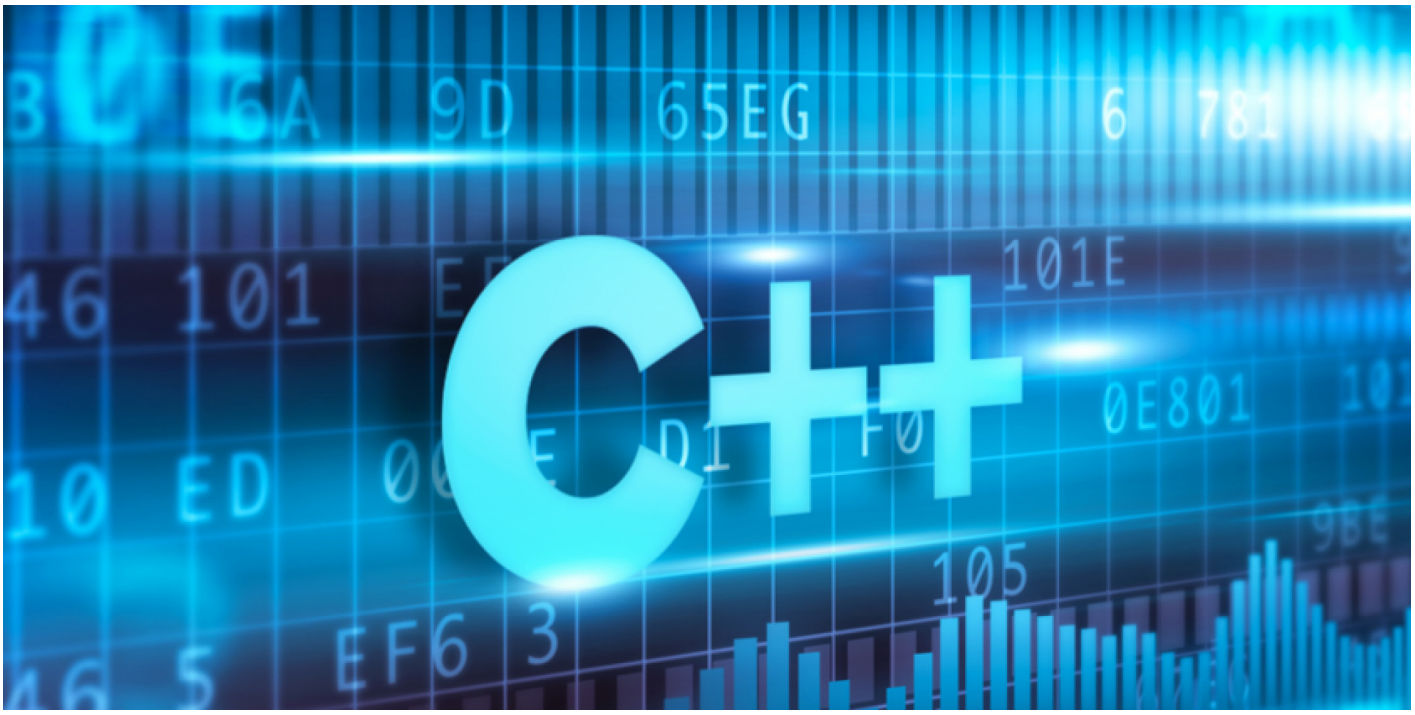
C++ tilida strukturalar va birlashmalar



Strukturalar

Structura nomi ko’cha manzili bu yolg’iz mohiyatga ikki ahamiyatni birlashtirish

uchun tariflanishi mumkin. C++da, biz struct odamovi so’zli structurani tariflaymiz:

struct ko’cha manzili

{

 int uy_raqami

 string ko’cha_ismi;

};

bu tariflash maydonlarisiz o’zgaruvchilarni elon qilish uchun foydalanadigan ko’cha manzilining yangi turi;

ko’cha manzili oq uy oq uy ozgaruvchisida azolar deb ataladigasan ikki qismi bor, uy_raqami va

ko’cha_nomi. Siz bu nuqtalardan har bir azolarning kirishida foydalanasiz, quyidagiga o’xshagan:

oq_uy,uy_raqami = 1600;

oq_uy,ko’cha_nomi = "Pennsylvania Avenue";

Structuralarga ko'rsatkichlar

Bu jo’shqin structura ahamiyatini ajratmasi uchun odatlangan, yangi operatordan foydalanish ko'rsatgich manzili * manzil_ko’rsatgichi yangi ko'rsatgich manzili. Faraz qiling manzil ko’rsatgich ko’rsatgichiga structuraning uy raqamini o’rnatishni xoxlaysiz manzil_ko’rsatgich,uy_raqami=1600;//XATOLIK

Baxtga qarshi,bu sintaksis xato. Bu nuqta operatori *operatoridan ko’ra ustunroq. Bu siz anglagan kompliyator o’ylari

(manzil_ko’rsatkich,uy_raqami)=1600;//XATOLIK

Manzil ko’rsatgichi ko’rsatgich bo’lishiga qaramay u structura emas. Siz operator ko’rsatgichiga va kompliyator xatolar haqida bayon qiladigan nuqtalarga etibor qila olmaysiz O’rniga siz birinchi*operatoriga keyin nuqtaga aniq etibor berishingiz kerak (manzil_ko’rsatgich,uy_raqami)=1600;//XATOLIK.Chunki bu shunchalik odatiy holatki C+ +ning loyihachilari.

Structuralar bilan ko'rsatgich a'zolar

Structuraning azosi ko'rsatgich bo'lishi mumkin. Bu holat odatda malumot structura qiymatlari orasida bolinganda vujudga keladi . Quyidagi misolni hisoblangTashkilotlarning ko'p sonli offislarida har bir ishchida ism va ish joylari bor.

struct Employee

{

string name;

StreetAddress* office;

}

Bu yerda biz ikki ishchining hisoblash idoralarini tariflaymiz:

StreetAddress accounting;

accounting.house_number = "1729";

accounting.street_name = "Park Avenue";

Employee harry;

harry.name = "Smith, Harry";

harry.office = &accounting;

Ko'cha manzili turi va viloyat: 2201 C ko'cha manzilli dastlabki o'zgaruvchilarni elon qilish.

32. Ko'cha manzili turi va viloyat: 2201 C Ko'cha NW. manzilidan tashkil topgan dastlabki o'zgaruvchilarni ko'rsatish.

33. Struktura turi sanasini 4-iyul yoki 31-dekabr kabi ko'rsatib tariflash.

34. Siz tariflagan 33 o'zgaruvchi turidan mustaqillik kuni va dastlabki 4-iyul kabi foydalanish.

35 Harryning offis manzilini qanday chop qilasiz?

amaliyot Hozir siz bu mashqlarni R7.23, R7.24, P7.16.boblar oxirida ko'rishingiz mumkin

ko'rsatgich ozgaruvchilaridan foydalanish va tariflash

Ko'rsatgich xotirada o'zgaruvchining turgan joyini bildiradi.

- T*turiko'rsatgich o'zgaruvchisining T turini bildiradi.
- & operator maydoni o'zgaruvchining turgan joyidir.
- * operator ko'rsatgich ko'rsatadigan o'zgaruvchining kirishiga ruxsat beradi

- Tajribasiz ko'rsatgich foydalanadigan xatolik.
- NULL ko'rsatgich biror obyektni ko'rsatmaydi

C++da nayzalar va ko'rsatgichlar orasidagi aloqani tushunish.

- Nayza o'zgaruvchisining nomi nayza elementing boshlanishiga ko'rsatgichdir.
- Ko'rsatgich arfimetigi berilgan elementlar raqamidan sakrab o'tadigan qo'shilgan butun son o'rnini qoplovchi nayza ko'rsatgichi yani maydonlash ko'rsatgichini anglatadi.
- Nayza/butun son o'rnini to'ldiradigan a[n] aniqlovchisi bo'lgan *(a+n) past holat ikkilamchi ko'rsatgichini ko'rsatadi.
- Nayzani funcsiyaga bosganda faqat manzil ko'rsatiladi.

C++string obyektlarini nayza xarakteridagi funksiyalar bilan ishlatish.

- Char turining ahamiyati shaxsiy xarakterga beriladi. Tom manolardagi xarakter-

lar birlikka asoslanib yopiladi.

- Tom manodagi string (juftlikka asoslanadi) u nol termini bilan char nayzasining ahamiyati
- Char tipining funksiyalari ko'plab

ko'rsatgichlardan foydalanadi.

- Char maydoni funksiya azosi string obyektini ko'rsatadi.
- Siz C++ozgaruvchilari bilan C stringlarni birinchi belgilab qoyishingiz mumkin • C++string obyekti bilan [] operator xarakterlariga birinchi kirishingiz mumkin.

Run vaqtigacha aniqlanmagan talabnoma xotirasi programma xotirasidan joy ajratadi va o'rnini aniqlaydi.

- Ajratilgan dinamic xotiradan foydalanish agar siz o'zingizga qancha qiymat ker-akligini bilmasangiz.

- To'dadan yangi operator ajratish.
- Dinamik ajratilgan obyekt bilan o'chirish yoki o'chirilgan obyekt bilan qaytarib olishingiz kerak.
- Erkin ko'rsatgichdan foydalanish

Birlashmalar va ular ustida amallar

Birlashmalar xotiraning bitta sohasida (bitta adres bo‘yicha) har xil turdagi bir nechta berilganlarni saqlash imkonini beradi.

Birlashma e’loni union kalit so‘zi, undan keyin identifikator va blok ichida har xil turdagi elementlar e’lonidan iborat bo‘ladi, masalan:

union Birlashma

{

int n;

unsigned long N;

char Satr[10];

};

Birlashmaning bu e’lonida kompilyator tomonidan Birlashma uchun uning ichidagi eng ko‘p joy egallovchi elementning - Satr satrining o‘lchamida, ya’ni 10 bayt joy ajratiladi. Vaqtning har bir momentida birlashmada, e’lon qilingan maydonlarning faqat bittasining turidagi berilgan mavjud deb hisoblanadi. YUqoridagi misol-da Birlashma ustida amal bajarilishida uning uchun ajratilgan xotirada yoki int turidagi n yoki unsigned long turidagi N yoki Satr satr qiymati joylashgan deb hisoblanadi.

Birlashma maydonlariga xuddi struktura maydonlariga muro-jaat qilgandek ‘.’ orqali murojaat qilinadi.

Strukturalardan farqli ravishda birlashma e’lonida faqat uning birinchi elementiga boshlang‘ich qiymat berish mumkin:

union Birlashma

```
{
 int n;
 unsigned long N;
 char Satr[10];
}
 birlashma={25};
```

Bu misolda birlashma birlashmasining n maydoni boshlang‘ich qiymat olgan hisoblanadi.

Birlashma elementi sifatida strukturalar 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 (4 bitga) ajratish va qaytadan birlashtirish mumkin.

Quyida baytni katta va kichik yarim baytlarga ajratishda birlashma va strukturadan foydalanilgan programmani matni kelti-rilgan.

```
#include
union BCD
          {unsigned char bayt;
           struct
     {
      unsigned char lo:4;
             unsigned char hi:4;
  } bin;
          } bcd;
```

```
int main()
{
        bcd.bayt=127;
        cout<<"\n Katta yarim bayt : "<<(int)bcd.bin.hi;
        cout<<"\n Kichik yarim bayt: "<<(int)bcd.bin.lo;
        return 0;
       }
```

Programma bosh funksiyasida bcd birlashmasining bayt o‘lchamida bayt maydoniga 127 qiymati beriladi va uning katta va kichik yarim baytlari chop etiladi.

Programma ishlashi natijasida ekranga quyidagi natijalar chiqadi:

Katta yarim bayt : 7

Kichik yarim bayt: 15

Masala. Haqiqiy turdagi sonning компьyuter xotirasidagi ichki ko‘rinishini chop qilish. Haqiqiy son float turida deb hisoblanadi va u xotirada 4 bayt joy egallaydi (1-ilovaga qarang). Qo‘yilgan masalani echish uchun birlashma xususiyatdan foydalaniladi, ya’ni xotiraning bitta adresiga haqiqiy son va belgilar massivi joylashtiriladi. Haqiqiy son xotiraga o‘qilib, belgilar massivining har bir elementining (baytining) ikkilik ko‘rinishi chop etiladi.

Programma matni:

```
#include
const unsigned char bitlar_soni=7;
const unsigned char format=sizeof(float);
```

```
void Belgi_2kodi(unsigned char blg);
union Son_va_Belgi
{
 float son;
 unsigned char belgi[format];
};
int main()
{
 Son_va_Belgi son_va_belgi;
 cin>>son_va_belgi.son;
 for(int b=format-1; b>=0; b--)
  Belgi_2kodi(son_va_belgi.belgi[b]);
 return 0;
}
void Belgi_2kodi(unsigned char blg)
{
 unsigned char l0000000=128;
 for(int i=0;i<=bitlar_soni;i++)
 {
  if(blg&l0000000)cout<<'1';
  else cout<<'0';
  blg=blg<<1;
 }
```

```
 cout<<' ';
}
```

Programmada Son_va_Belgi birlashmasini e’lon qilish orqali float turidagi x o‘zgaruvchisini va float turi formatining baytlardagi uzunligidagi belgilardan iborat belgi massivini xotiraning bitta joyiga joylashuviga erishiladi. Bosh funksiyada birlashma turidagi son_va_belgi o‘zgaruvchisi e’lon qilinadi va uning x maydoniga klaviaturadan haqiqiy son o‘qiladi. Keyin belgilar massividagi har bir elementning ikkilik kodi chop etiladi. Ikkilik kodni chop etish 8 marta baytni 7-razryadidagi sonni chop etish va bayt razryadlarini bittaga chapga surish orqali amalga oshiriladi. SHunga e’tibor berish kerakki, belgilar massividagi elementlarning ikkilik kodlarini chop qilish o‘ngdan chap tomonga bajarilgan. Bunga sabab, son ichki formatidagi baytlarning xotirada «kichik bayt - kichik adresda» qoidasiga ko‘ra joylashuvidir.

Programmaga -8.5 soni kiritilsa, ekranda

11000001 00001000 00000000 00000000

ko‘rinishidagi ikkilik sonlari ketma-ketligi paydo bo‘ladi.

Foydalanuvchi tomonidan aniqlangan berilganlar turi

C++ tilida foydalanuvchi tomonidan nafaqat struktura yoki birlashma turlari, balki ayni paytda mavjud (aniqlangan) turlar asosida yangi turlarni yaratishi mumkin.

Foydalanuvchi tomonidan aniqlanadigan tur typedef kalit so‘zi bilan boshlanadi, undan keyin mavjud tur ko‘rsatiladi va identi-fikator yoziladi. Oxirida yozilgan identifikator - yangi yaratilgan turning nomi hisoblanadi. Masalan,

typedef unsigned char byte;

ifodasi byte deb nomlanuvchi yangi turni yaratadi va o‘z mazmuniga ko‘ra unsigned char turi bilan ekvivalent bo‘ladi. Keyinchalik, programmada xotiradan bir bayt joy egallaydigan va [0..255] oraliqdagi qiymatlarni qabul qiladigan byte turidagi o‘zgaruvchi (o‘zgarmaslarni) e’lon qilish mumkin:

byte c=65;

byte Byte=0xFF;

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 (satr) e’lonidir.

Odatda echilayotgan masalaning predmet sohasi terminlarida ishlash uchun strukturalar qayta nomlanadi. Natijada murakkab tuzi-lishga ega bo‘lgan va zarur xususiyatlarni o‘ziga jamlagan yangi turlarni yaratishga muvofiq bo‘linadi.

Masalan, kompleks son haqidagi 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 maydonlariga murojaat qilish mumkin:

KSon.re=5.64;

KSon.im=2.3;