19

O‘N TO‘QQIZINCHI MA’RUZA · 2-QISM

Dasturlash 1 · C++ · 13-mavzu (2 qismdan 2-qism)

Fayllar bilan ishlash

Oqim holati va pozitsiya · Binar fayllar · CSV va RAII

<fstream> | ios::app | getline | binary

ISINISH

Kitobning 200-betini topish uchun kitobni boshidan o‘qiysizmi?

O‘tgan ma’ruzada fayl oqimlari, ochish rejimlari va o‘qish usullarini o‘rgandik.

1

Fayldagi 1000-yozuvni qanday tez topish mumkin?

2

Son o‘rnida so‘z uchrasa, `f >> x` nima qiladi?

3

Talabalar ro‘yxatini Excel dan qanday formatda olish qulay?

O‘QUV MAQSADLARI

Bu ma’ruza oxirida siz quyidagilarni bajara olasiz...

1. **Boshqarish** — oqim holati bayroqlari va fayl pozitsiyasini (seekg, tellg)
2. **Yozish/o‘qish** — strukturalarni binar faylga write/read bilan
3. **Tahlil qilish** — CSV faylni va xatolarni RAII uslubida qayta ishlash
4. **Tanlash** — vazifaga mos fayl turini: matnli yoki binar

04

Oqim holati va pozitsiya

good, eof, fail, bad · clear · ignore · seekg, tellg

fail()

clear()

seekg

tellg

Oqim o‘z holatini qanday bildiradi?

Metod	Qachon true
`good()`	Hech qanday xato yo‘q
`eof()`	Fayl oxiriga yetildi
`fail()`	Amal muvaffaqiyatsiz: son o‘rnida harf (yoki bad)
`bad()`	Jiddiy xato: disk xatosi, oqim buzilgan
`clear()`	Barcha xato bayroqlarini tozalaydi

Xato bayrog‘i o‘rnatilsa, `clear()` chaqirilmaguncha keyingi barcha o‘qishlar bajarilmaydi.

“12 34 abc 56”

```cpp
{ ofstream f("a.txt"); f << "12 34 abc 56"; }
ifstream f("a.txt");
int x = -1;
f >> x; cout << x << " ";
f >> x; cout << x << " ";
f >> x; cout << x << " fail=" << f.fail() << " ";
f >> x; cout << x << " ";
f.clear();
string s; f >> s;
f >> x; cout << "| " << s << " " << x;
```

▶ NATIJA

```
12 34 0 fail=1 0 | abc 56
```

abc

son emas — failbit, x = 0 (C++11)

keyingi f >> x

bajarilmaydi — oqim xato holatida

clear()

bayroqlar tozalandi; abc ni satr sifatida o‘qib, davom etamiz

POZITSIYA

seekg va tellg — faylning istalgan joyiga

```
{ ofstream f("abc.txt"); f << "ABCDEFGHIJKLMNOPQRSTUVWXYZ"; }
ifstream f("abc.txt");
f.seekg(0, ios::end);
cout << "Hajm: " << f.tellg() << endl;
f.seekg(10);
char c; f.get(c);
cout << "10-bayt: " << c << ", tellg = " << f.tellg() << endl;
f.seekg(-3, ios::end);
string o; f >> o;
cout << "Oxiri: " << o;
```

▶ NATIJA

```
Hajm: 26
10-bayt: K, tellg = 11
Oxiri: XYZ
```

`seekg(0, ios::end)`

oxiriga — tellg() = fayl hajmi

`seekg(10)`

0 dan sanaladi: A=0 … K=10

`seekg(-3, ios::end)`

oxiridan 3 bayt oldin

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

4-bo‘lim yakuni: Oqim holati va pozitsiya

1

int o‘qilayotganda “abc” uchrasa nima bo‘ladi?

Savol 1

2

Xato holatidan qanday chiqiladi?

Savol 2

3

Fayl hajmini qanday bilish mumkin?

Savol 3

AMALIY TOPSHIRIQ

Har kim o‘z qurilmasida · 10 daqiqa · O‘rta · 20 ball

4-topshiriq: Buzilgan o‘lchovlar

`olchovlar.txt` faylida datchik o‘lchovlari bor, lekin ba’zi qiymatlar buzilgan — son o‘rnida so‘z yozilgan. To‘g‘ri sonlarning yig‘indisini va sonini, buzilgan yozuvlar sonini toping. O‘qish xatosidan keyin oqim holatini `clear()` bilan tiklang.

Kirish: `olchovlar.txt` fayli: bo‘sh joy bilan ajratilgan elementlar — butun son yoki harf bilan boshlanadigan so‘z.

Chiqish: Uch qator: to‘g‘ri sonlar yig‘indisi, to‘g‘ri sonlar soni, buzilgan yozuvlar soni.

MISOL · olchovlar.txt

```
12 abc 7 -3
xato 5
```

MISOL · CHIQISH

```
21
4
2
```

Telefon kamerasi bilan skanerlang

1. Saytga kiring (login va parol)
2. Kodni yozing va sinab ko‘ring
3. ✓ Topshirish — natija o‘qituvchiga

05

Binar fayllar

Matnli va binar farqi · write / read · reinterpret_cast · yozuvni joyida yangilash

ios::binary | write | read | sizeof(T)

TAQQOSLASH

Matnli yoki binar?

	Matnli	Binar
Ko‘rinishi	Odam o‘qiy oladi	Maxsus dastursiz o‘qilmaydi
Yozuv o‘lchami	O‘zgaruvchan: 7 va 1234567	Qat’iy: `sizeof(T)` bayt
i-yozuvga o‘tish	Boshidan o‘qib borish	`seekg(i * sizeof(T))`
Tezlik	Son ↔ matn o‘zgartirish	O‘zgartirishsiz — tez
Ko‘chma	Hamma joyda bir xil	Tizimga bog‘liq bo‘lishi mumkin

Binar faylga faqat “oddiy” strukturalar (sonlar, char massivlari). string/vector/ko‘rsatkichda faqat manzil yoziladi — ma’nosiz!

WRITE / READ

Strukturalarni binar saqlash

```
struct Talaba { int id; float ball; char ism[8]; };
int main() {
    Talaba g[3] = {{1001, 91.5f, "Aziza"},
                   {1002, 78.0f, "Bekzod"},
                   {1003, 85.25f, "Charos"}};
    { ofstream f("g.bin", ios::binary);
      f.write(reinterpret_cast<const char*>(g), sizeof(g)); }
    ifstream f("g.bin", ios::binary);
    f.seekg(2 * sizeof(Talaba));
    Talaba t;
    f.read(reinterpret_cast<char*>(&t), sizeof(t));
    cout << sizeof(Talaba) << " " << t.id << " " << t.ism << " "
            << t.ball;
}
```

▶ NATIJA

```
16 1003 Charos 85.25
```

sizeof(Talaba) = 16

4 + 4 + 8 bayt

f.write(...)

butun massiv bir yo‘la

seekg(2 * 16)

3-yozuvga to‘g‘ridan-to‘g‘ri o‘tish

JOYIDA YANGILASH

Faqat bitta yozuvni o‘zgartirish

```
struct M { int kod, soni; };
int main() {
    M m[3] = {{1, 10}, {2, 20}, {3, 30}};
    { ofstream f("m.bin", ios::binary);
      f.write(reinterpret_cast<const char*>(m), sizeof(m)); }
    fstream f("m.bin", ios::in | ios::out | ios::binary);
    M x;
    f.seekg(1 * sizeof(M));
    f.read(reinterpret_cast<char*>(&x), sizeof(x));
    x.soni = 50;
    f.seekp(1 * sizeof(M));
    f.write(reinterpret_cast<const char*>(&x), sizeof(x));
    f.seekg(0);
    while (f.read(reinterpret_cast<char*>(&x), sizeof(x))) cout
            << x.soni << " ";
}
```

▶ NATIJA

```
10 50 30
```

`in | out | binary`

o‘qish va yozish bitta oqimda

`seekp(1 * sizeof(M))`

yozish pozitsiyasi — 2-yozuv

Faqat 8 bayt

butun fayl qayta yozilmaydi — ma’lumotlar bazalari shunday ishlaydi

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

5-bo‘lim yakuni: Binar fayllar

1

Strukturani binar faylga yozishning to‘g‘ri usuli?

Savol 1

2

Binar faylda i-yozuvga qanday o‘tiladi?

Savol 2

3

Nima uchun string maydonli struct ni binar yozib bo‘lmaydi?

Savol 3

06

CSV va RAII

CSV formati · getline(ss, maydon, ',') · stod va xatolar · fayl oqimlari RAII

CSV | stringstream | try/catch | RAII

CSV TAHLILI

Vergul bilan ajratilgan jadval

```
string csv = "Aziza,911-24,91.5\nBekzod,911-24,78\nCharos,912-24,abc\nDilnoza,912-24,96";
stringstream fayl(csv);
string qator;
double jami = 0; int n = 0, xato = 0;
while (getline(fayl, qator)) {
    stringstream ss(qator);
    string ism, guruh, b;
    getline(ss, ism, ','); getline(ss, guruh, ','); getline(ss,
           b, ',');
    try { jami += stod(b); n++; }
    catch (const invalid_argument &) { xato++; }
}
cout << n << " ta, o'rtacha " << jami / n << ", xato: " << xato;
```

▶ NATIJA

```
3 ta, o'rtacha 88.5, xato: 1
```

`getline(ss, ism, ',')`

ajratuvchi belgi — vergul

`stod(b)`

“abc” bo‘lsa — invalid_argument istisnosi

`xato++`

noto‘g‘ri qator o‘tkaziladi, dastur to‘xtamaydi

Fayl o‘zi yopiladi

ofstream va ifstream — RAII sinflari: fayl konstruktorda ochiladi, destruktorda buferi yozilib, avtomatik yopiladi.

Kichik blok

`{ ofstream f(...); f << x; }` — blok tugaganda yopiladi.

Funksiya ichida

Har bir amal o‘z oqimini ochadi — `close()` unutilmaydi.

Istisnoda ham

Xato tashlansa ham destruktor faylni yopadi.

`close()` ni qo‘lda chaqirish faqat faylni o‘sha joyda qayta ochish kerak bo‘lganda zarur.

KUNDALIK SINFI

Fayl ustida qulay interfeys

```
class Kundalik {
    string nom;
public:
    explicit Kundalik(string n) : nom(n) {}
    void qosh(const string &y) {
        ofstream f(nom, ios::app);
        if (!f) throw runtime_error("ochilmadi");
        f << y << '\n';
    }
    int soni() const {
        ifstream f(nom); string q; int k = 0;
        while (getline(f, q)) k++;
        return k;
    }
};
int main() {
    Kundalik k("kun.txt");
    k.qosh("Dushanba: C++"); k.qosh("Seshanba: fayllar");
    cout << k.soni();
}
```

▶ NATIJA

2

`ofstream f(nom, ios::app)`

har qo‘shishda qisqa vaqt ochiladi

`}`

metod tugaganda fayl avtomatik yopiladi

`throw`

ochilmasa — tushunarli istisno

TUSHUNISHNI TEKSHIRISH

O‘ylang – Juftlikda – Bo‘lishing · 3 daqiqa

6-bo‘lim yakuni: CSV va RAII

1

CSV qatorini maydonlarga qanday ajratasiz?

Savol 1

2

`stod("abc")` nima qiladi?

Savol 2

3

Nima uchun blok tugaganda ofstream ma’lumoti yo‘qolmaydi?

Savol 3

XULOSA

Asosiy xulosalar

1 Holat

fail → `clear()`; pozitsiya — `seekg/tellg`, `seekp`.

2 Binar

`write/read` + `reinterpret_cast`; yozuv o‘lchami `sizeof(T)`.

3 CSV va RAII

`getline(ss, m, ',')`, `stod` xatosini ushlash; oqim o‘zi yopiladi.

TAKRORLASH

Yakunlashdan oldin oltita savol

1. Xato bayroqlarini tozalash?
2. Fayl hajmini aniqlash?
3. Binar yozish funksiyasi?
4. CSV maydonini ajratish?
5. `f.fail()` qachon rost bo‘ladi?
6. RAII fayl oqimi uchun nimani kafolatlaydi?

LABORATORIYA ISHI

13-laboratoriya ishi: 6 ta dasturlash topshirig‘i

5 **O‘tgan talabalar**
Oson · 10 ball

6 **CSV: tovarlar**
O‘rta · 20 ball

7 **Raqamlarsiz nusxa**
Oson · 10 ball

8 **Binar fayl: yozuvlar**
Qiyin · 30 ball

9 **Teskari tartibdagi satrlar**
Oson · 10 ball

10 **Server jurnali tahlili**
O‘rta · 20 ball

Telefon kamerasi bilan skanerlang

Kod saytda yoziladi va server tomonidan avtomatik tekshiriladi.

O‘qituvchi kodingizni ko‘rib, baho qo‘yadi.

CHIQISH CHIPTASI

Darsni tark etishdan oldin

Javoblaringizni qog‘ozga yozing va o‘qituvchiga topshiring.

1

Talaba (id, ball) yozuvini binar faylga yozish va qayta o‘qish qadamlarini sanab bering.

2

O‘qish xatosidan keyin `clear()` chaqirilmasa nima bo‘ladi?

3

Baholar jurnali uchun matnli yoki binar fayldan qaysi birini tanlaysiz? Nega?