

**O'ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI VA
KOMMUNIKATSIYALARINI RIVOJLANTIRISH VAZIRLIGI**

TOSHKENT AXBROT TEXNALOGIYALARI UNIVERSITETI

QARSHI FILIALI



KOPYUTER INJINERINGI FAKULTETI

**5330501 Kompyuter injiniring ("Kompyuter injiniring") ta'lim
yo'nalishi**

Dasturiy injinniring kafedrasi

C++ tilida dasturlash fanidan

KURS ISHI

***Mavzu:* Oqimli sinflardan foydalanib dasturlash**

Bajardi:

Hakimov Nasriddin

Qabul qildi:

H.M. Normamatov

Qarshi-2016 y.

**O`ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI VA
KOMMUNIKATSIYALARINI RIVOJLANTIRISH VAZIRLIGI**

TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI

QARSHI FILIALI

KOMPYUTER INJINIRING FAKULTETI

“Tasdiqlayman”

“Dasturiy injinniring”

kafedrasi mudiri: _____

Yo’nalishi : *5330501 Kompyuter injiniring (“Kompyuter injiniring”, “AT -Servis”) ta’lim
yo’nalishi*

Fanning nomi: *C++ da dasturlash.*

KURS ISHI

Kurs ishi mavzusi: *Oqimli sinflardan foydalaniб dasturlash*

Topshirig’i: *Foydalanuvchi структура массивини яратинг. Оқимда (файлда)
уларни сақлаш функциясини яратинг. Дастурда фойдаланинг. Вариантда
кўрсатилган структура массивини файлда ўқиш ва чиқариш
функцияларини яратинг ва дастурда фойдаланинг. Берилган критерий
асосида файлдаги маълумотларни коррективировка (алмашиш) қилувчи
функция яратинг. Дастурни яратинг ва мавжуд бўлган файлни кўриб
чиқинг.*

Амалий топширик:

Yo'llanma materiallar: *Adabiyotlar, internet.*

Chizmachilik qismi: alqoritm, blok sxemalar

Kurs ishini himoya qilish muddati : _____

Guruh: KI-12-14

Talaba: Hakimov Nasriddin

Rahbar: _____ X.M.Нормаматов

Kurs ishini topshirig'ini qabul qildim: _____

TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI

QARSHI FILIALI

KOMPYUTER INJINIRING FAKULTETI

"C++ da dasturlash" fanidan

Kurs ishi bo'yicha topshiriq: Foydalanuvchi struktura massivini yarating. Oqimda (faylda) ularni saqlash funksiyasini yarating. Dasturda foydalaning. Variantda ko'rsatilgan struktura massivini fayldan o'qish va chiqarish funksiyalarini yarating va dasturda foydalaning. Berilgan kriteriy asosida fayldagi ma'lumotlarni korrektirovka (almashish) qiluvchi funksiya yarating. Dasturni yarating va mavjud bo'lgan faylni ko'rib chiqing

Talaba: Hakimov N.

I.Ishning mazmuni: "Standart bibliotekadaqi oqimli sinflar obyektlaridan foydalangan xolda kiritish va chiqarish dasturlashni o'rqnish".

II. Ishni topshirish muddati: _____

III. Mavzu bo'yicha dastlabki ma'lumotlar beruvchi adabiyotlar:

1. «Компьютерлаштиришни янада ривожлантириш ва ахборот-коммуникация технологияларини жорий этиш чора-тадбирлари тўғрисида»ги Ўзбекистон Республикаси Вазирлар Маҳкамасининг Қарори // «Халқ сўзи» газетаси, 2002 йил 8 июнь.
2. Каримов И.А «Эришилган ютуқларни мустаҳкамлаб, янги марралар сари изчил ҳаракат қилишимиз лозим». // «Халқ сўзи», 2006 й., 11-февраль.
3. Nazirov Sh.A., Qobulov R.V., Bobojanov M.R., Raxmanov Q.S. "C va C++ tili." Voris-nashriyot MCHJ, Toshkent 2013, 488 b.
4. Қосимов С.С "Ахборот технологиялари" Техника олий ўрта юртлари бакалавриат босқичи талабалари учун ўқув қўлланма сифатида тавсия этилган. Тошкент шаҳри "Алоқачи" нашриёти, 2006 й.
5. Aripov M., Begalov B., Begimqulov U., Mamarajabov M. Axborot texnologiyalari. Toshkent: Noshir, 2009. -368 b.
6. G'ulomov S.S., Begalov B.A. Informatika va axborot texnologiyalari. Toshkent, Fan, 2010,-686с.
7. Носиров Б. "C++ тилида дастурлаш" фанидан электрон ЎУМ., 1- қисм. Қарши. 2015й.
8. Shukurov O. Qorayev F. "Programmalashdan masalalar to'plami" Qarshi. 2008 y.
9. Ziyonet.uz – Ўзбекистон Республикаси ахборот-таълим портали.
10. acm.tuit.uz – Дастурий ечим тўғрилигини автоматик тестловчитизим.
11. inter-vuz.tuit.uz – Ахборот ресурслари кутубхонаси
12. lib.tuit.uz – ТАТУ электрон кутубхонаси
13. etuit.uz – ТАТУ вертуал таълим тизими
14. dls.tuitkf.uz – ТАТУ Қарши филиали масофавий таълим тизими

IV. Tushuntirish yozilmasining tuzilishi:

1. Kirish qismi.
2. Asosiy qismi.
3. Xulosalar.

Maslahatchilar: F.Q.Qorayev, I.S. Almanov, Sh.A. Murtozayev, R.N. Abdullayev, A.Dusayarov, J.E. Sherov, Ch. M. Xidirova, D.Nurjabova, A.Sh. Xudoynazarov, J.Saidov

Kurs ishi rahbari: _____ **H. M. Normamatov**

I BOB. MURAKKAB MA'LUMOTLAR TURLAR. STRUKTURALAR.....

- 1.1. Sinf haqida.
- 1.2. Strukturalar va massivlar.

II.BOB OQIMLI SINFLAR

- 2.1. Oqimli sinflar ierarxiyasi.
- 2.2. Oqimli sinflar metodlari.
- 2.3. Formatlash.
- 2.4. Matnli oqimlar.
- 2.5. Fayllar bilan ishlash

III AMALIY MASALA

- 3.1. Masalaning quyilishi va tahlili
- 3.2. Algoritm blok-sxemasi
- 3.3. Algoritm dasturiy kodi va natijalar
- 3.4 Hulosa
- 3.5 Foydalanilgan adabiyo

Sinf sifatida ma'lumotlar to'plami va ularga boqlangan funktsiyalarni qarash mumkin. Sinflar C++ tili ichki imkoniyatlarini kengaytirib dasturchi oldida amaliyotning murakkab masalalarini echishda ancha qulaylik yaratadi.

C++ tilida inkapsulyatsiya printsipi sinf deb ataluvchi nostandart tiplarni(foydalanuvchi tiplarini) qosil qilish orqali qimoya qilinadi. To'qri aniqlangan sinf ob'ektini butun dasturiy modul sifatida ishlatish mumkin. qaqiqiy sinfning barcha ichki ishlari yashirin bo'lishi lozim. To'qri aniqlangan sinfning foydalanuvchilari uning qanday ishlashini bilishi shart emas, ular sinf qanday vazifani bajarishini bilsalar etarlidir.

C++ tili ob'ektga mo'ljallangan dasturlash printsiplarini qo'llab quvvatlaydi. Bu printsiplar quyidagilardir: inkapsulyatsiya, merosxo'rlik, polimorfizm

Inkapsulyatsiya. Agarda muqandis ishlab chiqarish jarayonida rezistorni qo'llasa, u buni yangidan ixtiro qilmaydi, omborga (magazinga) borib mos parametrlarga muvofiq kerakli detalni tanlaydi. Bu qolda muqandis joriy rezistor qanday tuzilganligiga e'tiborini qaratmaydi, rezistor faqatgina zavod xarakteristikalariga muvofiq ishlasa etarlidir. Aynan shu tashqi konstruktsiyada qo'llaniladigan yashirinlik yoki ob'ektni yashirinligi yoki avtonomligi xossasi inkapsulyatsiya deyiladi.

Inkapsulyatsiya yordamida berilganlarni yashirish ta'minlanadi. Bu juda yaxshi xarakteristika bo'lib foydalanuvchi o'zi ishlatayotgan ob'ektning ichki ishlari qaqida umuman o'ylamaydi. qaqiqatan qam, xolodilnikni ishlatishda refrijektorni ishlash printsipini bilish shart emas. Yaxshi ishlab chiqilgan dastur ob'ektini qo'llashda uning ichki o'zgaruvchilarining o'zaro munosabati qaqida qayqurish zarur emas.

Yana bir marta takrorlash joizki, rezistorni samarali qo'llash uchun uning ishlash printsipi va ichki qurilmalari qaqidagi ma'lumotlarni bilish umuman shart emas. Rezistorning barcha xususiyatlari inkapsulyatsiya qilingan, ya'ni yashirilgan. Rezistor faqatgina o'z funktsiyasini bajarishi etarlidir.

C++ tilida inkapsulyatsiya printsipli sinf deb ataluvchi nostandart tiplarni(foydalanuvchi tiplarini) qosil qilish orqali qimoya qilinadi.

Sinflar qanday tuzilishga ega ekanligi bilan keyinroq tanishib chiqamiz.

To'qri aniqlangan sinf ob'ektini butun dasturiy modul sifatida ishlatish mumkin. qaqiqiy sinfnig barcha ichki ishlari yashirin bo'lishi lozim. To'qri aniqlangan sinfnig foydalanuvchilari uning qanday ishlashini bilishi shart emas, ular sinf qanday vazifani bajarishini bilsalar etarlidir.

Merosxo'rlik. Acme Motors kompaniyasi injenerlari yangi avtomobil konstruktsiyasini yaratishga aqd qilishsa, ular ikkita variantdan birini tanlashlari lozim. Birinchisi, avtomobilning konstruktsiyasini boshidan boshlab yangidan ixtiro qilish, ikkinchisi esa mavjud Star modelini o'zgartirishdir. Star modeli qariyb ideal, faqatgina unga turbokompressor va olti tezlanishli uzatma qo'shish lozim. Bosh muqandisikkinchi variantni tanladi. Ya'ni noldan boshlab qurishni emas, balki Star avtomobiliga ozgina o'zgartirish qilish orqali yaratishni tanladi. Uni yangi imkoniyatlar bilan rivojlantirmoqchi bo'ldi. Shuning uchun, yangi modelni Quasar deb nomlashni taklif qildi. Quasar-Star modeliga yangi detallarni qo'shish orqali yaratilgan.

C++ tili qam shunday merosxo'rlikni qimoya qiladi. Bu yangi berilganlar tipi (sinf), oldindan mavjud bo'lgan sinfni kengaytirishdan qosil bo'ladi. Bunda yangi sinf oldingi sinfnig merosxo'ri deb ataladi.

Polimorfizm. Akselatorni bosilishida Star modeliga nisbatan yangi yaratilgan Quasar modelida boshqacharoq amallar bajarilishi mumkin. Quasar modelida dvigatelga yoqilqini sepuvchi injektor sistemasi va Star modelidagi korbyurator o'rniga turbokompressor o'rnatilgan bo'lishi mumkin. Lekin foydalanuvchi bu farqlarni bilishi shart emas. U rulga o'tirgach oddiygina akselatorni bosadi va avtomobilning mos reaksiyasini kutadi.

C++ tili bir xil nomdagi funktsiya turli ob'ekt tomonidan ishlatilganda turli amallarni bajarishi imkoniyatini ta'minlaydi. Bu funktsiya va sinfnig polimorfligi

deb nomlanadi. Poli – ko'p, morfe – shakl degan ma'noni anglatadi. Polimorfizm – bu shaklning ko'p xilligidir. Bu tushunchalar bilan keyinchalik batafsil tanishamiz.

Sinflar va sinf a'zolari. Yangi tip sinfni e'lon qilish bilan tuziladi. Sinf - bu bir – biri bilan funktsional orqali boqlangan o'zgaruvchilar va metodlar to'plamidir. Sinflarga amaliyotdan ko'pgina misollar keltirish mumkin. Masalan, avtomobilni qildirak, eshik, o'rindiqlik, oyna va boshqa qismlardan tashkil topgan kolleksiya yoki qaydash tezligini oshirish, to'xtatish, burish imkoniyatlariga ega bo'lgan ob'ekt deb tasavvur qilish mumkin. Avtomobil o'zida turli eqtiyot qismlarni va ularni funktsiyalarini inkapsulyatsiya qiladi. Avtomobil kabi sinfda qam inkapsulyatsiya qator imkoniyatlarni beradi. Barcha ma'lumotlar bitta ob'ektda yiqilgan va ularga osongina murojaat qilish, ularni o'zgartirish va ko'chirish mumkin. Sizning sinfingiz bilan ishlovchi dasturiy qismlar, ya'ni mijozlar sizning ob'ektingizdan, uning qanday ishlashidan tashvishlanmasdan, bema'lol foydalanishlari mumkin.

Sinf o'zgaruvchilarning ixtiyoriy kombinatsiyasidan, shuningdek boshqa sinflar tiplaridan iborat bo'lishi mumkin. Sinfdagi o'zgaruvchilar o'zgaruvchi – a'zolar yoki xossalar deyiladi. Sar sinfi o'rindiqlik, radiopriyomnik, shina va boshqa o'zgaruvchi - a'zolardan iborat. O'zgaruvchi – a'zolar faqatgina o'zlarining sinflarida yotadilar. qildirak va motor avtomobilning qanday tarkibiy qismi bo'lsa, o'zgaruvchi – a'zolar qam sinfning shunday tarkibiy qismidir.

Sinfdagi funktsiyalar odatda o'zgaruvchi a'zolar ustida biror bir amal bajaradilar. Ular funktsiya – a'zolar yoki sinf metodlari deb aytiladi. Mashina sinfi metodlari qatoriga Haydash() va Tuxtatish() metodlari kiradi. Mushuk sinfi qayvonni yoshi va oqirligini ifodalovchi o'zgaruvchi – a'zolarga ega bo'lishi mumkin. Shuningdek, bu sinfning funktsional qismi Uxlash(), Miyovlash(), SichqonTutish() metodlaridan ibo/rat bo'ladi.

Funktsiya – a'zolar qam o'zgaruvchi a'zolar singari sinfda yotadi. Ular o'zgaruvchi a'zolar ustida amallar bajaradi va sinfni funktsional imkoniyatlarini aniqlaydi.

Sinflar orasidagi munosabatlar. Murakkab sinflarni qosil qilishda oldin uni tashkil etuvchi oddiyroq sinflarni e'lon qilib, keyin esa ularni birlashtirish orqali sinfni qosil qilish maqsadga muvofiqdir. Masalan, qildirak sinfi, motor sinfi, uzatish korobkasi sinfi va boshqa sinflarni qosil qilib, keyin esa ularni birlashtirish orqali avtomobil sinfini qurish oldimizga turgan masalani echishni ancha osonlashtiradi.

Ikkinchi misolni ko'rib chiqamiz. To'qri to'rtburchak chiziqlardan tashkil topgan. Chiziq esa ikki nuqta orqali aniqlanadi. qar bir nuqta x va u koordinatalar yordamida aniqlanadi. To'qri to'rtburchak to'rtta nuqtani birlashtiruvchi to'rtta chiziq yordamida aniqlanadi. Bunda qar bir nuqta grafikda koordinatalarga ega bo'ladi. Shuning uchun oldin qar bir nuqtaning x, u koordinatalarini saqlash uchun Nuqta sinfi e'lon qilingan..

2.Sinf va obyektning e'lon qilinishi

Sinfni e'lon qilish. Sinfni e'lon qilish uchun class kalitli so'zi, undan so'ng ochiluvchi figurali qavs, co'ng xossalar va metodlari ro'yxati ishlatiladi. Sinfni e'lon qilish yopiluvchi figurali qavs va nuqtali vergul orqali yakunlanadi. Masalan, Mushuk sinfini quyidagicha e'lon qilish mumkin.

```
Class Mushuk  
  
{  
  
    unsigned int itsYosh ;  
  
    unsigned int itsOgirlik ;  
  
    void Miyovlash()  
  
}
```

Mushuk sinfini e'lon qilishda xotira zaqiralanmaydi. E'lon qilish, kompilyatorga Mushuk sinfini mavjudligini, qamda unda qanday qiymatlar saqlashi mumkinligi (itsYosh, itsOgirlik) va u qanday amallarni bajarishi mumkinligi (Miyovlash() metodi) qaqida xabar beradi. Bundan tashqari, bu e'lon qilish orqali kompilyatorga Mushuk sinfining o'lchami, ya'ni qar bir Mushuk sinfi ob'ekti uchun kompilyator qancha joy ajratishi lozimligi qaqida qam ma'lumot beradi. Masalan,

joriy misolda butun qiymat uchun to'rt bayt talab qilinsa, Mushuk sinfi ob'ekti o'lchovi sakkiz bayt bo'ladi. (itsYosh o'zgaruvchisi uchun to'rt bayt, itsOgirlik o'zgaruvchisi uchun to'rt bayt). Miyovlash() metodi xotiradan joy ajratishni talab qilmaydi.

Oqimli sinflar ierarxiyasi

C++da oqimli sinflar kutubxonasi ikkita asosiy ios va streambuf sinflar asosida tuzilgan, streambuf sinfi kiritish-chiqarish fizik qurilmalari bilan xotirada joylashgan kiritish-chiqarish buferlarni o'zaro bo'g'lanishini va tashkilini ta'minlaydi. Streambuf sinfining metodlarini va ma'lumotlarini dasturchi ochiq ishlatmaydi. Mavjud bo'lgan sinflar asosida yangi sinflarni yaratishda dasturchiga xam sinfga murojaat etish ruxat etilgan. **ios** sinfi formal ***kiritish chiqarish*** va ***xatolarni tekshirish*** vositalarga ega.

- Standart oqimlar (istream, ostream, iostream) terminal bilan ishlash uchun xizmat qiladi.
- Satrli oqimlar (istrstream, ostrstream, strstream) xotirada joylashtirilgan satrli buferlardan kiritish-chiqarish uchun xizmat qiladi.
- Faylli oqimlar(ifstream, ofstream, fstream) fayllar bilan ishlash uchun xizmat qiladi.

Oqimli sinflar, ularning metodlari va ma'lumotlari dasturda murojaat etish ruxsatiga ega bo'ladi, qachonki unga kerakli bosh fayl kiritilgan bo'lsa.

- iostream.h – ios, ostream, istream uchun.
- strstream.h – strstream, istrstream, ostrstream uchun
- fstream.h – fstream, ifstream, ofstream uchun

Quyidagi ob'ekt-oqimlar dasturda main funksiyasini chaqirish oldidan avvaldan aniqlangan va ochilgan bo'ladi:

- extern istream cin; //Klaviaturadan kiritish standart oqimi

- `extern ostream cout; //`Ekranga chiqarish standart oqimi
- `extern ostream cerr; //`Xatolar xaqidagi xabar chiqarish standart oqimi.

Oqimli sinflar metodlari

Oqimdan qiritish uchun `istream` sinfdagi ob'ektlar ishlatiladi, oqimga chiqarish uchun - `ostream` sinfdagi ob'ektlar.

`istream` sinfda quyidagi funksiyalar tavsiflangan:

- `istream get (char& S)` - `istream` dan `S` ga simvolni o'qiydi. Xato xolatida `S` `OXFF` qiymatini oladi.
- `int get()` - `istream` dan keyingi simvolni chiqaradi. Faylni oxirini aniqlagach `EOF`ni qaytaradi.
- `istream& get(char* buffer,int size,char delimiter='\n')` - Bu funksiya `istream`dan simvollarni chiqaradi va ularni buferga nusxalaydi. Operatsiya yoki faylning oxiriga yetganda, yoki `size` fayllardan nusxa olgan jarayonda, yoki ko'rsatilgan ajratuvchini aniqlaganda to'xtaydi. Ajratuvchi esa nusxalanmaydi va `streambuf` qoladi. O'qib bulingan simvollar ketma-ketligi xardoim nul simvol bilan tugatiladi.
- `istream& getline(char* buffer,int size, char delimiter='\n')` - Ajratuvchi oqimdan chiqariladi, lekin, buferga kiritilmaydi. Bu esa satrlarni oqimdan chiqaruvchi asosiy funksiya. O'qib chiqilgan simvollar nul simvoli bilan ta'momlanadi.
- `istream& read(char* buffer,int size)` - Ajratuvchilarni qo'llanmaydi va buferga o'qilgan simvollar nul simvoli bilan tugamaydi.
- `int peek()` - `istream` dan simvolni chiqarmasdan `istream`ga qaytaradi.
- `int gcount()` - Formatlanmagan oxirgi kiritish operatsiyasi vaqtida o'qilgan simvollar sonini qaytaradi.

- `istream& putback(S)` - Agar get doirasidagi streambuf ob'ektida bo'sh fazo mavjud bo'lsa, unda o'sha yerga S simvoli joylashtiriladi.
 - `istream& ignore(int count=1,int target=EOF)` - Quyidagilar bajarilmaguncha istream dan simvol chiqarilaveradi:
 - funksiya count simvollarni chiqarmaguncha;
 - target simvoli aniqlanmaguncha;
 - faylni oxiriga yetmaguncha.
 - ostream sinfida quyidagi funksiyalar tavsiflangan:
 - `ostream& put(char C)` - ostream ga S simvolni joylashtiradi.
 - `ostream& write(const char* buffer,int size)` - Buferda mavjudlarni ostreamga yozadi. Xatoga duch kelmaguncha yoki size simvollarni nusxasi olmaguncha simvollarni nusxasi olinadi. Bufer formatlanmasdan yoziladi. Nol simvollarga ishlov berish boshqa ishlov berishlardan farq qilmaydi. Quyidagi funksiya ishlov berilmagan (binar va matnli) ma'lumotlarni ostreamga uzatadi.
 - `ostream& flush()` - streambuf buferni olib tashlaydi.
- Bu funksiyalardan tashqari istream sinfda `>>`, ostream sinfda esa `<<` operatsiyalar qayta yuklangan. `<<` va `>>` operatsiyalar ikkita operandga ega. Chap operandi – bu istream (ostream) sinfning ob'ekti, o'ng operandi esa – bu dasturlash tilida ko'rsatilgan tipdagi ma'lumot.

Misol:

```
char ch, next, lookahead;
while ( cin.get( ch ))
{ switch (ch) {
  case '/':
    // izoxni peek() yordamida tekshirish
    // agar xa bo'lsa satr qolganini o'tkazish
    next = cin.peek();
    if ( next == '/' ) cin.ignore( lineSize, '\n' );
    break;
  case '>':
```

```
// leksemaga tekshirish >>=  
next = cin.peek();  
if ( next == '>' ) {  
    lookahead = cin.get();  
    next = cin.peek();  
    if ( next != '=' ) cin.putback( lookahead );  
}
```

Formatlash

Ushbu ma'lumotlar uchun cout, cin, cerr, clog standart potoklarga kiritish << va chiqarish >> operatsiyalarni to'g'ridan to'g'ri qo'llash qayta uzatish qiymatlarni tashki tavsiflash aytib o'tilmagan formatlardan foydalanishga olib keladi.

Chiqaruvchi axborotni tavsiflash formatlari va ma'lumotlarni qiritishda qabul qilish qoidalarini dasturlovchi orqali formatlash bayroqlari yordamida o'zgartiriladi. Bu bayroqlar ios bazaviy sinfdagi xamma oqimlardan meros bo'lgan. Formatlash bayroqlari aloxida qayd etilgan bitlar ko'rinishida amalga oshirilgan va long x_flags sinfnining protected komponentasida saqlanadi. Ularga murojaat etish uchun tegishli public funksiyalar mavjud.

Formatlash bayroqlardan tashqari ios sinfnining kuydagi protected komponentalari ishlatiladi:

int x_width – chiqarish maydonning minimal eni.

int x_precision – qiritishda xaqiqiy sonlarning tavsiflash aniqligi (kasr qisimning raqamlar soni);

int x_fill – chiqarishda to'ldiruvchi simvol, probel – ko'rsatilmagan xolda.

Ushbu maydonlarni qiymatlarini olish (o'rnatish) uchun quyidagi funksiyalar komponentalari ishlatiladi:

- `int width();`
- `int width(int);`
- `int precision();`
- `int precision(int);`
- `char fill();`
- `char fill(char);`

Agar bir marta `cout.fill`, yordamida to'ldiruvchi simvol tanlansa `cout.fill` qayta chaqirilmaguncha o'zgarmaydi.

Manipulyatorlar

Manipulyatorlar - oqim ishini modifikatsiyalashini imkon etuvchi maxsus funksiyalar. Manipulyatorlarning xususiyati shundaki, ularni `>>` yoki `<<` operatsiyalarning o'ng operand sifatida foydalanish mumkin. Chap operand sifatida esa xar doimgidak oqim (oqimga ilova) ishlatiladi, va xudda shu oqimga manipulyator ta'sir etadi.

`endl` Yangi qator simvolini qo'yib, bufer ostream bo'shatish

- `dec` O'nlik sanoq tizimida chiqarish (ko'zda tutilgan)
- `hex` O'n oltilik sanoq tizimida chiqarish
- `oct` Sakkizlik sanoq tizimida chiqarish

- `ws` Bo'shlik belgisini o'tkazish
quyidagi manipulyatorlar uchun `#include <iomanip>` ulash talab etiladi
- `setfill( ch )` `ch` simvol bilan bo'sh joyni to'ldirish
- `setprecision( n )` `n` ga teng bo'lgan suzuvchi nuqtali sonni chiqarish aniqliligini o'rnatish
- `setw( w )` `w` ga teng bo'lgan qiritish yoki chiqarish maydonning enini o'rnatish
- `setbase( b )` `b` asosiga ega bo'lgan butun sonlarning chiqarish

Fayllar bilan ishlash

Fayllarni ochish va yopish

C++da fayllar bilan ishlash `fstream` kutubxonasidagi biron-bir sinflar yordamida amalga oshiriladi.

Ftsream kutubxonasi fayllarni o'qib olish uchun javob beradigan `ifstream` sinfiga, xamda faylga axobotning yozib olinishiga javob beradigan `ofstream` sinfiga ega.

Biron-bir faylni yozish yoki o'qish uchun ochish uchun, `ofstream` turdagi yoki mos xolda `ifstream` turdagi o'zgaruvchini yaratish kerak. Bunday o'zgaruvchini initsiallashtirishda fayl nomi o'zgaruvchi nomidan keyin qavs ichida berilgan belgilar massivi ko'rinishida uzatiladi.

Masalan, C diskida joylashgan 'text.txt' faylini ochish kerak. Buning uchun kodning quyidagi fragmenti qo'llanadi:

```
ifstream ifl ("C:\text.txt");  
  
ofstream ofl("C:\text.txt");  
  
char s[20]="C:\text.txt";  
  
ifstream ifJ (s);
```

Bu yerda `ifl`, `ifl` va `ofl` - o'zgaruvchilar nomi bo'lib, ular orqali fayl bilan ma'lumotlarni ayirboshlash amalga oshiriladi. Agar fayl xam dasturning bajarilayotgan fayli joylashtirilgan papkada bo'lsa, u xolda faylning nomi to'liq ko'rsatilmayligi mumkin (faqat fayl nomi, unga borish yo'lisiz). Bundan tashqari fayl nomini to'g'ridan-to'ri ko'rsatish o'rniga, uning nomidan iborat belgilar massivlarini ko'rsatish mumkin.

Faylga yozish

Axborotni faylga yozish uchun `put` komandasidan foydalanish mumkin. Bu komanda orqali standart turdagi yakka o'zgaruvchi yoki biron-bir belgilar massivi

uzatiladi. Belgilar massivi uzatilgan xolda xam massivdagi belgilar sonini uzatish kerak.

Bundan tashqari "<<"operatoridan foydalanish mumkin. Bu operatoridan kodning bitta satrida turli turdagi qiymatlarni uzatgan xolda ko'p martalab foydalanish mumkin. Satr xaqida gap ketganda, chiqarish satr oxiri belgisi, ya'ni '\n' paydo bo'lishidan oldin amalga oshiriladi. Belgisiz turga ega bo'lgan barcha o'zgaruvchilar oldin belgilarga o'zgartirib olinadi.

```
ifstream ofl ("C:\text.txt");

char a='M';

ofl.put(s);

char s[9]="The text";

ofl.put(s,9);

ofl<<"The text";

int i=100;

ofl<<i;

char ss[]="The text";

int k=200;

ofl<<"The text"<<k<<ss<<200;
```

Fayldan o'qish

Axborotni fayldan o'qib olish uchun ">>" operatoriga ekvivalent bo'lgan get funksiyasi qo'llanadi. Put funksiyasi kabi, get funksiyasi xam xar qanday o'zgaruvchilarning standart turlari yoki / va belgilar massivlari bilan ishlay oladi. Shuningdek get ga xar jixatdan ekvivalent bo'lgan getline funksiyasi mavjud: farqi faqat shundaki, getline funksiyasi satr oxiridan oxirgi belgini qaytarmaydi.

```
ifstream ofl ("C:\text.txt");
```

```
char s; char ss[9];
```

```
s=ofl.get ();
```

```
cout<<s;
```

```
ofl.get(s);
```

```
cout<<s;
```

```
ofl.getline(ss,9);
```

```
cout<<ss;
```

```
ofl>>ss;
```

```
cout<<ss;
```

Fayl oxirini aniqlash

Fayl ichidagisini, fayl oxiri uchramaguncha, o'qish dasturdagi oddiy fayl operatsiyasi xisoblanadi. Fayl oxirini aniqlash uchun, dasturlar oqim ob'ektining eof funksiyasidan foydalanishlari mumkin. Agar fayl oxiri xali uchramagan bo'lsa, bu funksiya 0 qiymatini qaytarib beradi, agar fayl oxiri uchrasa, 1 qiymatini qaytaradi. While tsiklidan foydalanib, dasturlar, fayl oxirini topmaganlaricha, qo'yida ko'rsatilganidek, uning ichidagilarini uzluksiz o'qishlari mumkin:

```
while (! Input_file.eof())  
  
    {  
  
        //Operatorlar  
  
    }
```

Ushbu xolda dastur, eof funksiyasi yolg'on (0) ni qaytarguncha, tsiklni bajarishda davom etadi.

Xuddi shunday, keyingi dastur - WORD_EOF.CPP fayl ichidagisini bitta so'z bo'yicha bir martada, fayl oxiri uchramaguncha, o'qiydi:

```
#include <iostream>  
  
#include <fstream.h>  
  
void main(void)  
  
{  
  
    ifstream input_file("BOOKINFO.DAT");  
  
    char word[64] ;  
  
    while (! input_file.eof())  
  
    {  
  
        input_file >> word;  
  
        cout << word << endl;  
  
    }  
  
}
```

Fayllar bilan ishlashda xatolarni aniqlash

Xatolarni kuzatib borishda dasturlarga yordam berish uchun, fayl ob'ekting fail funksiyasidan foydalanish mumkin. Agar fayl operatsiyasi jarayonida xatolar bo'lmagan bo'lsa, funksiya yolg'on (0) ni qaytaradi. Biroq, agar xato uchrasa, fail funksiyasi xaqiqatni qaytaradi. Masalan, agar dastur fayl ochadigan bo'lsa, u, xatoga yo'l qo'yilganini aniqlash uchun, fail funksiyasidan foydalanishi kerak. Bu quyida shunday ko'rsatilgan:

```
ifstream input_file("FILENAME.DAT");

if (input_file.fail())

{

    cerr << "Ochilish xatosi FILENAME.EXT" << endl;

    exit(1);

}
```

TEST_ALL.CPP dasturi turli xato vaziyatlarni tekshirish uchun fail funksiyasidan foydalanadi:

```
#include <iostream>

#include <fstream.h>

void main(void)

{

    char line[256] ;

    ifstream input_file("BOOKINFO.DAT");

    if (input_file.fail()) cerr << "Ochilish xatosi BOOKINFO.DAT" << endl;

    else

    {

        while ((! input_file.eof()) && (! input_file.fail()))
```

```
{  
  
    input_file.getline(line, sizeof(line)) ;  
  
    if (! input_file.fail()) cout << line << endl;    } }  
}
```

Faylning kerak bo'lmay qolganda berkitilishi

Dasturni tugallash uchun operatsiya tizimi o'zi ochgan fayllarni berkitadi. Biroq, odatga ko'ra, agar dasturga fayl kerak bo'lmay qolsa, uni berkitishi kerak. Faylni berkitish uchun dastur, quyida ko'rsatilganidek, dastur close funksiyasidan foydalanishi kerak:

```
input_file.close ();
```

Faylni yopayotganingizda, dastur ushbu faylga yozib olgan barcha ma'lumotlar diskka tashlanadi va ushbu fayl uchun katalogdagi yozuv **yangilanadi**.

O'qish va yozish operatsiyalarining bajarilishi

Xozirga qadar gap borayotgan dasturlar belgili satrlar utsida operatsiyalar bajarar edi. Dasturlaringiz murakkablashgan sari, extimol, sizga massivlar va tuzilmalarni o'qish va yozish kerak bo'lib qolar. Buning uchun dasturlar read va write funksiyalaridan foydalanishlari mumkin. read va write funksiyalaridan foydalanishda ma'lumotlar o'qiladigan yoki yozib olinadigan ma'lumotlar buferini, shuningdek buferning baytlarda o'lchanadigan uzunligini ko'rsatish lozim. Bu quyida ko'rsatilganidek amalga oshiriladi:

```
input_file.read(buffer, sizeof(buffer));

output_file.write(buffer, sizeof(buffer));
```

Masalan, STRU_OUT.CPP dasturi tuzilma ichidagisini EMPLOYEE.DAT fayliga chiqarish uchun, write funksiyasidan foydalanadi:

```
struct employee

{

    char name[64];

    int age;

    float salary;

} worker = { "Djon Doy", 33, 25000.0 };

ofstream emp_file("EMPLOYEE.DAT");

emp_file.write((char *) &worker, sizeof(employee));

}
```

Odatda write funksiyasi belgilar satriga ko'rsatkich oladi. Xuddi shunday tarzda STRU_IN.CPP dasturi read metodidan xizmatchi xaqidagi axborotni fayldan o'qib olish uchun foydalanadi:

```
ifstream emp_file("EMPLOYEE.DAT");

emp_file.read((char *) &worker, sizeof(employee));

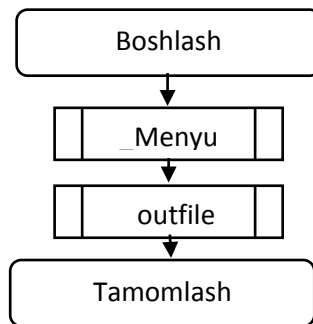
cout << worker.name << endl;

cout << worker.age << endl;

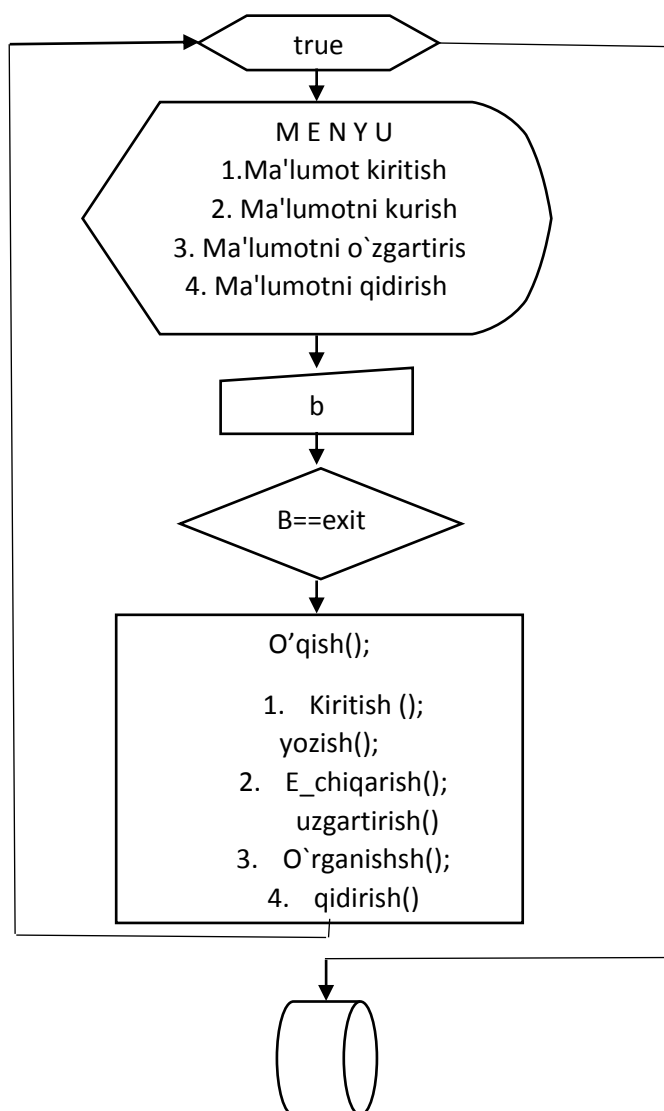
cout << worker.salary << endl;
```

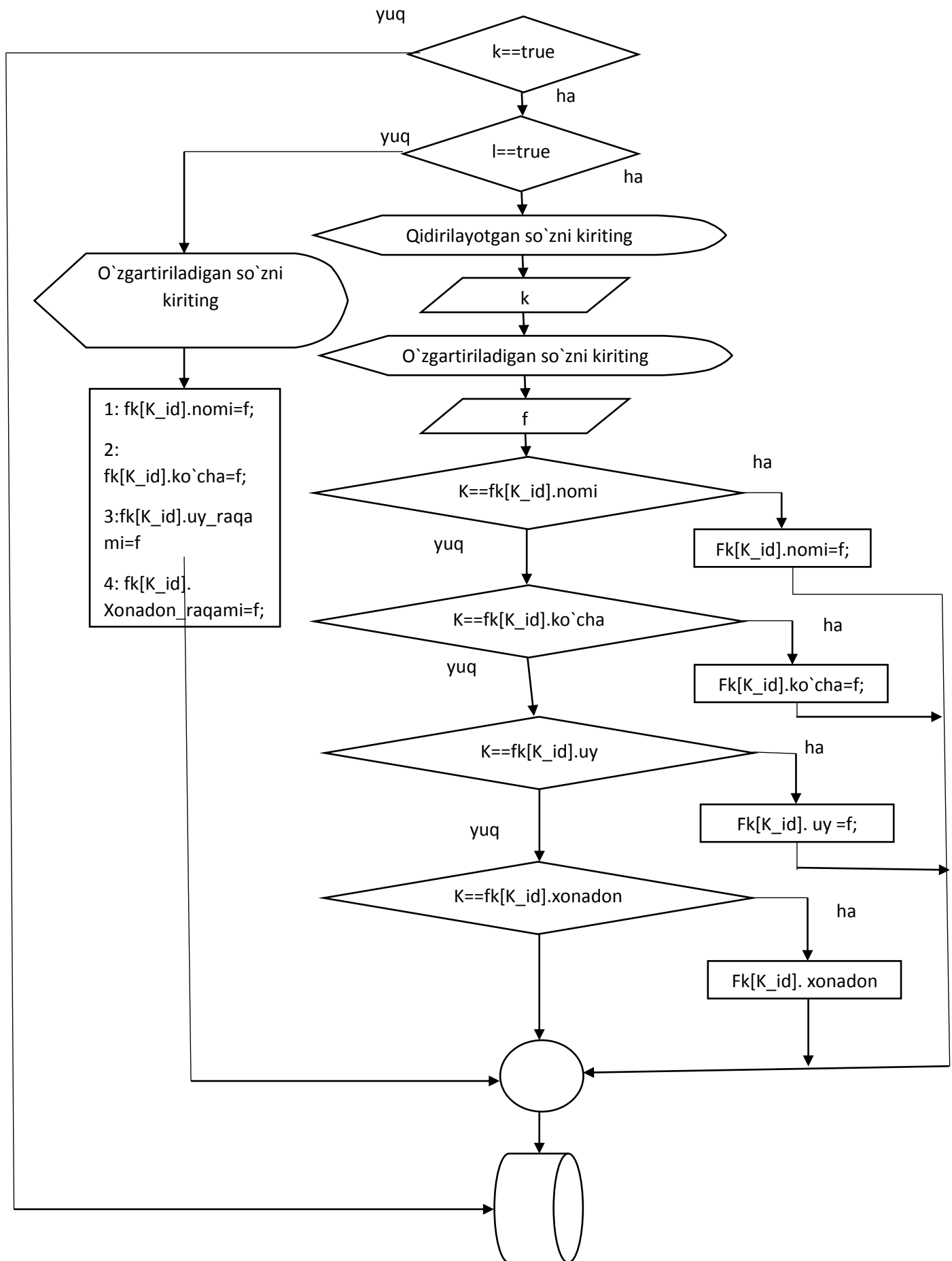
Algoritm

Asosiy Funksiya

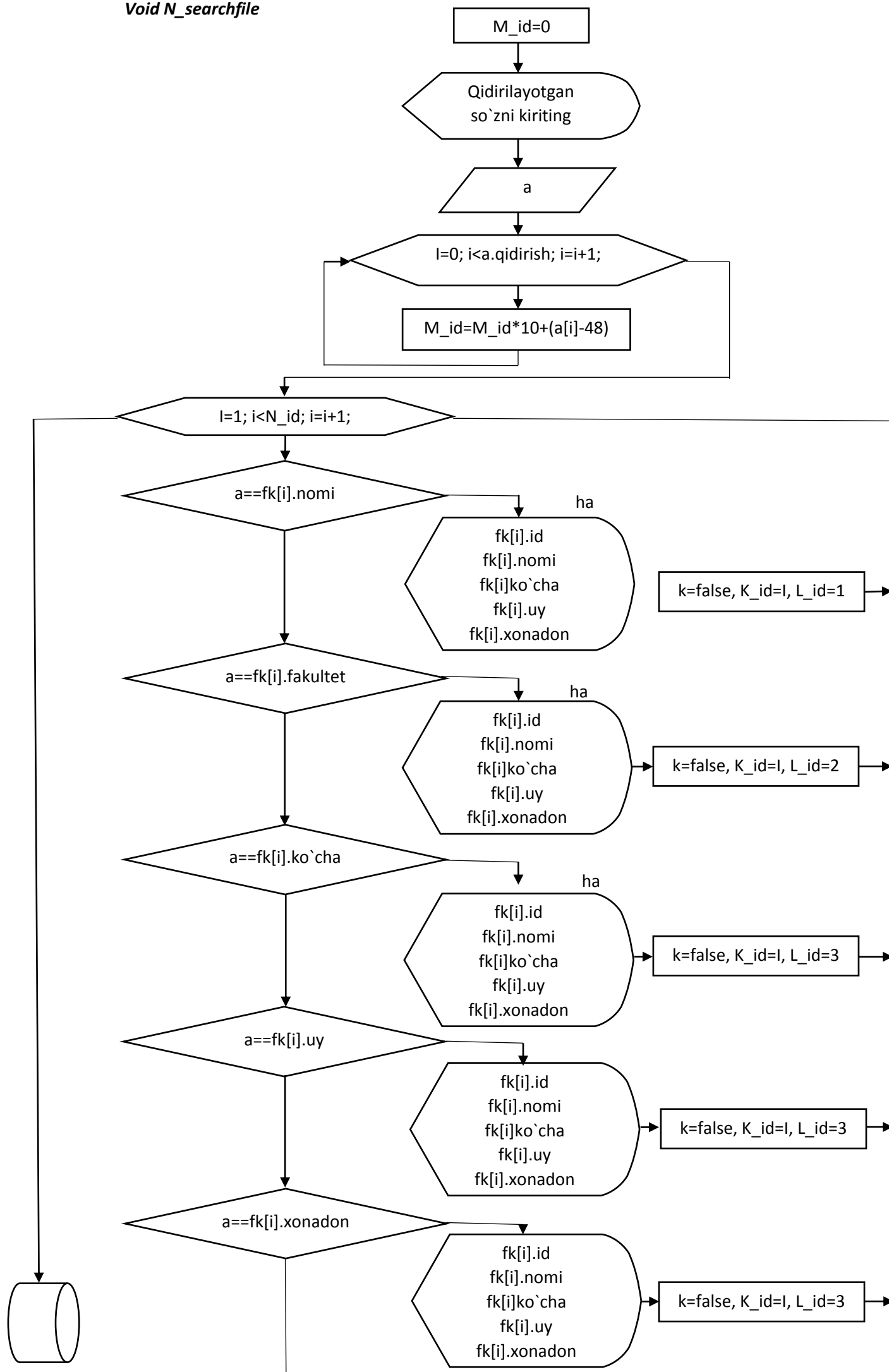


N_Menyu funksiya





Void N_searchfile



III AMALIY MASALA

```
#include<iostream>
```

```
#include<fstream>
```

```
using namespace std;
```

```
void qidirish();
```

```
void uzgartirish();
```

```
void menyu();
```

```
void kiritish();
```

```
void yozish();
```

```
void oqish();
```

```
void E_chiqarish();
```

```
struct Manzil
```

```
{
```

```
string nomi, kucha, uy_raqami, xonadon_raqami;
```

```
int id;
```

```
};
```

```
int z=1, x, y;
```

```
haven A[10];
```

```
bool k=true, l=false;
```

```
int main()
```

```
{
```

```
    menu();
```

```
    write();
```

```
    return 0;
```

```
}
```

```
// Menyu funksiyasi
```

```
void menyu()
```

```
{
```

```
    string b;
```

```
    while(true)
```

```
    {
```

```
        cout<<"\t\t\t H E V E N \n";
```

```
        cout<<"\t1. Ma'lumot kiritish uchun:\n";
```

```
        cout<<"\t2. Ma'lumotni kurish uchun:\n";
```

```
        cout<<"\t3. Ma'lumotni o`zgartirish uchun:\n";
```

```
        cout<<"\t4. Ma'lumotni qidirish uchun:\n";
```

```
        cin>>b; cout<<"\n_____ \n\n";
```

```
        if(b=="CHIQISH" || b=="chiqish" || b=="Chiqish") break;
```

```
    else
```

```
    {
```

```
        oqish();
```

```
    char g=b[0]; switch(g)
```

```
    {
```

```
        case '1': kiritish(); yozish(); break;
```

```
        case '2': E_chiqarish(); break;
```

```
        case '3': qidirish(); uzgartirish(); yozish(); break;
```

```
case '4': qidirish(); break;
```

```
default: cout<<"\tXato !!!\n
```

```
\n\n"; break;
```

```
}
```

```
}
```

```
}
```

```
};
```

```
//O`zgartirish funksiyasi
```

```
void uzgartirish()
```

```
{
```

```
if(!k)
```

```
{
```

```
string f, k;
```

```
if(l)
```

```
{
```

```
cout<<"Qidirilayotgan so`zni kiriting: "; cin>>k;
```

```
cout<<"O`zgartiriladigan so`zni kiriting: "; cin>>f;
```

```
if(k==A[x].nomi) A[x].nomi=f;
```

```
else if(k==A[x].kucha) A[x].kucha=f;
```

```
else if(k==A[x].uy_raqami) A[x].uy_raqami=f;
```

```
else if(k==A[x].xonadon_raqami) A[x].xonadon_raqami=f;
```

```
}
```

```
else
```

```
{
```

```
cout<<"O`zgartiriladigan so`zni kiriting: "; cin>>f;
```

```
switch(y)
```

```
{
```

```

case 1: A[x].nomi=f; break;
case 2: A[x].kucha=f;
case 3: A[x].uy_raqami=f;
case 4: A[x].xonadon_raqami=f;
}
}
}
};

```

// Qidirish tizimi

```
void qidirish()
```

```

{
string a;
int M_id=0;
cout<<"Qidirilayotgan so`zni kiriting: "; cin>>a;
for(int i=0; i<a.length(); i++)
{
M_id=M_id*10+(a[i]-48);
}
for(int i=1; i<=z && k; i++)
{
if(a==A[i].nomi)
{
cout<<A[i].id<<" "<<A[i].nomi<<" "<<A[i].kucha<<" "<<A[i].uy_raqami<<" "<<A[i].xonadon_raqami;
k=false; x=i; y=1;
}
else if(a==A[i].kucha)
{
cout<<A[i].id<<" "<<A[i].nomi<<" "<<A[i].kucha<<" "<<A[i].uy_raqami<<" "<<A[i].xonadon_raqami;
k=false; x=i; y=2;
}
else if(a==A[i].uy_raqami)
{

```

```

cout<<A[i].id<<" "<<A[i].nomi<<" "<<A[i].kucha<<" "<<A[i].uy_raqami<<" "<<A[i].xonadon_raqami;
k=false; x=i; y=3;
}
else if(a==A[i].xonadon_raqami)
{
cout<<A[i].id<<" "<<A[i].nomi<<" "<<A[i].kucha<<" "<<A[i].uy_raqami<<" "<<A[i].xonadon_raqami;
k=false; x=i; y=4;
}
else if(M_id==A[i].id)
{
cout<<A[i].id<<" "<<A[i].nomi<<" "<<A[i].kucha<<" "<<A[i].uy_raqami<<" "<<A[i].xonadon_raqami;
k=false; l=true; x=i;
}
}
if(k) cout<<"Bunday so`z mavjud emas !!!";
cout<<"\n\n";
};

```

// Ekrandan kiritish funksiyasi

```

void kiritish()
{
cout<<"Yozishdan tuxtash uchun 'Chiqish' \n\n nomi   kucha   uy_raqami   xonadon_raqami\n";
while(true)
{
string b;
cin>>b;
if(b=="CHIQISH" || b=="chiqish" || b=="Chiqish")
break;
else
{
A[++z].id=z;
A[z].nomi=b;
cin>>A[z].kucha>>A[z].uy_raqami>>A[z].xonadon_raqami;
}
}
cout<<"_____ \n\n";

```

```
}
```

```
};
```

```
// Faylga yozish funksiyasi
```

```
void yozish()
```

```
{
```

```
ofstream f_wirat;
```

```
f_wirat.open("Manzil.bd");
```

```
for(int i=1; i<=z; i++)
```

```
{
```

```
f_wirat<<A[i].id<<" "<<A[i].nomi<<" "<<A[i].kucha<<" "<<A[i].uy_raqami<<"  
"<<A[i].xonadon_raqami<<"\n";
```

```
}
```

```
ofstream wirat;
```

```
wirat.open("Manzil.dat");
```

```
wirat<<z;
```

```
};
```

```
// Fayldan o`qish funksiyasi
```

```
void oqish()
```

```
{
```

```
ifstream read;
```

```
read.open("Manzil.dat");
```

```
read>>z;
```

```
ifstream f_read;
```

```
f_read.open("Manzil.bd");
```

```
for(int i=1; i<=z; i++)
```

```
{
```

```
f_read>>A[i].id>>A[i].nomi>>A[i].kucha>>A[i].uy_raqami>>A[i].xonadon_raqami;
```

```
}
```

```
};
```

```
// Ekranga chiqarish funksiyasi
```

```
void E_chiqarish()
```

```
{
```

```
for(int i=1; i<=z; i++)  
{  
    cout<<A[i].id<<" "<<A[i].nomi<<" "<<A[i].kucha<<" "<<A[i].uy_raqami<<" "  
    "<<A[i].xonadon_raqami<<"\n";  
}  
};
```

Xulosa

Men kurs ishimda moqimli shinflar xaqida malumotlar berdim va strukturalarning qurilishiga misoollar keltirdim sinflar bilan ishlashni ko`rsatdim , fayllar bilab ishlshni ko`rib chiqdim

Amaliy misolimni qurilishini strukturada ishladim . va blok sxemasini qurdim

Men dasturimni dev C++ da bajardim faylar bilan ishladin kiritish chiqarish oqimlaridan foydalandim oqimlisinflarning mazmuni biln tanishdim

C++ tilida dastur tuzishda oqimli sinflarning vazifasini o`rgandim