

C++ DASTURLASH TILIDA FAYLLAR BILAN ISHLASH VA ULARNI YARATISH USULLARI

Manshurov Sherzod To'ychiboyevich

Islom Karimov nomidagi TDTU Olmaliq filiali

“Matematika va informayika” katta o'qtuvchisi

E-mail: manshurov_sh@mail.ru

ANOTATSIYA

C++ dasturlash tili keng qo'llaniladigan til va odatda dasturchilar tomonidan ishlatiladi. Fayllar yaratish C++ dasturlashining asosiy qismidir, shuning uchun bu mavzuga katta e'tibor qaratish zarurdir.

C++ da fayllarni yaratish usullari juda oson va qulaydir. C++ dasturchilari fayllarni yaratish uchun standard kutubxonalardan foydalanasiz. Standart kutubxonalar juda kuchli va yorqin hisoblanadi va fayllarni yaratish uchun kerakli funksiyalarni o'z ichiga oladi.

Fayl yaratish uchun, "fstream" kutubxonasi ishlatiladi. Bu kutubxona fayllarni ochish, yozish va yopish funksiyalarini o'z ichiga oladi. "ofstream" fayllarni yaratish uchun va fayllarga yozish uchun ishlatiladi, "ifstream" esa fayllarni ochish va fayllardan ma'lumot olish uchun ishlatiladi.

Fayllarni yaratish uchun asosiy qadam - faylni ochish va faylga yozish yoki fayldan ma'lumot olish. Fayllarni ochish uchun "ofstream" kutubxonasi yoki "ifstream" kutubxonasi ishlatiladi. Faylni ochganingizdan keyin, faylga yozish yoki fayldan o'qish funksiyalari ishlatiladi. Faylni yopish uchun "close" funksiyasi ishlatiladi.

Fayllarni yaratish jarayoni keng qo'llaniladi va C++ dasturchilari uchun qulaydir. Standart kutubxonalar va ularning funksiyalari fayllarni yaratish, ochish, yozish va yopish jarayonini osonlashtiradi.

Kalit so'zlar: C++, fayllar, fayl kiritish/chiqarish, ofstream, ifstream, fstream, open(), close(), is_open(),

KIRISH

C++ dasturlash fayllar dasturchilar tomonidan ma'lum maqsadlarga erishish uchun yaratiladi. Fayllar odatda ma'lumotlarni saqlash, ulardan foydalanish va ma'lumotlar bilan ishlashni osonlashtirish uchun foydalaniladi.

Fayllar odatda ma'lumotlar turi, tashqi xususiyatlari va turli formatlarga ega bo'lishi mumkin. Fayllar matn, rasmlar, audio va video, ma'lumotlar, dasturlar va boshqa ko'plab ma'lumotlar turlarini saqlash uchun ishlatiladi.

Fayllar nomi, turi va formati odatda katta-kichik harflarda yoziladi va fayl nomi bilan bir qatorda turli ajratuvchi belgilar foydalaniladi, masalan ".txt" fayl formatini bildiradi.

Fayllar tashqi xotiralarda joylashgan bo'lishi mumkin, masalan kichik disketlarda yoki katta xotira qurilmalarida saqlanishi mumkin. Fayllar har xil formatlarda saqlanishi mumkin, va ularning formati, ularni ochish va tahrir qilish uchun foydalaniladigan dasturlar bilan bog'liq.

Fayllar va direktoriyalar

Fayllar va kataloglar operatsion tizimda fayl va papkalarni yaratish va tartibga solishning asosiy vositalaridir. Fayllar bir yoki bir nechta satrlardan iborat bo'lishi mumkin bo'lgan matn, raqamli, audio yoki video ma'lumotlarni saqlash uchun ishlatiladi. Boshqa tomondan, kataloglar fayllarni tartibga solish uchun ishlatiladi. Fayllar, bir yoki undan ko'p qatorlar mavjud bo'lgan matnli, raqamli, audio yoki video ma'lumotlarni saqlash uchun ishlatiladi. Direktoriyalar esa fayllarni tuzish uchun ishlatiladi.

Direktoriya, bir nechta fayllarni yoki boshqa direktoriyalarni saqlash uchun eng oddiy vosita bo'lib, bu tizimni yaxshi tashkil etishda juda muhimdir. Fayllar va direktoriyalar, operatsion tizimida faqat nomi yoki joylashgan manzili bilan aniqlanadi. Fayllarning nomlari odatda matnli shaklda, masalan, "myfile.txt" kabi ishlatiladi.

C++ dasturlash tilida, fayllar va direktoriyalar bilan ishlash uchun standart C++ kutubxonasida "fstream" nomli kutubxona mavjud. Bu kutubxona fayl bilan ishlashga imkon beradi: yangi fayllarni yaratish, mavjud fayllarga murojat, fayllardan

ma'lumotlarni yuklash va yozish, fayllardan o'qish, yaratilgan faylni o'chirish, fayl o'lchamini olish va boshqalar kabi ishlar amalga oshirilishi mumkin.

Файллари yaratish

C++ da fayllar yaratish uchun quyidagi usullar mavjud:

1. I/O standart kutubxonasidan foydalanish: Fayl yaratish uchun, I/O standart kutubxonasidagi fstream sinfi yordamida fayl ochish kerak. fstream sinfi, ifstream (faylni o'qish uchun) va ofstream (faylga yozish uchun) sinflari bilan hamkorlik qiladi. Masalan:

```
#include <iostream>
#include <fstream>
using namespace std;
int main() {
    ofstream myfile; // yaratilayotgan fayl uchun ofstream sinfi obyekt
    myfile.open("example.txt"); // faylni ochish
    myfile << "Bu faylga yozilgan yozuv" << endl; // faylga yozuv kiritiladi
    myfile.close(); // faylni yopish
    return 0;
}
```

Файллари saqlash va ochish

Fayllarni saqlash va ochish C++ da faqatgina qo'llanmalar yordamida amalga oshiriladi.

Faylni saqlash uchun, ofstream (output file stream) klassi yordamida yaratilgan obyekt yordamida fayl ochishimiz va faylga ma'lumotlar yozilishi mumkin. Misol uchun:

```
#include <fstream>
using namespace std;

int main() {
```

```
ofstream myfile;  
myfile.open("example.txt");  
myfile << "This is some text.";  
myfile.close();  
return 0;  
}
```

Ushbu kod faylni ochadi, uni "example.txt" nomi bilan saqlashni boshlaydi va faylga "This is some text." yozuvini yozadi.

Faylni ochish uchun esa ifstream (input file stream) klassi yordamida yaratilgan obyekt yordamida fayl ochishimiz va fayldan ma'lumot olishimiz mumkin. Misol uchun:

```
#include <fstream>  
#include <iostream>  
using namespace std;  
  
int main() {  
    string line;  
    ifstream myfile("example.txt");  
    if (myfile.is_open()) {  
        while (getline(myfile, line)) {  
            cout << line << '\n';  
        }  
        myfile.close();  
    }  
    else cout << "Unable to open file";  
    return 0;  
}
```

Ushbu kod esa faylni ochadi, uni "example.txt" nomi bilan ochadi va fayldan qator-qator joylashtirilgan ma'lumotlarni o'qib chiqaradi. Faylni ochish vaqti `is_open()` metod yordamida tekshiriladi. Faylni o'qish yakunlanganda fayl yopiladi.

Muhim e'tibor bermang, fayl nomi va yo'nalishi mavjud bo'lishi kerak, aks holda fayl yaratish yoki ochishda xatolik yuzaga kelishi mumkin.

Faylni ochish va ma'lumotlarni o'qish

Faylni ochish va faylning ma'lumotlarini o'qish uchun C++ dasturida `fstream` kutubxonasidan foydalaniladi. Quyidagi kodda faylni ochish, fayldan ma'lumot o'qish va o'qilgan ma'lumotlarni konsolga chiqarish ko'rinishdadir:

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int main() {
    // Faylni ochish uchun fayl nomi bilan obyekt yaratamiz
    ifstream infile("example.txt");

    // Fayldan ma'lumotlarni o'qish
    string line;
    while (getline(infile, line)) {
        cout << line << endl;
    }

    // Faylni yopish
    infile.close();
}
```

```
return 0;  
}
```

Bu kod example.txt nomli faylni ochadi va faylning har bir qatorini o'qib chiqaradi. Faylni yopish uchun esa close funksiyasidan foydalaniladi.

Anonim fayllar va kataloglar

Anonim fayllar va kataloglar, C++ dasturlash tilida, fayllar va direktoriyalar bilan ishlashda qulayliklar yaratish uchun foydalaniladigan konseptlardir. Anonim fayllar, nomi olmayan fayllar hisoblanadi va direktoriyalar ichidagi qo'shimcha fayllar sifatida ishlatiladi.

Anonim fayllar, qo'shimcha ma'lumotlarni saqlash uchun foydalaniladi. Masalan, dasturlash davomida, fayllar bilan ishlashda vaqt va yodgorlik savdo qilishda, anonim fayllardan foydalanish juda ko'p keladi. Anonim fayllarni yaratish uchun std::tmpnam va std::tmpfile funksiyalari mavjud.

Kataloglar esa, fayllarni qo'shish, o'chirish va boshqa amallarni bajarish uchun yaratilgan direktoriyalar hisoblanadi. Kataloglar ichidagi fayllar va direktoriyalarni ko'rish va uni yaratish uchun std::opendir va std::mkdir funksiyalari foydalaniladi.

Anonim fayllar va kataloglar, dasturchilar uchun fayllar bilan ishlashni osonlashtiradigan yordamchi vositalardir.

Anonim fayllarni yaratish

Anonim fayllar (yoki tmp fayllar) C++ dasturlash tilida fayllar bilan ishlashda o'z-o'zini ko'rsatadigan ajratilgan fayllar hisoblanadi. Bu fayllar qo'shimcha saqlash joyi va muvofiqligi kerak bo'lmaydigan vaqtchiq ma'lumotlarni saqlash uchun foydalaniladi. Anonim fayllarni yaratish uchun tmpfile() funksiyasidan foydalaniladi.

Quyidagi misol anonim fayl yaratishni ko'rsatadi:

```
#include <cstdio>
```

```
int main() {  
    FILE* tmp_file = tmpfile();  
  
    if (tmp_file != nullptr) {  
        fputs("Hello, world!", tmp_file);  
        rewind(tmp_file);  
  
        char buf[64];  
        fgets(buf, sizeof(buf), tmp_file);  
  
        printf("Read from temp file: %s\n", buf);  
  
        fclose(tmp_file);  
    }  
    else {  
        perror("Failed to create temporary file");  
    }  
  
    return 0;  
}
```

Bu dastur anonim fayl yaratadi, faylga "Hello, world!" matnini yozadi, faylni o'qib ko'radi va faylni yopadi. tmpfile() funksiyasi muvaffaqiyatli ish bajarilganda FILE* tipidagi fayl nusxasini qaytaradi, aks holda esa nullptr qiymatini qaytaradi. Fayl yaratilmay qolsa, perror() funksiyasi xatolik haqida xabar chiqaradi. fputs() funksiyasi matnli ma'lumotni faylga yozish uchun, rewind() funksiyasi faylni boshiga qaytish uchun va fgets() funksiyasi faylni o'qish uchun ishlatiladi. Faylni yopish uchun fclose() funksiyasidan foydalaniladi.

Файллари birlashtirish va ajratish

C++ da fayllarni birlashtirish va ajratish uchun, fstream kutubxonasidagi ofstream va ifstream sinflari ishlatiladi.

ofstream sinfi, faylni yaratish va yozish uchun ishlatiladi, ifstream sinfi esa faylni ochish va o'qish uchun ishlatiladi. Shu sinflardan foydalanib, fayllarni birlashtirish va ajratish amallari bajariladi.

Quyidagi kod misoli, file1.txt va file2.txt nomli ikkita faylni yaratadi, file3.txt nomli faylda, file1.txt va file2.txt dan olingan ma'lumotlarni yozadi va keyin file4.txt va file5.txt nomli fayllarga file3.txt dan olingan ma'lumotlarni yozadi.

Bu fayllarni birlashtirish va ajratish uchun asosiy qadamlarni ko'rsatadi. Fayllarni boshqa usullarda ajratish va birlashtirish ham mumkin, misol uchun stringstream sinfi yordamida.

YAKUNIY QISM

Bu ko'chirish jarayonidagi bir nechta variantlar bilan fayllarni nusxalash, faylni yaratish, matn yozish, matnni o'qish va faylni ochishning bir qancha qulay usullarini o'rganish uchun qo'llanilgan dasturlash tillari va ularning imkoniyatlarini ko'rib chiqdik. C++ tilida "fstream" kutubxonasi va uning "ofstream" va "ifstream" sinflari yordamida fayllarni yaratish, matn yozish va ochish haqida ko'proq tushuntirildi. Anonim fayllar yaratish bilan bir qator ishlar bajarildi va fayllarni birlashtirish va ajratish haqida ham ma'lumot berildi.

ADABIYOTLAR

1. Stroustrup, Bjarne. The Design and Evolution of C++. Addison-Wesley Professional, 1994.
2. Stroustrup, Bjarne. The C++ Programming Language. Addison-Wesley Professional, 2013. ISBN-13: 978-0321563842.

3. Meyers, Scott. Effective C++: 55 Specific Ways to Improve Your Programs and Designs. Addison-Wesley Professional, 2005. ISBN-13: 978-0321334879.
4. Lippman, Stanley B., Josée Lajoie, and Barbara E. Moo. C++ Primer. Addison-Wesley Professional, 2012. ISBN-13: 978-0321714114.
5. Koenig, Andrew, and Barbara E. Moo. Accelerated C++: Practical Programming by Example. Addison-Wesley Professional, 2000. ISBN-13: 978-0201703535.
6. Williams, Anthony. C++ Concurrency in Action: Practical Multithreading. Manning Publications, 2019. ISBN-13: 978-1617294693.
7. Josuttis, Nicolai M. The C++ Standard Library: A Tutorial and Reference. Addison-Wesley Professional, 2012. ISBN-13: 978-0321623218.
8. Alexandrescu, Andrei. Modern C++ Design: Generic Programming and Design Patterns Applied. Addison-Wesley Professional, 2001. ISBN-13: 978-0201704310.
9. Sutter, Herb. Exceptional C++: 47 Engineering Puzzles, Programming Problems, and Solutions. Addison-Wesley Professional, 1999. ISBN-13: 978-0201615623.