18

O‘N SAKKIZINCHI MA’RUZA · 1-QISM

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

Fayllar bilan ishlash

Fayl va fayl oqimlari · Ochish rejimlari · Matnli fayldan o‘qish

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

ISINISH

Dastur yopilganda o‘yindagi rekordingiz qayerga ketadi?

O‘zgaruvchilar xotirada yashaydi — dastur tugashi bilan yo‘qoladi. Qanday saqlab qolamiz?

1

Kompyuterni o‘chirib yoqsangiz ham qolib ketadigan ma’lumotlarga misol ayting.

2

Faylga “qo‘shib yozish” va “ustidan yozish” farqi nima?

3

Fayl topilmasa yoki unda son o‘rniga matn bo‘lsa, dastur nima qilishi kerak?

O‘QUV MAQSADLARI

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

1. **Tushuntirish** — fayl, oqim va bufer tushunchalarini
2. **Qo‘llash** — ofstream, ifstream, fstream sinflarini va ochilishni tekshirishni
3. **Tanlash** — mos ochish rejimini: out, app, in, trunc, ate, binary
4. **Farqlash** — o‘qish usullarini: >>, getline, get
5. **Oldini olish** — while (!f.eof()) xatosining

01

Fayl va fayl oqimlari

<fstream> · ofstream, ifstream, fstream · ochilishni tekshirish · bufer

ofstream

ifstream

if (!f)

flush

<FSTREAM>

Fayllar — cin/cout kabi oqimlar

Sinf	Vazifasi	O‘xshashi
`ofstream`	Faylga yozish (output file stream)	cout kabi: `f << x;`
`ifstream`	Fayldan o‘qish (input file stream)	cin kabi: `f >> x;`, `getline(f, s);`
`fstream`	Ham o‘qish, ham yozish	Ikkalasi

Har doim tekshiring: `if (!f) { ... }` — fayl ochilmagan bo‘lishi mumkin (yo‘q, ruxsat yo‘q, disk to‘la).

Birinchi fayl

```cpp
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

int main() {
    ofstream yoz("salom.txt");
    yoz << "Salom, fayl!" << endl << 2026 << " " << 3.14
            << endl;
    yoz.close();
    ifstream oqi("salom.txt");
    string qator;
    int n = 0;
    while (getline(oqi, qator))
        cout << ++n << ": " << qator << endl;
    return 0;
}
```

▶ NATIJA

```
1: Salom, fayl!
2: 2026 3.14
```

`ofstream yoz(...)`

fayl yo‘q bo‘lsa yaratiladi

`yoz << ...`

cout kabi yozish

`while (getline(...))`

qatorlar tugaguncha o‘qish

BUFER

Nega ma’lumot darhol diskka tushmaydi?

1

f << x

Ma’lumot avval xotiradagi buferga yig‘iladi

2

Bufer to‘ladi

Yoki `flush()`, `endl`, `close()` chaqirildi

3

Diskka yozish

Bufer mazmuni bir yo‘la diskka o‘tadi

4

Halokat bo‘lsa

Buferda qolgan ma’lumot yo‘qolishi mumkin

Diskka murojaat xotiradan minglab marta sekin — bufer tezlikni oshiradi. `endl` = '\n' + flush.

TUSHUNISHNI TEKSHIRISH

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

1-bo‘lim yakuni: Fayl va fayl oqimlari

1

Faylga yozish uchun qaysi sinf ishlatiladi?

Savol 1

2

Fayl ochilganini qanday tekshirasiz?

Savol 2

3

`endl` va `'\n'` farqi nima?

Savol 3

AMALIY TOPSHIRIQ

Har kim o‘z qurilmasida · 10 daqiqa · Oson · 10 ball

1-topshiriq: Fayldagi sonlar

`sonlar.txt` faylida butun sonlar yozilgan (bo‘sh joy yoki yangi qator bilan). Faylni `ifstream` bilan oching, sonlar sonini va yig‘indisini ekranga chiqaring.

Kirish: `sonlar.txt` fayli: 0…10000 ta butun son ($|son| \le 10^9$).

Chiqish: Ekranga ikki qator: sonlar soni va yig‘indisi.

MISOL · sonlar.txt

```
3 8 -1
10
```

MISOL · CHIQISH

```
4
20
```

Telefon kamerasi bilan skanerlang

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

02

Ochish rejimlari

ios::in, out, app, trunc, ate, binary · | bilan birlashtirish

ios::out	ios::app	ios::in	|

REJIMLAR

Faylni qanday ochamiz?

Rejim	Ma’nosi
`ios::in`	O‘qish (ifstream standarti). Fayl mavjud bo‘lishi shart
`ios::out`	Yozish (ofstream standarti). Bor bo‘lsa — tozalanadi!
`ios::app`	Har bir yozuv oxiriga qo‘shiladi, eski ma’lumot saqlanadi
`ios::trunc`	Ochishda mazmunni o‘chirish
`ios::ate`	Ochilgach pozitsiya oxirida
`ios::binary`	Baytlar o‘zgartirishsiz o‘qiladi/yoziladi

Diqqat: `ofstream f("a.txt");` — mavjud faylni so‘roqsiz tozalaydi. Qo‘shish uchun `ios::app`. Birlashtirish: `ios::in | ios::out`.

APP VA OUT

Qo‘shib yozish va ustidan yozish

```
void chop(const string &n) {
    ifstream f(n); string q;
    while (getline(f, q)) cout << q << "; ";
    cout << endl;
}
int main() {
    { ofstream f("k.txt"); f << "1-kun\n"; }
    { ofstream f("k.txt", ios::app); f << "2-kun\n"; }
    chop("k.txt");
    { ofstream f("k.txt"); f << "yangi\n"; }
    chop("k.txt");
}
```

▶ NATIJA

```
1-kun; 2-kun;
yangi;
```

`{ ... }`

blok tugaganda fayl avtomatik yopiladi

`ios::app`

2-kun oxiriga qo‘shildi

`standart rejim`

keyingi ochishda hammasi o‘chdi

TEKSHIRUV

Mavjud bo‘lmagan fayl

```
ifstream f("yoq.txt");
if (!f.is_open()) cout << "ifstream yaratmaydi" << endl;
fstream g("yoq.txt", ios::in | ios::out);
if (!g) cout << "in|out ham ochmaydi" << endl;
ofstream h("yoq.txt");
if (h) cout << "ofstream yaratdi";
```

▶ NATIJA

```
ifstream yaratmaydi
in|out ham ochmaydi
ofstream yaratdi
```

ifstream

o‘qish uchun fayl mavjud bo‘lishi shart

in | out

ham o‘qish ham yozish — baribir mavjud bo‘lishi kerak

ofstream

yo‘q faylni yaratadi

TUSHUNISHNI TEKSHIRISH

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

2-bo‘lim yakuni: Ochish rejimlari

1

Eski ma’lumotni saqlab, oxiriga yozish rejimi?

Savol 1

2

`ofstream f("a.txt");` mavjud faylga nima qiladi?

Savol 2

3

Ikki rejimni qanday birlashtirasiz?

Savol 3

AMALIY TOPSHIRIQ

Har kim o‘z qurilmasida · 10 daqiqa · Oson · 10 ball

2-topshiriq: Jurnal yakuni (ios::app)

`jurnal.txt` faylida yozuvlar bor (har biri alohida qatorda). Avval faylni o‘qib, bo‘sh bo‘lmagan qatorlarni sanang. So‘ng faylni **qo‘shish rejimida** (`ios::app`) ochib, oxiriga `--- k ta yozuv ---` qatorini qo‘shing. Eski yozuvlar o‘chmasligi kerak.

Kirish: `jurnal.txt` fayli: 0…1000 qator.

Chiqish: Natija — o‘zgargan `jurnal.txt` fayli (eski qatorlar + yakuniy qator).

MISOL · jurnal.txt

```
08:00 dars boshlandi
08:45 tanaffus
09:00 amaliyot
```

MISOL · jurnal.txt

```
08:00 dars boshlandi
08:45 tanaffus
09:00 amaliyot
--- 3 ta yozuv ---
```

Telefon kamerasi bilan skanerlang

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

03

Matnli fayldan o‘qish

>> · getline · get · to‘g‘ri o‘qish sikli · eof() tuzog‘i

f >> x

getline(f, s)

f.get(c)

eof() ✗

UCH USUL

Nimani o‘qiymiz?

Usul	Nima o‘qiydi	Qachon
`f >> x`	Bo‘sh joylarni tashlab, bitta so‘z yoki son	Sonlar, so‘zlar
`getline(f, s)`	Butun qator — '\n' gacha	Gaplar, qator bo‘yicha ishlov
`f.get(c)`	Bitta belgi — bo‘sh joy va '\n' ham	Belgilarni sanash, nusxalash

To‘g‘ri sikl — o‘qishning o‘zini shart qilish: `while (f >> x)`, `while (getline(f, s))`.

KLASSIK XATO

while (!f.eof()) nima uchun noto‘g‘ri?

Noto‘g‘ri: while (!f.eof())

- eof() faqat o‘qish muvaffaqiyatsiz bo‘lgandan keyin rost bo‘ladi
- Sikl bir marta ortiqcha aylanadi
- x eski qiymatda qoladi — oxirgi son ikki marta hisoblanadi
- Fayl “10 20 30\n”: 4 ta son, yig‘indi 90

To‘g‘ri: while (f >> x)

- O‘qish muvaffaqiyatli bo‘lsagina tana bajariladi
- Fayl “10 20 30\n”: 3 ta son, yig‘indi 60
- Xuddi shunday: `while (getline(f, s))`

Qoida: eof() ni sikl sharti sifatida ishlatmang.

Sonlarni o‘qib o‘rtachasini topish

```
{ ofstream f("s.txt"); f << "10 20 30\n"; }
ifstream f("s.txt");
int x, soni = 0, sum = 0;
while (f >> x) {
    soni++;
    sum += x;
}
cout << soni << " ta, o'rtacha " << (double)sum / soni << endl;
ifstream g("s.txt");
char c; int belgi = 0;
while (g.get(c)) belgi++;
cout << belgi << " belgi";
```

▶ NATIJA

```
3 ta, o'rtacha 20
9 belgi
```

`while (f >> x)`

3 marta — to‘g‘ri

`get(c)`

bo‘sh joy va '\n' ham belgi: 8 + 1 = 9

`(double)sum`

butun bo‘lishdan saqlanish

TUSHUNISHNI TEKSHIRISH

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

3-bo‘lim yakuni: Matnli fayldan o‘qish

1

Bo‘sh joylari bilan butun qatorni o‘qish?

Savol 1

2

`while (!f.eof()) { f >> x; ... }` qanday xato beradi?

Savol 2

3

Bo‘sh joyni ham o‘qiydigan usul?

Savol 3

AMALIY TOPSHIRIQ

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

3-topshiriq: Fayl statistikasi

`matn.txt` faylidagi qatorlar, so‘zlar va belgilar sonini toping (Linux’dagi `wc` dasturi kabi). Belgilarga yangi qator belgisi kirmaydi, bo‘sh joylar esa kiradi.

Kirish: `matn.txt` fayli: 1…1000 qator, ASCII matn.

Chiqish: Bitta qatorda uch son: qatorlar, so‘zlar, belgilar.

MISOL · matn.txt

```
Salom dunyo
C++ fayllar bilan ishlash
```

MISOL · CHIQISH

```
2 6 36
```

Telefon kamerasi bilan skanerlang

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

XULOSA

Asosiy xulosalar

1 Oqimlar

`ofstream` — yozish, `ifstream` — o‘qish, `fstream` — ikkalasi; `if (!f)` bilan tekshiring.

2 Rejimlar

Standart out — tozalaydi; `ios::app` — qo‘shadi; `|` bilan birlashtirish.

3 O‘qish

`>>`, `getline`, `get`; sikl — `while (f >> x)`, `eof()` emas.

TAKRORLASH

Yakunlashdan oldin oltita savol

1. Fayldan o‘qish uchun sinf?
2. `ofstream f("x.txt")` mavjud faylni nima qiladi?
3. Oxiriga qo‘shish rejimi?
4. To‘g‘ri o‘qish sikli?
5. `fstream` qanday oqim?
6. Fayl ochilganini qanday tekshiramiz?

CHIQISH CHIPTASI

Darsni tark etishdan oldin

Javoblaringizni qog‘ozga yozing va o‘qituvchiga topshiring.

1

Foydalanuvchi kiritgan 5 ta sonni faylga yozadigan dastur rejasini yozing.

2

`while (!f.eof())` sikli nima uchun xato ekanini tushuntiring.

3

Log faylga yangi yozuv qo‘shish uchun qaysi ochish rejimi kerak?