

**Farg‘ona transport va servis
texnikumi, axborot texnologiyaklari va
energetika kafedrası mudir, katta
o‘qituvchi Nargiz Alimova taqrizi
ostida**

**M.Sayidolimova
Farg‘ona transport va servis texnikumi,
axborot texnologiyaklari va energetika
kafedrası o‘qituvchi
m.sayidolimova@gmail.com**

CCC+ KLASLAR BILAN ISHLASH. KLASLARDA VORISLIK

Annotatsiya: Mazkur maqolada hozirgi kunda rivojlanib borayotgan va ijtimoiy hayotimizning ajralmas qismiga aylanga IT sohasi bo‘yicha e‘lon qilingan maqolalar hamda qilingan ishlar tahlil etilgan. Xususan CCC+ klasslar bilan ishlash va vorislikka e‘tibor qaratilgan.

Kalit so‘lar: Dasturlash tili, klasslar, CCC+, vorislik klaslarda, programma, ta‘minot, C++ tili.

РАБОТА С CCC+ КЛАССАМИ. НАСЛЕДОВАНИЕ В КЛАССАХ

Аннотация: В данной статье проанализированы опубликованные статьи и проделанная работа в сфере IT, которая в настоящее время развивается и становится неотъемлемой частью нашей общественной жизни. В частности, основное внимание уделяется работе с классами CCC+ и наследованию.

Ключевые слова: Язык программирования, классы, CCC+, наследование классов, программное обеспечение, программное обеспечение, язык C++.

WORK WITH CCC+ CLASSES. SUCCESSION IN CLASSES

Annotation: This article analyzes the published articles and the work done in the field of IT to become an integral part of our currently developing and social life. In particular, the focus is on working with CCC+ classes and succession.

Key words: Programming language, classes, CCC+, in successor classes, software, provisioning, C++ language.

Ma‘lumki, programma mashina kodlarining qandaydir ketma-ketligi bo‘lib, aniq bir hisoblash vositasini amal qilishini boshqaradi. Programma ta‘minotini yaratish jarayonini osonlash-tirish uchun yuzlab programmalash tillari yaratilgan. Barcha programmalash tillarini ikki toifaga ajratish mumkin:

- quyi darajadagi programmalash tillari;
- yuqori darajadagi programmalash tillari.

C++ tilida so‘z deb bir nechta belgilar ketma-ketligi tushuniladi. Xizmatchi so‘z deb C++ tilidagi standart nom tushuniladi. Bu maxsus ma‘noni anglatadi va uni ma‘lumotlarga berib bo‘lmaydi.

C++ dasturlash tili C tiliga asoslangan. C esa o‘z navbatida B va BCPL tillaridan kelib chiqqan. BCPL 1967 yilda Martin Richards tomonidan tuzilgan va operatsion sistemalarni yozish uchun mo‘ljallangan edi. C++ tilini 80-yillarda AT&T Bell Labs korxonasi ishchisi Byarnom Straustrup

tuzgan. Avtorning aytishicha, bu tilni tuzishda u hech qanday qog'ozlarni qoramagan va hamma ishni yo'l yo'lakay, ketma-ketlikda tuzib chiqqan. Uning asosiy maqsadi, o'zi va do'stlari uchun qulay dasturlash tili yaratishdan iborat edi. C++ dasturlash tilining asosi C hisoblanadi va shu tilni(C) misolida C++ tilini tuzib chiqdi. C dasturlash tilini mukammallashtirgan eng asosiy narsa bu – obyektga mo'ljallangan dasturlashni olib kirgani hisoblanadi. Chunki dasturlashda, tez, qulay, tartibli va ixcham yozish uchun ob'yektga mo'ljallangan dasturlash tillaridan foydalanish lozim bo'ladi. Dastlabki C++ dasturlash tilidan foydalanganlar, bu albatta Bell Labs korxonasi xodimlari

```

27         parol=x;
28     }
29     int inson::getparol()
30     {
31         return parol;
32     }
33     void inson::chiqar()
34     {
35         cout<<" ism fami
36         <<"\n kasbi: "
37         <<"\n yoshi: "
38         <<"\n maoshi: "
39     }
40     int main()

```

C++ dasturlash tili odatda foydalanuvchilarga birinchi bo'lib tavsiya etiladigan til sifatida qaraladi. C++ tilini oraliq til sifatida qarashimiz mumkin. U quyi darajada to'g'ridan to'g'ri qurilmalar uchun ham ishlay olsa, yuqori darajadagi til sifatida ham turli loyihalarda namoyon bo'la oladi. C++ yuqori darajadagi dasturlash tili bo'lib, u Bjarne Stroustrup tomonidan C tilining kengaytirilgan varianti sifatida yaratiladi. Bu tilning ustuvor tomoni unda Obyektga yo'naltirilgan dasturlashning mavjud bo'lganida deb qarash mumkin. C++ tili turli yo'nalishdagi mahsulotlar yaratishida ahamiyatli va mashhur bo'lgan, jumladan video o'yinlar, desktop dasturlari, serverlar, turli aloqa vositalarini misol sifatida ko'rish mumkin. C++ da yaratilgan mashhur dasturlar va platformalar hisoblanadi.

- ❖ Windows;
- ❖ Facebook (server)
- ❖ Adobe Photoshop, Illustrator, Premiere Pro;
- ❖ Mozilla Firefox;
- ❖ MySQL server;
- ❖ AutoDesk Maya;

C++ kabi kompilyatsiya qiluvchi dasturlash tillarini yana bir afzalligi hosil bo'lgan dastur kompyuterda kompilyatorsiz ham bajarilaveradi. Interpretatsiya qiluvchi tillarda esa tayyor dasturni ishlatish uchun albatta mos interpretator dasturi talab qilinadi. C ++ universal dasturlash tili bo'lib 束 xar xil darajadagi masalalar uchun yechim topish mumkin. C ++ tilining yana bir ahamiyatli tomoni shundaki, u ANSI standarti talablariga to'liq javob beradi to'g'rirog'i ANSI standarti talablari asosida yaratilgan. C ++ tilining asosiy tushunchalaridan biri bu klasslardir. Klass bu – foydalanuvchi tomonidan yaratilgan (ifodalangan) til. C ++ tilida C tilining deyarli barcha imkoniyatlari saqlangan. Bunga asosiy sabab C ++ tili yaratilayotgan bir paytda ko'plab foydalanuvchilar C tilida dasturlar yaratgandi. Bunday tayyor xoldagi dasturlarga qayta o'zgarish

kiritganda xam C ++ kompilyatori dastur matnidan xatoliklar topmaydi. Ya'ni dasturni xar ikkala tilda xam foydalanib tuzish mumkin.

```

3   using namespace std;
4   class talaba
5   {
6       private:
7       string ism;
8       public:
9       talaba(); // standart konstruktor
10      talaba(strin);
11  };
12  talaba::talaba
13  {
14      cout<<"tal
15  }
16  int main()
17  {

```

The screenshot shows a code editor with a search window open. The search results list various C++ standard library components, with 'string' highlighted in blue. The code in the background defines a 'talaba' class with a private 'ism' attribute and a public constructor and method call.

C ++ – bu ob'ektga yo'naltirilgan dasturlash tili. C++ dasturlash tilida hamma narsa uning xususiyatlari va usullari bilan bir qatorda sinflar va obyektlar bilan bog'liq. Sinf – bu bizning dasturimizda foydalanishimiz mumkin bo'lgan foydalanuvchi belgilaydigan ma'lumot turi va u obyekt tuzuvchisi yoki ob'ektlarni yaratish uchun "reja" sifatida ishlaydi. Bitta sinfning bir nechta ob'ektlarini yaratishingiz mumkin. Lekin bir klassga tegishli metodlar, boshqa klassdagi obyektlar uchun mavjud bo'lmasligi tabiiy. Misol uchun matnlar uchun mavjud metodlarni, butun yoki o'nli sonlarga qo'llab bo'lmaydi. C++ dagi sinflar C++ dagi sinflar-bu metodlar, xususiyatlar, hali mavjud bo'lmagan obyektlarni tavsiflovchi mavhumlik. Obyektlar - o'ziga xos xususiyatlar va metodlarga ega bo'lgan mavhumlikning o'ziga xos vakili. Bitta sinf asosida yaratilgan obyektlar ushbu sinfning namunalari deb nomlanadi. Ushbu obyektlar turli xil xatti-harakatlarga, xususiyatlarga ega bo'lishi mumkin, ammo baribir bir xil sinfning obyektlari bo'ladi

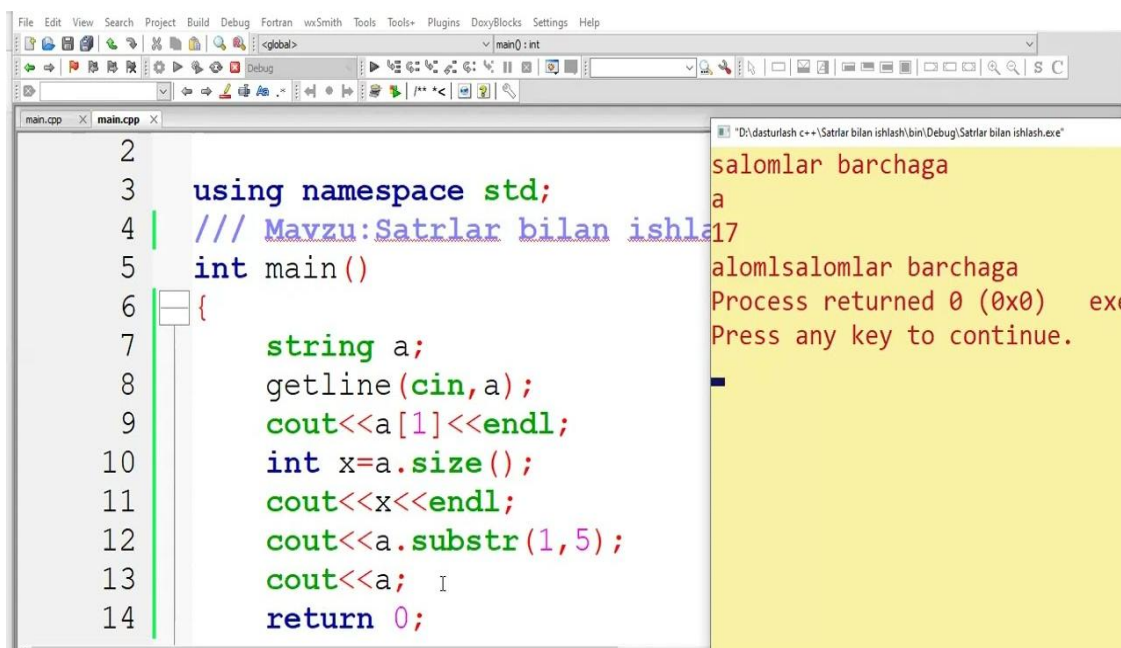
Sinf komponentalari sodda holda tiplangan ma'lumotlar va funktsiyalardan iborat bo'ladi. Figurali qavslarga olingan komponentalar ro'yxati sinf tanasi deb ataladi. Sinfga tegishli funktsiyalar komponenta-funksiyalar yoki sinf funktsiyalari deb ataladi. Sinf komponenta o'zgaruvchilari sifatida o'zgaruvchilar, massivlar, ko'rsatkichlar ishlatilishi mumkin. Elementlar ta'riflanganda initsializatsiya qilish mumkin emas. Buning sababi shuki sinf uchun hotiradan joy ajratilmaydi. Komponenta elementlariga komponenta funktsiyalar orqali murojat qilinganda faqat nomlari ishlatiladi. Sinfdan tashqarida sinf elementlariga emas obyekt elementlariga murojaat qilish mumkin. Bu murojaat ikki hil bo'lishi mumkindir. Obyekt- nomi;

element-nomi.

Obyktga–korsatgich–element nomi. Sinf elementlari sinfga tegishli funktsiyalarida ishlatilishidan oldin ta'riflangan bo'lishi shart emas. Xuddi shunday bir funktsiyadan hali ta'rifi berilmagan ikkinchi funktsiyaga murojaat qilish mumkin. Komponentalarga murojaat huquqlari. Komponentalarga murojaat huquqi murojaat spetsifikatorlari yordamida boshqariladi.

Obyektga yo'naltirilgan dasturlashning qulayliklaridan biri bu klasslardan boshqa klass yaratish imkoniyati. Bizga kerak bo'lgan yangi klass, avval yaratilgan boshqa klass bilan o'xshashlik

joylari bo'lsa, biz bu klassdan voris klass yaratishimiz mumkin. Bunda asl klass - ota, yoki super klass deb ataladi. Super klassdan yaratilgan voris klass otaning barcha yoki tanlangan xususiyatlari va metodlarini meros olish bilan birga, o'ziga xos xususiyat va metodlariga ega bo'ladi. Bu spetsifikatorlar: Protected – himoyalangan; Private – hususiy; Public – umumiy; Himoyalangan komponentalardan sinflar ierarhiyasi qurilganda foydalaniladi. Oddiy holda Protected spetsifikatori Private spetsifikatoriga ekvivalentdir. Umumiy ya'ni Public tipidagi komponentalarga dasturning ixtiyoriy joyida murojaat qilinishi mumkin. Xususiy ya'ni Private tipidagi komponentalarga sinf tashqarisidan murojaat qilish mumkin emas. Agar sinflar Struct xizmatchi 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 xizmatchi so'zi orqali ta'riflangan bo'lsa, uning hamma komponentalari xususiy bo'ladi. Lekin bu huquqni murojaat spetsifikatorlari yordamida uzgartirish mumkindir.



The screenshot shows a C++ IDE with a code editor on the left and a console window on the right. The code in the editor is as follows:

```
2
3 using namespace std;
4 // Mavzu: Satrlar bilan ishlash
5 int main()
6 {
7     string a;
8     getline(cin, a);
9     cout<<a[1]<<endl;
10    int x=a.size();
11    cout<<x<<endl;
12    cout<<a.substr(1,5);
13    cout<<a;
14    return 0;
```

The console window shows the following output:

```
salomlar barchaga
a
17
alomsalomlar barchaga
Process returned 0 (0x0)   exe
Press any key to continue.
```

Ba'zida klassimiz xususiyatlar va ular bilan ishlaydigan metodlarga to'lib ketishi, bu esa o'z navbatida obyektga murojat qilishni qiyinlashtirishi mumkin. Shunday holatlarda ba'zi xususiyatlarni alohida klass ko'rinishida yozish va keyinchalik bu klassdan yaratilgan obyektning boshqa obyektning xususiyati sifatida foydalanish mumkin.

Pythonda hech qanday vazifani bajarmaydigan pass operatori mavjud. Bu operatordan bo'sh metodlar yaratishda foydalanish mumkin. Misol uchun siz klassingiz uchun muhim metodlarni bilasiz, lekin metod badani hali tayyor emas. Agar metod badanini bo'sh qoldirsangiz, Python **IndentationError** xatosini qaytaradi. Shunday xolatlarda, funksiya badaniga pass operatorini qo'yib ketishimiz mumkin

Foydalanilgan manbalar:

1. Madraximov Sh. C++ tilda programmalash asoslari. – Toshkent: 2009.
2. <https://python.sariq.dev/oop/30-vorislik-polimorfizm>
3. <https://programmer.uz/course/cpp-dasturlash-tili>
4. <https://staff.tiame.uz/storage/users/126/presentations/ty77pnmZ37WBb1O5VfwqWzKD0oiTQJ1xln4N2Efn.pdf>
5. <https://uzbekdevs.uz/darsliklar/cpp/cpp-da-sinflar-obektlar>

6. https://elibrary.namdu.uz/_dasturlash_tilida_massivlar_ustida_amallar_va_tenglamalar_sistemasi_gauss_usulida_yechish.
7. <https://www.google.com/search?>
8. <https://eduproger.uz/blog/dasturlash-asoslari-c/c-tilida-sinflan>