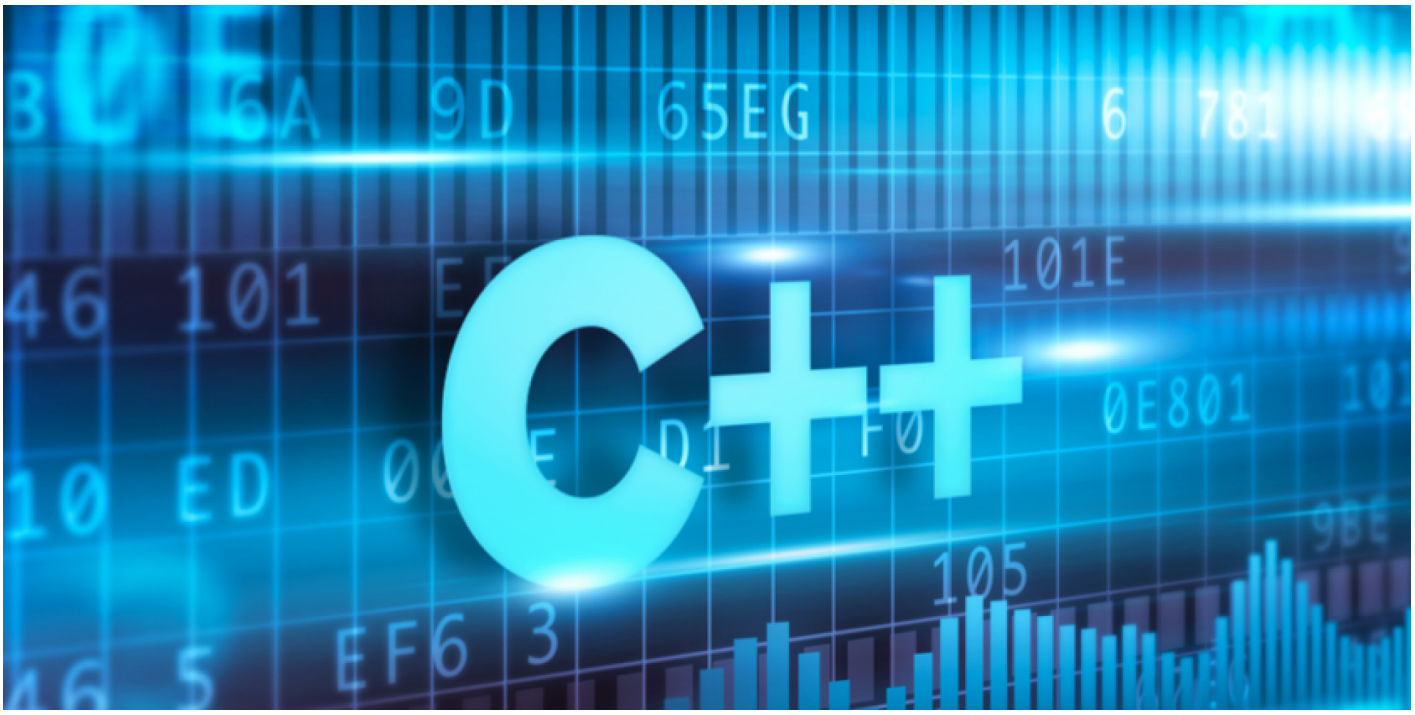
C++tilida sinflar



Sinflarni eng soda holda qo’yidagicha tasvirlash mumkin:

Sinf-kaliti Sinf-soni {komponentalar ruyhati}

Sinf komponentalari sodda holda tiplangan ma'lumotlar va funktsiyalardan iborat bo’ladi. Figurali qavslarga olingan komponentalar ro’yhati sinf tanasi deb ataladi. Sinfga tegishli funktsiyalar komponenta-funktsiyalar yoki sinf funktsiyalari deb ataladi. Sinf kaliti sifatida Struct hizmatchi so’zi ishlatilishi mumkin. Masalan qo’yidagi konstruktsiya kompleks son sinfini kiritadi.

```
Struct complex 1
{ double real;
double imag;
void define (double re=0.0, double im=0.0)
{ real=re; imag=im;}
void display (void)
{cout<=”real=”<<real;
cout<=”imag=”<<imag;
```

```
}
};
```

Strukturadan bu sinfning farqi shuki komponenta ma'lumotlardan (real, imag) tashqari ikkita komponenta funktsiya (define() va display ()) kiritilgan. Bu kiritilgan sinf o’zgaruvchilar tipi deb qaralishi mumkin. Bu tiplar yordamida konkret ob'ektlarni qo’yidagicha tasvirlash mumkin:

Misol uchun:

```
Complex x,y;
Complex dim[8];
Complex *p=1x;
```

Sinfga tegishli ob'ektlar qo’yidagicha tasvirlanadi;

Sinf-nomi . ob'ekt-nomi

Dasturda ob'ekt komponentasiga quyidagicha murojaat qilish mumkin:

Sinf-nomi.ob'ekt-nomi :: komponenta-nomi yoki soddaroq holda ob'ekt-nomi. Element-nomi

Misol uchun:

```
x!=real=1.24;
x!=imag=0.0;
dim[3]. Real=0.25;
dim[3]. Imag=0.0;
```

Sinfga tegishli funktsiyalarga qo’yidagicha murojaat qilinadi:
funktsiya-nomi.ob'ekt-nomi;
Misol uchun:
X. define.(Bu holda real=0.9 va imag=0.0)
X. define.(Bu holda kompleks son 4.3+i*20.0)
Display funktsiyasi ekranda kompleks son qiymatlarini tasvirlaydi. Sinfga tegishli ob'ektga ko’rsatkich orqali komponentalarga quyidagicha murojat qilinadi:

Ob'ektga-ko’rsatkich>element-nomi

Yuqorida ko’rsatilgan P ko’rsatkich orqali H ob'ekt elementlariga qo’yidagicha qiymat berish mumkin:
P>real=2.3
P>imag=6.1
Huddi shu shaklda sinfga tegishli funktsiyalarga murojat qilinadi:
P>display;
P>define(2.3, 5.4);

Kompanenta o’zgaruvchilar va kompanenta funktsiyalar.

Sinf kompanenta o’zgaruvchilari sifatida o’zgaruvchilar , massivlar, ko’rsatkichlar
ishlatilishi mumkin . Elementlar ta'riflanganda initsializatsiya qilish mumkin emas . Buning sababi shuki sinf uchun hotiradan joy ajratilmaydi. Kompanenta elementlariga kompanenta funktsiyalar orqali murojat qilinganda faqat nomlari ishlatiladi. Sinfdan tashqarida sinf elementlariga emas ob'ekt elementlariga murojaat qilish mumkin. Bu murojaat ikki hil bo’lishi mumkindir.
Ob'ekt- nomi . Element - nomi.
Ob'ktga – korsatgich – element nomi.
Sinf elementlari sinfga tegishli funktsiyalarida ishlatilishidan oldin ta'riflangan bo’lishi shart emas. Huddi shunday bir funktsiyadan hali ta'rifi berilmagan ikkinchi funktsiyaga murojaat qilish mumkin.
Komponentalarga murojaat huquqlari. Komponentalarga murojaat huquqi murojaat spetsifikatorlari yordamida boshqariladi. Bu spetsifikatorlar :

Protected – himoyalangan;
Private – hususiy;
Public – umumiy;
Himoyalangan kompanentalardan sinflar ierarhiyasi qurilganda foydalaniladi. Oddiy holda Protected spetsifikatori Private spetsifikatoriga ekvivalentdir. Umumiy ya'ni Public tipidagi komponentalarga dasturning ihtiyoriy joyida murojaat qilinishi mumkin. Hususiy ya'ni Private tipidagi

komponentalarga sinf tashqarisidan murojaat qilish mumkin emas. Agar sinflar Struct hizmatchi so’zi bilan kiritilgan bo’lsa, uning hamma komponentalari umumiy Public bo’ladi, lekin bu huquqni murojaat spetsifikatorlari yordamida o’zgartirish mumkin. Agar sinf Class hizmatchi so’zi orqali ta'riflangan bo’lsa, uning hamma komponentalari hususiy bo’ladi. Lekin bu huquqni murojaat spetsifikatorlari yordamida uzgartirish mumkindir. Bu spetsifikator yordamida Sinflar umumiy holda quyidagicha ta'riflanadi:

```
class class_name
{
int data_member; // Ma'lumot-element
void show_member(int); // Funktsiya-element
};
```

Sinf ta'riflangandan so’ng, shu sinf tipidagi o’zgaruvchilarni(ob'ektlarni) qo’yidagicha ta'riflash mumkin:

```
class_name object_one, object_two, object_three;
```

Qo’yidagi misolda employee, sinfi kiritilgandir:

```
class employee
{
public:
char name[64] ;
long employee_id;
float salary;
void show_employee(void)
{
cout << "Imya: " << name << endl;
cout << "Nomer slujathego: " << employee_id << endl;
cout << "Oklad: " << salary << endl;
};
};
```

Bu sinf uch o’zgaruvchi va bitta funktsiya-elementga ega. Qo’yidagi EMPCLASS.CPP dastur ikki employee ob'ektini yaratadi. Nuqta operatordan foydalanib ma'lumot elementlarga qiymat beriladi so’ngra show_employee

elementidapn foydalanib hizmatchi haqidagi ma'lumot ekranga chiqariladi:

```
#include
#include
class employee
{
public:
char name [64];
long employee_id;
float salary;
void show_employee(void)
{
cout << "Imya: " << name << endl;
cout << "Nomer slujathego: " << employee_id << endl;
cout << "Oklad: " << salary << endl;
};
};
void main(void)
{
employee worker, boss;
strcpy(worker.name, "John Doe");
worker.employee_id = 12345;
worker.salary = 25000;
strcpy(boss.name, "Happy Jamsa");
boss.employee_id = 101;
boss.salary = 101101.00;
worker.show_employee();
boss.show_employee();
}
```