

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

**TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI
QARSHI FILIALI**



KOMPYUTER INJINIRINGI FAKULTETI

5330500 Kompyuter injiniring ta'lim yo'nalishi

Dasturiy injiniring kafedrasi

C++ tilida dasturlash fanidan

KURS ISHI

Mavzu: Satrlar bilan ishlash uchun klass

Bajardi:

Nematov Jamshid

Qabul qildi:

H.M.Normamatov

Qarshi-2016

**O`ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI VA
KOMMUNIKATSIYALARINI RIVOJLANTIRISH VAZIRLIGI
TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI
QARSHI FILIALI
KOMPYUTER INJINIRING FAKULTETI**

“TASDIQLAYMAN”

“Dasturiy injiniring” kafedrası

mudiri: _____

Yo`nalishi : 5330500 *Kompyuter injiniring* (“*Kompyuter injiniring*”, “AT -Servis”)
ta`lim yo`nalishi

Fanning nomi: C++ tilida dasturlash.

KURS ISHI

Kurs ishi mavzusi: Satrlar bilan ishlash uchun class

Topshirig`i: O`zunligi tayin qilingan satrlari bilan ishlash uchun class qurish va satr ustida bir qancha standart amallarni aniqlash.

Амалий топшириқ: Satrlarni ulash, satrda qism satrni izlash, satrni teskarilash, ikki satrni leksografik tarzda solishtirish.

Yo`llanma materiallar: *Adabiyotlar, internet.*

Chizmachilik qismi: algoritm, blok sxemalar

Kurs ishini himoya qilish muddati : _____

Guruh: KI-11-14 Talaba:

Nematov Jamshid

Rahbar: _____

X.M.Normamatov

Kurs ishini topshirig`ini qabul qildim: _____

TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI
QARSHI FILIALI
KOMPYUTER INJINIRING FAKULTETI

“C++ da dasturlash” fanidan

Kurs ishi bo'yicha topshiriq: O`zunligi tayin qilingan satrlari bilan ishlash uchun class qurish va satr ustida bir qancha standart amallarni aniqlash

Talaba: *Nematov Jamshid*

I.Ishning mazmuni:Satrlar bilan ishlash uchun class

II.Ishni topshirish muddati: _____

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

1. «Kompyuterlashtirishni yanada rivojlantirish va axborot-kommunikatsiya texnologiyalarini joriy etish chora-tadbirlari to'g'risida»gi O'zbekiston Respublikasi Vazirlar Mahkamasining Qarori // «Xalqso'zi» gazetasi, 2002 yil 8 iyun.
2. Karimov I.A. «Eri shilgan yutuqlarini mustahkamlab, yangi marralarga erishish harakat qilishimiz lozim». // «Xalqso'zi», 2006 y., 11-fevral.
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. Qosimov S.S. “Axborot texnologiyalari” Texnika oliyo'rtayurtlaribakalavriat bosqichitalabalar uchun o'quv qo'llanmasifatida tavsiya etilgan. Toshkent shaxri “Aloqachi” nashriyoti, 2006 y.
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, -686c.
7. Носиров Б. “С++ тилида дастурлаш” фанидан электрон ЎУМ., 1- қисм. Қарши. 2015й.
8. Shukurov O. Qorayev F. “Programmashdan masalalar to'plami” Qarshi. 2008 y.
9. Ziyonet.uz – O'zbekiston Respublikasi axborot-ta'lim portali.
10. acm.tuit.uz – Дастурий ечим тўғ'ри лигини автоматик тестловчи тизим.
11. inter-vuz.tuit.uz – Axborot resurslar kutubxonasi
12. lib.tuit.uz – TATU elektron kutubxonasi

13.etuit.uz –

TATUvertualta'limtizimi

14.dls.tuitkf.uz –

TATUQarshifilialimasofaviyta'limtizimi

IV.Tushuntirish yozilmasining tuzilishi:

1. Kirish qismi.
2. Nazariy qism
3. Asosiy qismi.
4. Xulosalar.
5. Foydalangan adabiyotlar.

Maslahatchilar: _____

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

REJA

I.KIRISH.....	6
II.SATRLAR VA ULAR USTIDA AMALLAR	7
2.1 Satr uzunligini aniqlash funksiyalari.....	7
2.2 Satrlarni nusxalash.....	8
2.3 Satrlarni ulash.....	9
2.4 Satrlarni solishtirish.....	10
2.5 Satrni teskari tartiblash.....	11
III. SINIF TUSHUNCHASI.....	12
2.1 Sinf xaqida.....	12
2.2 Конструкторлар ва деструкторлар.....	17
IV AMALIY MASALA	21
4.1. Masalaning quyilishi va tahlili.....	21
4.2. Algoritm blok-sxemasi.....	21
4.3. Algoritm dasturiy kodi va natijalar.....	28
V. HULOSA.....
32VI. FOYDALANILGANADABIYOTLAR.....

I KIRISH

Insoniyat o'zining tarixiy taraqqiyoti jarayonida har xil ish qurollarini yaratgan. Bu ish qurollari uning jismoniy mehnatini yengillashtirishga xizmat qilgan. Bularga oddiy bolta, tesha, arradan tortib hozirgi zamon qudratli mashina va traktorlarini misol sifatida keltirish mumkin.

Inson bu davrda faqat mehnat qurollarini yaratish bilan chegaralanib qolmay, balki u o'zining aqliy mehnatini yengillashtirish qurollarini ham yaratdi. Bunga oddiy hisob-kitob toshlaridan tortib, hozirgi kunda ham o'z kuchi va qulayligini yo'qotmagan cho'tlar misol bo'la oladi.

XX asrning 30-40 yillariga kelib, EHMlarning birinchi loyihalari paydo bo'la boshladi. Birinchi EHM yaratish ishlarini 1937 yilda AQSHning Ayova shtatida joylashgan universitetning professori A. Atanasov boshladi. Millati bolgar bo'lgan bu olim yaratmoqchi bo'lgan EHM matematik-fizikaning ayrim masalalarini yechishga mo'ljallangan edi. Ammo ikkinchi jahon urushi bu ishlarni oxirigacha yetkazish imkonini bermadi. Atanasovning buyuk xizmatlari shundaki, u birinchi bo'lib EHMlarda ikkilik sanoq sistemasini qo'llashning qulayligini ko'rsatadi.

Axborot kommunikatsion texnologiyalarini taraqqiy etishida bevosita dasturlash tillarining o'rni beqiyos. Ayniqsa, hozirgi davrga kelib C++, Java, Delphi dasturlash tillar yordamida shaxsiy kompyuterlar uchun amaliy dasturiy to'plamlardan tashqari SmartPhone va Planshetlar uchun operatsion tizim (iOS, Android, Windows mobile, Symbian va h.k) va ilovalar yaratilmoqda. Informatsion texnologiyalarning yana bir muhim jihatlaridan biri shundaki, bu fan jadal sur'atlarda o'sib, yil sayin yangidan-yangi yo'nalishlarga, mutaxassisliklarga tarmoqlanib ketmoqda: algoritmik, mantiqiy, obyektga yo'naltirilgan, vizual, parallel dasturlash texnologiyalari, animatsiya, multimediya, Web, ma'lumotlar bazasini boshqarish tizimlari, ko'p prosessorli, neyron arxitekturali kompyuterlar va hokazo. Ko'rinib turibdiki, informatika meta fan darajasiga ko'tarilib, uni bitta o'quv kursi chegarasida to'liq o'zlashtirishning imkoni bo'lmay qoldi.

Informatsion texnologiyalar sohasi bo'yicha rus va ingliz tillarida qo'llanmalar juda ko'p chop etilmoqda. Oxirgi yillarda o'zbek tilidagi qo'llanmalar ham ko'payib qoldi.

Shu qullanmalardan foydalangan xolda dastrulash tilarini chuqurroq o'rgannamiz deb uylayman

II.SATRLAR VA ULAR USTIDA AMALLAR

2.1 Satrlar va ular ustida amallar

Standart S++ tili ikki xildagi belgilar majmuasini qo'llab-quvvatlaydi. Birinchi toifaga, an'anaviy, "tor" belgilar deb nomlanuvchi 8-bitli belgilar majmuasi kiradi, ikkinchisiga 16-bitli "keng" belgilar kiradi. Til kutubxonasida har bir guruh belgilari uchun maxsus funktsiyalar to'plami aniqlangan.

S++ tilida satr uchun maxsus tur aniqlanmagan. Satr char turidagi belgilar massivi sifatida qaraladi va bu belgilar ketma-ketligi satr terminatori deb nomlanuvchi nol kodli belgi bilan tugaydi ('\0'). Odatda, nol-terminator bilan tugaydigan satrlarni ASCIIZ –satrlar deyiladi. Sart konstanta deb qo'shtirnoqlar ichiga olingan belgilar ketma-ketligiga aytiladi:

“Ushbu belgilar ketma-ketligiga satr deyiladi.”

Quyidagi jadvalda S++ tilida belgi sifatida ishlatilishi mumkin bo'lgan konstantalar to'plami keltirilgan.

Belgilar sinflari	Belgi konstantalar
Katta harflar	'A' ... 'Z', 'A'...'YA'
Kichik harflar	'a' ... 'z', 'a'...'ya'
Raqamlar	'0' ... '9'
Bo'sh joy	gorizontal tabulyatsiya (ASCII kodi 9), satrni o'tkazish (ASCII kodi 10), vertikal tabulyatsiya (ASCII kodi 11), formani o'tkazish (ASCII kodi 12), karetkani qaytarish (ASCII kodi 13)
Puntuatsiya belgilari (ajratuvchilar)	! " # \$ % & ' () * + - , . / : ; < = > ? @ [\] ^ _ { } ~
Boshqaruv belgilari	ASCII kodi 0...1Fh oralig'ida va 7Fh bo'lgan belgilar
Probel	ASCII kodi 32 bo'lgan belgi
O'n otilik raqamlar	'0'...'9', 'A'...'F', 'a'...'f'

Satr massivi e'lon qilinishida satr oxiriga terminator qo'yilishini va natijada satrga qo'shimcha bitta bayt qo'shilishi inobatga olinishi kerak:

```
char satr[10];
```

Ushbu e'londa satr satri uchun jami 10 bayt ajratiladi – 9 satr hosil qiluvchi belgilar uchun va 1 bayt terminator uchun.

Satr o'zgaruvchilari e'lon qilinishida boshlang'ich qiymatlar qabul qilishi mumkin. Bu holda kompilyator avtomatik ravishda satr uzunligi hisoblaydi va satr oxiriga nol-terminatorni qo'shib qo'yadi:

```
char Hafta_kuni[]="Juma";
```

Ushbu e'lon quyidagi e'lon bilan ekvivalent:

```
char Hafta_kuni[] = {'J', 'u', 'm', 'a', '\0'};
```

Satr qiymatini o'qishda oqimli o'qish operatori ">>" o'rniga getline() funksiyasini ishlagan ma'qul hisoblanadi, chunki oqimli o'qishda probellar inkor qilinadi (garchi ular satr belgisi hisoblansa ham) va o'qilayotgan belgilar ketma-ketligi satrdan "oshib" ketganda ham belgilarni kiritish davom etishi mumkin. Natijada satr o'ziga ajratilgan o'lchamdan ortiq belgilarni qabul qilishi mumkin. Chu sababli, getline() funksiyasi ikkita parametrga ega bo'lib, birinchi parametr o'qish amalga oshirilayotgan satrga ko'rsatgich, ikkinchi parametrda esa kiritilishi kerak bo'lgan belgilar soni ko'rsatiladi. Satrni getline() funksiyasi orqali o'qishga misol ko'raylik:

```
#include <iostream.h>
int main()
{
char satr[6];
cout<<"Satrni kiriting: "<<"\n";
cin.getline(satr,6);
cout<<"Siz kiritgan satr: "<<satr;
return 0;
}
```

Programmada satr satri 5 ta belgini qabul qilishi mumkin, ortiqchalari tashlab yuboriladi. getline() funksiyasiga murojaatda ikkinchi parametr qiymati o'qilayotgan satr uzunligidan katta bo'lmasligi kerak.

Satr bilan ishlaydigan funksiyalarning aksariyati string.h kutubxonasida jamlangan. Nisbatan ko'p ishlatiladigan funksiyalarning tavsifini keltiramiz.

2.2 Satrlarni nusxalash

Satr qiymatini biridan ikkinchisiga nusxalash mumkin. Buning uchun bir qator standart funksiyalar aniqlangan bo'lib, ularning tavsiflari quyida keltiramiz.

strcpy() funksiyasi prototipi

```
char* strcpy(char* str1, const char* str2)
```

ko'rinishga ega va bu funksiya str2 ko'rsatib turgan satrdagi belgilarni str1 ko'rsatib turgan satrga baytma-bayt nusxalaydi. Nusxalash str2 ko'rsatib turgan satrdagi nol-terminal uchraguncha davom etadi. Chu sababli, str2 satr uzunligi str1 satr uzunligidan katta emasligiga ishonch hosil qilish kerak, aks holda berilgan sohasida (segmentida) str1 satrdan keyin joylashgan berilganlar "ustiga" str2 satrning "ortiqcha" qnomi yozilishi mumkin.

Navbatdagi programma qnomi "Satrni nusxalash!" satrini Str satrga nusxalaydi:

```
char Str[20];
```

```
strcpy(Str, "Satrni nusxalash!");
```

Zarur bo'lganda satrning qaysidir joyidan boshlab, oxirigacha nusxalash mumkin. Masalan, "Satrni nusxalash!" satrini 8 belgisidan boshlab nusxa olish zarur bo'lsa, uni quyidagicha yechish mumkin:

```

#include <iostream.h>
#include <string.h>
int main()
{
    char Str1[20]="Satrni nusxalash!";
    char Str2[20];
    char* kursatgich=Str1;
    kursatgich+=7;
    strcpy(Str2,kursatgich);
    cout<<Str2<<endl;
    return 0;
}

```

strcpy() funksiyasining strcpy() funksiyasidan farqli joyi shundaki, unda bir satrdan ikkinchisiga nusxalanadigan belgilar soni ko'rsatiladi. Uningsintaksisi quyidagi ko'rinishga ega:

```
char* strcpy(char* str1, const char* str2, size_t num)
```

Agar str1 satr uzunligi str2 satr uzunligidan kichik bo'lsa, ortiqcha belgilar "kesib" tashlanadi. strcpy() funksiyasi ishlatilishiga misol ko'raylik:

```

#include <iostream.h>
#include <string.h>
int main()
{
    char Uzun_str[]="01234567890123456789";
    char Qisqa_str[]="ABCDEF";
    strcpy(Qisqa_str,Uzun_str,4);
    cout <<"Uzun_str= " <<Uzun_str<<endl;
    cout <<"Qisqa_str=" <<Qisqa_str<<endl;
    return 0;
}

```

Programmada Uzun_str satri boshidan 4 belgi Qisqa_str satriga oldingi qiymatlar ustiga nusxalanadi va natijada ekranga

```

01234567890123456789
0123EF

```

xabarlarini chop etiladi.

strdup() funksiyasiga yagona parametr sifatida satr-manbaga ko'rsatgich uzatiladi. Funktsiya, satrga mos xotiradan joy ajratadi, unga satrni nusxalaydi va yuzaga kelgan satr-nusxa adresini qaytaradi. strdup() funksiya sintaksisi:

```
char* strdup(const char* source)
```

Quyidagi programma bo'lagida satr1 satrining nusxasi xotiraning satr2 ko'rsatgan joyida paydo bo'ladi:

```

char* satr1="Satr nusxasini olish.";
char* satr2;
satr2=strdup(satr1);

```

2.3 Satrlarni ulash

Satrlarni ulash (konkatenatsiya) amali yangi satrlarni hosil qilishda keng qo'llaniladi. Bu maqsadda string.h kutubxonasida strcat() va strncat() funktsiyalari aniqlangan.

strcat() funktsiyasi sintaksisi quyidagi ko'rinishga ega:

```
char* strcat(char* str1, const char* str2)
```

Funktsiya ishlashi natijasida str2 ko'rsatayotgan satr, funktsiya qaytaruvchi satr – str1 ko'rsatayotgan satr oxiriga ulanadi. Funktsiyani chaqirishdan oldin str1 satr uzunligi, unga str2 satr ulanishi uchun yetarli bo'lishi hisobga olingan bo'lishi kerak.

Quyida keltirilgan amallar ketma-ketligi bajarilishi natijasida satr satriga qo'shimcha satr ostilari ulanishi ko'rsatilgan:

```
char satr[80];  
strcpy(satr, "Bu satrga ");  
strcat(satr, "satr osti ulandi.");
```

Amallar ketma-ketligini bajarilishi natijasida satr satri "Bu satrga satr osti ulandi." qiymatiga ega bo'ladi.

strncat() funktsiyasi strcat() funktsiyadan farqli ravishda str1 satrga str2 satrning ko'rsatilgan uzunligidagi satr ostini ulaydi. Ulanadigan satr osti uzunligi funktsiyaning uchinchi parametri sifatida beriladi. Funktsiya sintaksisi

```
char* strncat(char* str1, const char* str2, size_t num)
```

Pastda keltirilgan programma bo'lagida str1 satrga str2 satrning boshlang'ich 10 ta belgidan iborat satr ostini ulaydi:

```
char satr1[80]="Programmash tillariga misol bu-";  
char satr2[80]="C++,Pascal, Basic";  
strncpy(satr1,satr2,10);  
cout<<satr1;
```

Amallar bajarilishi natijasida ekranga "Programmash tillariga misol bu-C++,Pascal" satri chop etiladi.

2.4 Satrlarni solishtirish

Satrlarni solishtirish ularning mos o'rindagi belgilarini solishtirish (katta yoki qichikligi) bilan aniqlanadi. Buning uchun string.h kutubxonasida standart funktsiyalar mavjud.

strcmp() funktsiyasi sintaksisi

```
int strcmp(const char* str1, const char* str2)
```

ko'rinishiga ega bo'lib, funktsiya str1 va str2 solishtirish natijasi sifatida son qiymatlarni qaytaradi va ular quyidagicha izohlanadi:

- <0 – agar str1 satri str2 satridan kichik bo'lsa;
- =0 – agar str1 satri str2 satriga teng bo'lsa;
- >0 – agar str1 satri str2 satridan katta bo'lsa.

Funktsiya harflarning bosh va kichikligini farqlaydi. Buni misolda ko'rishimiz mumkin:

```
char satr1[80]="Programmalash tillariga bu- C++,pascal, Basic.“;
char satr2[80]="Programmalash tillariga bu- C++,Pascal, Basic.“;
int i;
i= strcmp(satr1,satr2);
```

Natijada *i* o'zgaruvchisi musbat qiymat qabul qiladi, chunki solishtirilayotgan satrlardagi "pascal" va "Pascal" satr ostilarida birinchi harflar farq qiladi. Keltirilgan misolda *i* qiymati 32 bo'ladi – farqlanuvchi harflar kodlari ayrimasi 32ga teng . Agar funktsiyaga

```
i= strcmp(satr2,satr1);
```

ko'rinishida murojaat qilinsa *i* qiymati –32 bo'ladi.

Agar satrlardagi bosh yoki kichik harflarni farqlamasdan solishtirish amalini bajarish zarur bo'lsa, buning uchun stricmp() funktsiyasidan foydalanish mumkin. Yuqorida keltirilgan misoldagi satrlar uchun

```
i=strncmp(satr2,satr1);
```

amali bajarilganda *i* qiymati 0 bo'ladi.

strncmp() funktsiyasi sintaksisi

```
int strncmp(const char* str1, const char* str2, size_t num)
```

ko'rinishida bo'lib, str1 str2 satrlarni boshlang'ich num sonidagi belgilarini solishtiradi. Funktsiya harflar registrini inobatga oladi. Yuqorida misolda aniqlangan satr1 va satr2 satrlar uchun

```
i=strncmp(satr1,satr2,31);
```

amali bajarilishida *i* qiymati 0 bo'ladi, chunki satrlar boshidagi 31 belgilar bir xil.

strncmpi() funktsiyasi strncmp() funktsiyasidek amal qiladi, farqli tomoni shundaki, solishtirishda harflarning registrini hisobga olinmaydi. Xuddi shu satrlar uchun

```
i=strncmpi(satr1,satr2,32);
```

amali bajarilishi natijasida *i* o'zgaruvchi qiymati 0 bo'ladi.

2.5 Satrni teskari tartiblash

Satrni teskari tartiblashni uchun strrev() funktsiyasidan foydalanish mumkin.

Bu funktsiya quyidagicha prototipga ega:

```
char* strrev(char* str)
```

Satr reversini hosil etishga misol:

```
char str[]="telefon";
```

```
cout <<strrev(str);
```

amallar bajarilishi natijasida ekranga "nofelet" satri chop etiladi.

III. SINF TUSHUNCHASI

3.1 Sinflar

Синф тушунчаси C++ тилидаги энг муҳим тушунчалардан биридир. Синфсинтаксиси структура синтаксисига ўхшашдир ва унинг кўриниши куйидагича:

```
class <синф номи>
{
// синфнинг ёпик берилганлар–аъзолари ва функциялар–
// аъзолари
public:
// синфнинг очик берилганлар–аъзолари ва функциялар–
// аъзолари
}
```

<объектлар рўйхати>

Одатда синф тавсифида <объектлар рўйхати> қисми шарт эмас. Синф объектларикейинчалик, зарурат бўйича эълон қилиниши мумкин. Гарчи <синф номи> қисми ҳам мажбурий бўлмаса ҳам, унинг бўлгани маъкул. Чунки <синф номи> берилганларнингтурининг янги номи бўлиб, унинг ёрдамида шу синф объектлари аниқланади.Синф ичида эълон қилинган функция ва берилганлар шу синф аъзоларихисобаланди. Синф эълонининг ичида эълон қилинган ўзгарувчилар берилганлар-аъзолар,синф ичида эълон қилинган функциялар функциялар–аъзолар дейилади. Келишув бўйича синф ичидаги барча функция ва ўзгарувчилар шу синф учун ёпик ҳисобланади, яъниуларни фақат шу синф аъзолари ишлатиши мумкин. Синфнинг очик аъзоларини эълонқилиш учун public калит сўзи ва «:» белгисидан фойдаланилади. Синф эълонидаги publicсўзидан кейин эълон қилинган функциялар ва ўзгарувчиларга синфнинг бошқа аъзолари ва программанинг шу синф ишлатилган ихтиёрий жойидан мурожаат қилиш мумкинбўлади.Синф эълонига мисол:

```
class Sinf_1
{
// синфнингёпикэлементи
int a;
public:
int get_a();
```

```
void set_a(int_num);  
}
```

Гарчи `int get_a()` ва `void set_a(int_num)` функциялари `Sinf_1` синф ичида эълонқилинган бўлса ҳам, улар ҳали аниқлангани йўқ. Функцияни аниқлаш учун синф номи ва

«::» белгиларини ёзиш орқали амалга оширилади. Бу ерда «::» – кўриш соҳасиникенгайтириш амали дейилади. Функция-аъзони аниқлашнинг умумий шакли қуйидагича:

```
<тур><синф номи>::<функция номи> (<параметрлар рўйхати>)  
{  
// функция танаси  
}
```

Юқорида эълон қилинган `Sinf_1` синфнинг `int get_a()` ва `void set_a(int_num)`

функция-аъзолари аниқлашга мисол келтирилган:

```
int Sinf_1 :: get_a()  
{ return a; }  
void Sinf_1:: set_a(int num) {a=num;}
```

`Sinf_1` синфини эълон қилиш шу синф туридаги объектларини юзага келтирмайди.

Синф объектларини юзага келтириш учун синф номини берилганлар тури спецификаторисифатида ишлатиш зарур бўлади. Масалан,

```
Sinf_1 obj1,obj2;
```

Синф объекти яратилгандан кейин «.» ёрдамида синфнинг очик аъзоларига мурожаат қилиш мумкин бўлади. Мисол учун

```
obj1.set_a(20);  
obj2.set_a(50);
```

мурожаатлар орқали `obj1` ва `obj2` объектларнинг а ўзгарувчиларига қийматлар берилади. Ҳар бир объект синфда эълон қилинган ўзгарувчиларнинг ўз нусхаларига эга бўлади. Шусабабли, `obj1` объектидаги а ўзгарувчи `obj2` объектдаги а ўзгарувчидан фарқ қилади.

Синф элементларига кўрсаткичлар орқали ҳам амалга ошириш мумкин.

Қуйидагимисол буни намоён этади.

```
class Nuqta  
{  
public:  
int x,y;  
void Koord_Qiymat_Berish( int _x, int _y);  
};
```

```

voidNuqta::Koord_Qiymat_Berish(int _x, int _y)
{
x= _x; y=_y;
}
int main()
{
Nuqta x0y;
Nuqta * Koord_kurssatgich= & x0y;
//...
x0y.x=0;
Koord_kursatgich -> y=0;
Koord_Kursatgich -> Koord_Qiymat_Berish(10,15);
//...
return 0;
}

```

Шуни қайд қилиш керакки, синф объектларига “.” ва ”->” орқали мурожаатқилишнинг компиляция нуқтаи-назаридан ҳеч бир фарқи йўқ. Компилятор “.” билан мурожаатни ”->” билан алмаштиради. Масалан,

```
obj1.set_a(20);
```

кўрсатмаси компилятор томонидан

```
(&obj1)-> set_a(20);
```

кўринишидаги кўрсатма билан алмаштирилади.

C++ тили синфнинг берилганлар-аъзоларига маълум бир чекланишлар қўяди:

– берилганлар-аъзолар auto, extern ёки register модификаторлари билан аниқланиши мумкин эмас;

– синфнинг берилганлар-аъзолари шу синф туридаги объект бўлиши мумкин эмас, лекин улар шу синфга кўрсаткич ёки мурожаат(&) бўлиши, бошқа синфобъекти бўлиши мумкин.

Синф, унинг аъзолари ишлатилишидан олдин эълон қилинган бўлиши керак. Бироқ, айрим ҳолларда синфда ҳали эълон қилинмаган синфга кўрсаткич ёки мурожаат(&) эълон қилишга зарурат бўлиши мумкин. Бу ҳолда синфнинг тўлиқ бўлмаган эълонидан фойдаланишга тўғри келади.

Синфнинг тўлиқ бўлмаган эълони куйидаги кўринишга эга:

```
class <синф номи>;
```

Мисол кўрайлик.

```
class Sinf2; // синфнинг тўлиқмас эълони
```

```
class Sinf1
```

```
{
int x;
```

```

Sinf2 * sinf2; // sinf2 синфига кўрсаткич
public:
Sinf1(int _x) {x=_x;}
};
int main()
{
//...
return 0;
}
class Sinf2 // Sinf2 синфининг тўлиқ эълони
{
int a;
public:
Sinf2();
};

```

Шуни қайд этиш керакки, синф эълони структура эълониغا ўхшаш, фарқлиравишда:– синф эълонида public, protected ёки private мурожаат модификаторлари ишлатилади;

- struct калит сўзи ўрнида class ёки union калит сўзлари ишлатилиши мумкин;
- одатда синф таркибида берилганлардан ташқари функция-аъзолар киради;
- синф конструктор ёки деструктор деб номланувчи махсус функция-аъзоларигаэга бўлади.

Қуйида struct ва union калит сўзлари билан аниқланган синфларга мисол келтирилган.

```

struct Nuqta
{
private:
int x; int y;
public:
int Olish_X();
int Olish_Y();
void Qiymat_Berish_X(int _x);
void Qiymat_Berish_Y(int _y);
};
union Bit
{
Bit(unsigned int n);
void Bit_Chop_Qilish();
};

```

```
unsigned int num;  
unsigned char c [sizeof(unsigned int)];};
```

Мурожаат спецификатори, ундан кейин жойлашган барча синф элементларига қўлланилади, токи бошқа спецификатор учрамагунча ёки синф эълони тугамагунча. Қуйида спецификаторлар тавсифи келтирилган.

Синфга мурожаат спецификаторлари: `private` - берилганлар-аъзоларга ва функциялар-аъзоларга фақат шу синф

функциялар-аъзолари мурожаат қилиши (ишлатиши) мумкин;

`protected`-берилганлар-аъзоларга ва функциялар-аъзоларга фақат шу синф ваш усинфдан ҳосил бўлган синфлар функциялар-аъзолари мурожаат қилиши (ишлатиши) мумкин;

`public` - берилганлар-аъзоларга ва функциялар-аъзоларга фақат шу синф функция-аъзолари ва синф объекти мавжуд бўлган программа функциялари мурожаат қилиши (ишлатиши) мумкин;

С++ тилида структура ва бирлашмалар синф турлари деб қаралади.

Структура ва синфлар бир-бирига ўхшаш, фақат келишув бўйича мурожаат билан фарқ қилади: структурада келишув бўйича барча

элементлар `public` мурожаатига эга бўлса, синфда улар

`private` мурожаатида бўлади. Бирлашмада ҳам, худди структурадек келишув бўйича элементлар `public` мурожаатда бўлади.

Структура ва бирлашмалар ўз конструктор ва деструкторларига эга бўлиши мумкин. Шу билан биргаликда бирлашмаларни синф сифатида ишлатишга маълум бирчекловлар мавжуд. Биринчидан, улар бирорта синф вориси бўлиши мумкин эмас, ўзларихам бошқа синфлар учун она синф бўла олмайди. Улар `static` атрибутли аъзоларга, ҳамда конструктор ва деструкторли объектларга эга бўлиши мумкин эмас.

Бирлашмани синф сифатида ишлатилишига мисол кўрайлик.

```
#include <iostream.h>  
union Bit  
{  
    Bit(unsigned int n);  
    void Bit_Chop_Qilish();  
    unsigned int num;  
    unsigned char c[sizeof(unsigned int)];  
};  
Bit::Bit(unsigned int n){ num=n; };  
void Bit::Bit_Chop_Qilish( )  
{  
    int i,j;
```

```

for (j=sizeof(unsigned int)-1; j>=0; j--)
{
cout<<"Байтнинг иккилик кўриниши "<<j<<": ";
for (i=128; i; i>>=1)
{
if (i & c[j]) cout<<'1';
else cout<<'0';
}
cout<<endl; }
}
int main( )
{
Bit bit(2000);
Bit.Bit_Chop_Qilish();
return 0;
}

```

Бу мисолда объектга берилган ишорасиз бутун қийматнинг иккилик кўриниши чопэтилади.

3.2 Конструкторлар ва деструкторлар

Объектни яратишда уни инициализациялаш керак. Бу мақсадда C++ тилида конструктор деб номланувчи махсус функция-аъзо аниқланган. Синф конструктори ҳарсафар синф объекти яратилиши пайтида чақирилади. Конструктор номи ўзи аъзо бўлган синф номи билан устма–уст тушади ва қайтарувчи қийматга эга бўлмайди. Масалан,

```

#include <iostream.h>
class Sinf
{
int var;
public:
Sinf(); // Конструктор
void Chop_etish_var();
};
Sinf::Sinf()
{
cout<< "Конструктор ишлади \n";
var=0;
}
Sinf::Chop_etish_var(){cout<<var;}

```

```

int main()
{
    Sinf ob;
    ob.Chop_Etish_var();
    //...
    return 0;
}

```

Бу мисолда Sinf конструктори экранга хабар чиқаради ва ёпиқ var ўзгарувчинини инициализациялайди.

Шуни қайд этиш керакки, программа тузувчи конструктор чақирадиган код ёзмаслиги керак. Барча зарур ишни компилятор амалга оширади.

Юқорида қайд қилингандек конструктор у тегишли синф объекти яратилаётган пайтда чақиради. Ўзнавбатида объект бу объектни эълон қилувчи оператор бажарилишида яратилади. Шунинг учун ҳам C++ тилида ўзгарувчини эълон қилувчи оператор бажариловчи оператор хисобланади. Глобал объектлар учун конструктор программа бажарилиши бошланганда чақиради. Локал объектлар учун конструктор ўзгарувчи эълонининг ҳар бир бажарилишида чақиради.

Конструкторга нисбатан тескари амал бажарадиган функция-аъзоларга деструкторлар дейилади. Бу функция-аъзо объект ўчирилишида чақиради. Одатда деструктор объект томонидан эгалланган хотирани бўшатиш учун хизмат қилади. Унинг номи синф номи билан мос тушади, фақат олдида '~' белгиси қўйилади.

Қуйида деструктор аниқланган синфга мисол келтирилган.

```

#include <iostream.h>
class Sinf;
{
    int var;
public:
    Sinf(); // Конструктор
    ~Sinf(); // Деструктор
    void Chop_etish_var();
    Sinf::Sinf()
    {
        cout << "Конструктор ишлади \n";
        var=0;
    }
    Sinf::~Sinf()
    {

```

```

cout<< “Деструктор ишлади \n”; }
void Sinf::Chop_etish_var( )
{
cout<<var<<endl;
}
int main()
{
Sinf ob;
ob.Chop_Etish_var();
//...
return 0;
}

```

Деструктор объект ўчирилишида чақиради. Глобал объектлар программатугашида ўчирилади. Локал объектлар – уларни кўриш соҳасидан чиқишда ўчирилади.

Шуни алоҳида қайд этиш керакки, конструктор ва деструкторларга кўрсаткичларҳосил қилиш мумкин эмас.

Агар синф ўзгарувчиларини инициализация қилиш зарур бўлса, параметрликонструктор ишлатилди. Юқорида келтирилган мисолга ўзгартириш киритамиз.

```

#include <iostream.h>
class Sinf
{
int a, b;
58
public:
Sinf(int x, int y); // Конструктор
~Sinf(); // Деструктор
void Chop_etish_var();
}
Sinf::Sinf(int x, int y)
{
cout<< “Конструктор ишлади \n”;
a=x; b=y;
}
Sinf::~~Sinf()
{
cout<< “Деструктор ишлади \n”;
}

```

```

void Sinf::Chop_etish_var()
{
cout<<a<<b<<endl;
}
int main()
{
Sinf ob(5,10);
ob.Chop_Etish_var();
//...
return 0;
}

```

Бу ерда ob объекти эълонида конструкторга узатилган қийматлар синф таркибидаги a ва b ёпиқ ўзгарувчиларни инициализация қилишда ишлатилади.

Параметрли конструкторга қиймат узатиш синтаксиси қуйидаги ифоданинг қисқарган шакли ҳисобланади:

```

“Sinf ob(5,10); “
ифодаси
“Sinf(5,10);“

```

ифодаси билан эквивалент. Аксарият ҳолларда қисқартирилган шакл ишлатилади.

Конструктордан фарқли равишда деструктор параметрга эга бўлиши мумкин эмас,

чунки ўчирилаётган объект ўзгарувчиларига қиймат бериш маънога эга эмас.

Конструктор учун аниқланган бир нечта қоидаларни келтирамиз:

- конструктор учун қайтарилувчи қиймат тури кўрсатилмайди;
- конструктор қиймат қайтармайди;
- конструктор ворислик билан ўтмайди;
- конструктор const, volatile, static ёки virtual модификаторлари билан эълонқилинмайди.

Агар синф аниқланишида конструктор эълон қилинмаса, компилятор ўзи келишувбўйича параметрсиз конструкторни ҳосил қилади.

Деструктор учун қуйидаги қоидалар аниқланган:

- деструктор параметрларга бўлиши мумкин эмас;
- деструктор қиймат қайтармайди;
- деструктор ворислик билан ўтмайди;
- синф биттадан ортиқ деструкторга эга бўлиши мумкин эмас;
- деструктор const, volatile, static ёки virtual модификаторлари билан эълонқилинмайди.

Агар синфда деструктор эълон қилинмаса, компилятор ўзи келишув бўйича конструкторни ҳосил қилади.

Одатда синф берилганлари-аъзолари конструктор танасида инициализацияланади.

Лекин инициализациянинг бошқа усул билан – элементларни инициализациялаш рўйхати орқали амалга ошириш мумкин. Элементларни инициализациялаш рўйхати функция сарлавҳаси аниқланишидан кейин икки нукта (‘:’) кейин жойлашади ва унда вергул билан ажратилган ҳолда берилганлар-аъзолар ва таянч синфлар ёзилади. Ҳар бир элемент учун қавс ичида инициализацияда ишлатиладиган бир ёки бир нечта параметрлар кўрсатилади.

Қуйидаги мисолда элементларни инициализациялаш рўйхати орқали синф ўзгарувчиларини инициализация қилиш кўрсатилган.

```
class Sinf
{
int a, b;
public:
Sinf(int x, int y); // Конструктор
};
Sinf::Sinf( int x, int y): a(x), b(y);
{
cout<< “Конструктор ишлади \n”;
}
//...
```

Келтирилган программа бўлагида конструктор бажарадиган иш олдинги мисолдаги конструктор иши билан эквивалент.

Албатта, синф элементларини инициализация қилишнинг қайси шаклини қўллаш программа тузувчига боғлиқ. Бироқ, шундай ҳолатлар бўладики, унда элементларни инициализациялаш рўйхатидан фойдаланмасликнинг иложи йўқ: синфнинг берилганлар–константаларига ва кўрсаткичларига бошланғич қиймат беришда; синф аъзоси объект бўлганда ва бу объект конструктори бир ёки бир нечта параметрларга қиймат бериш ниталаб қилган ҳолларда.

Келишув бўйича конструкторларни ишлатишда коллизиядан (бир нарсани икки хил тушунишдан) қочиш керак, яъни синфда бир нечта конструктор бўлганда компилятор қачон уларнинг қайси бири чақирилишини аниқ билиши керак. Қуйидаги мисолда бу ҳолат билан боғлиқ хато кўрсатилган.

```
class S
```

```
{
public:
S(); // Келишув бўйича конструктор
S(int i=0); // Келишув бўйича конструктор ўрнида
//ишлатилиши мумкин бўлган конструктор
};
int main()
{
S ob1(10); // S::S(int) конструктори ишлатилади.
Sob2; //Нотўғри. S::S(int)ёки S::S()
// конструкторларинингқайси бири
// чақирилиши номаълум.
```

Бу зиддиятни ҳал қилиш йўли – бу синф эълонидан келишув бўйича конструкторни ўчиришдир.

IV AMALIY MASALA

4.1. Masalaning quyilishi va tahlili

Курс ишимавузулари:

№	Mavzu	Mavzuning tavsifi	Amallar/metodlar	Qo'llanilishi
1.	Satrlar bilan ishlash uchun klass	Uzunligini tayin qilingan satrlari bilan ishlash uchun klass kurish va satrukturada bir qancha standart amallarni aniqlash	Satrlarni ulash, satrdagi qismlarni izlash, satrni teskarilash, ikki satrni leksikografi ktarzdasolishtirish	

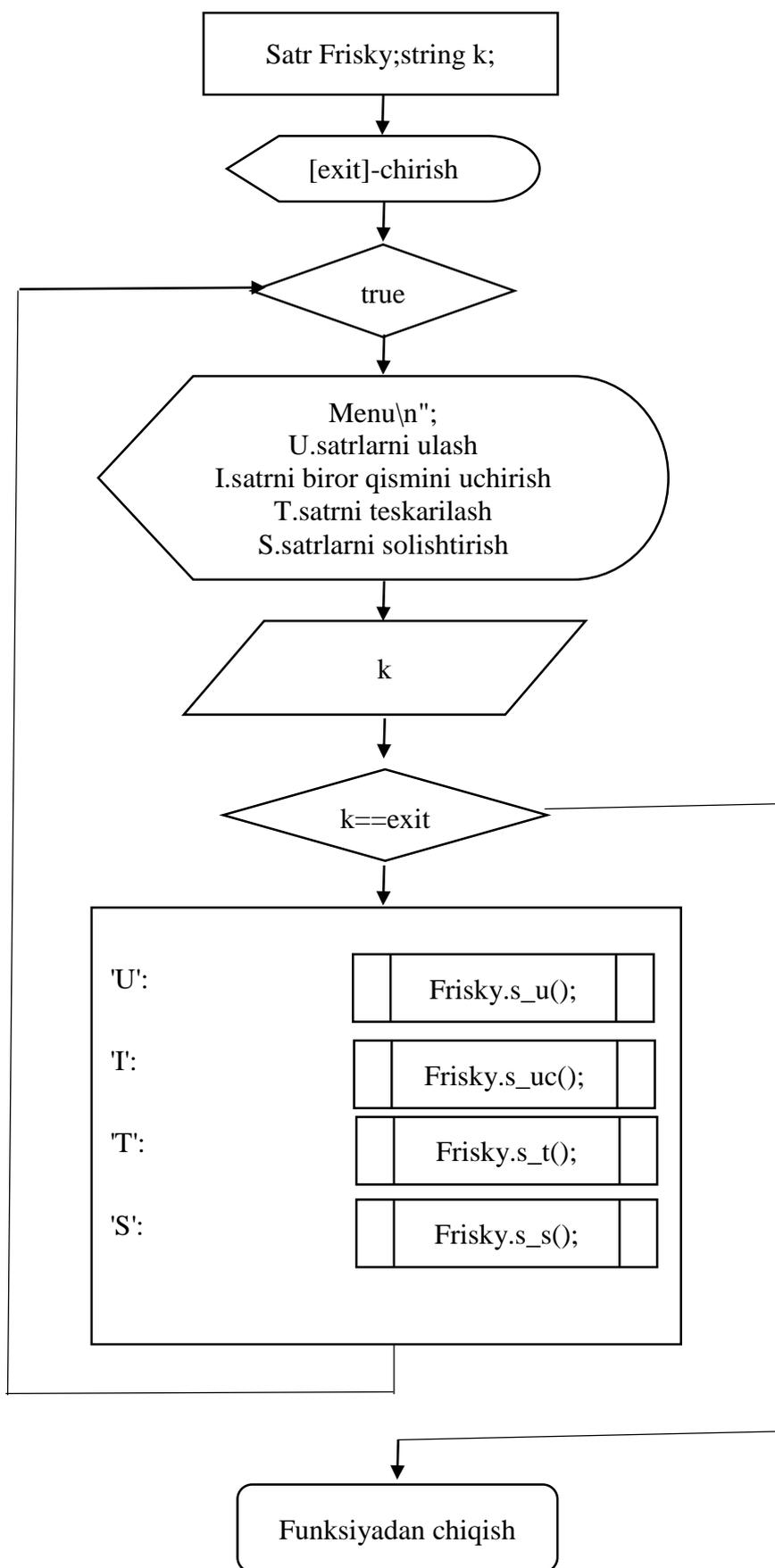
4.2. Algoritm blok-sxemasi

class Satr

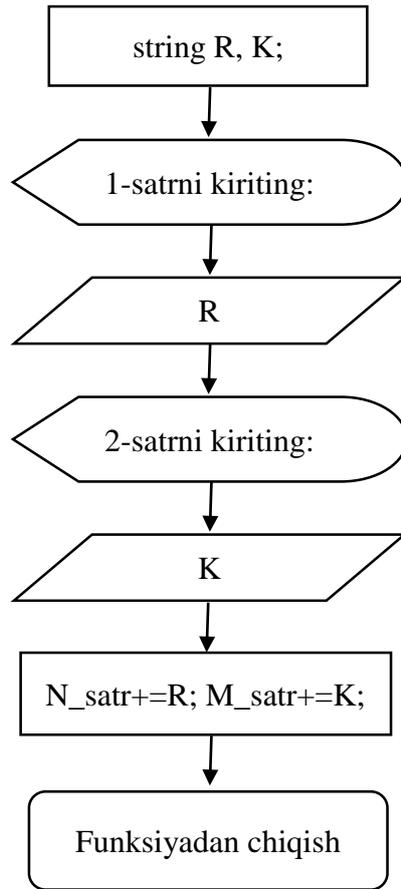
```
public:
    Satr();
    ~Satr();
    void s_uc();
    void s_t();
    void s_u();
    void s_s();

private:
    string N_satr, M_satr;
```

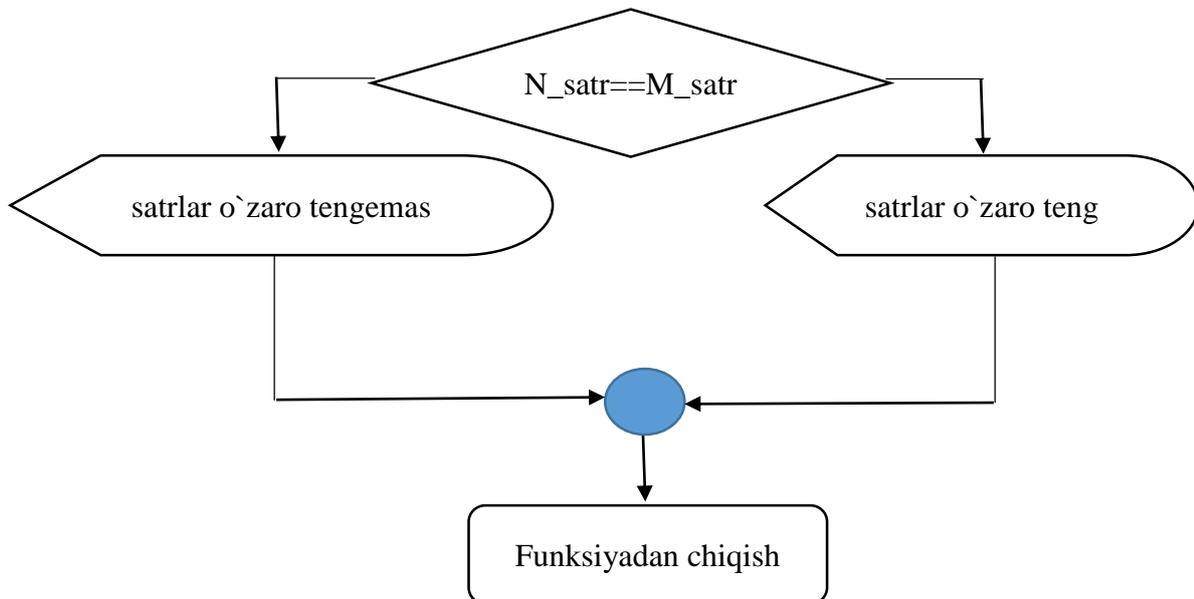
Asosiy funksiya int main()



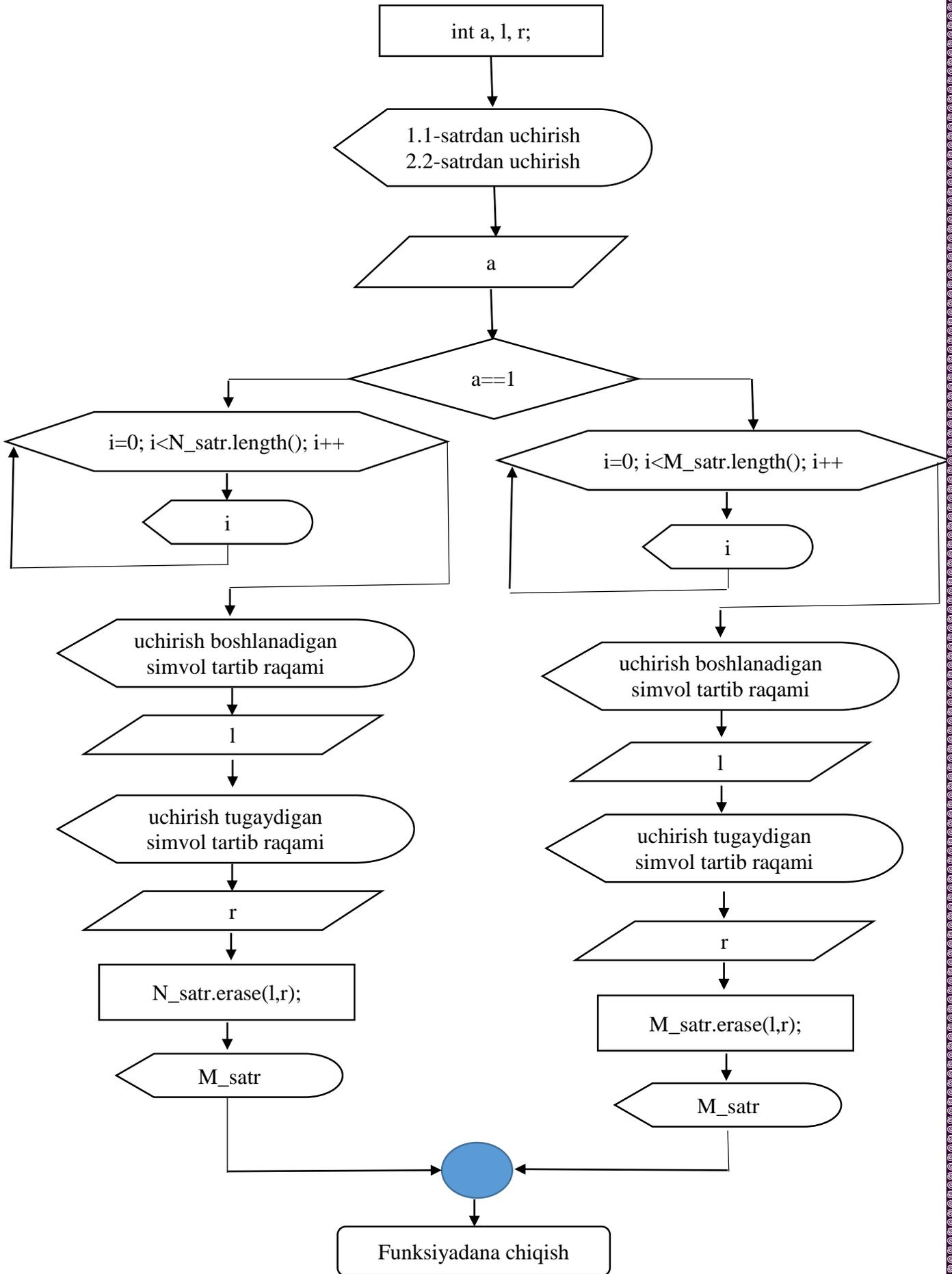
//KonstruktrSatr::Satr()



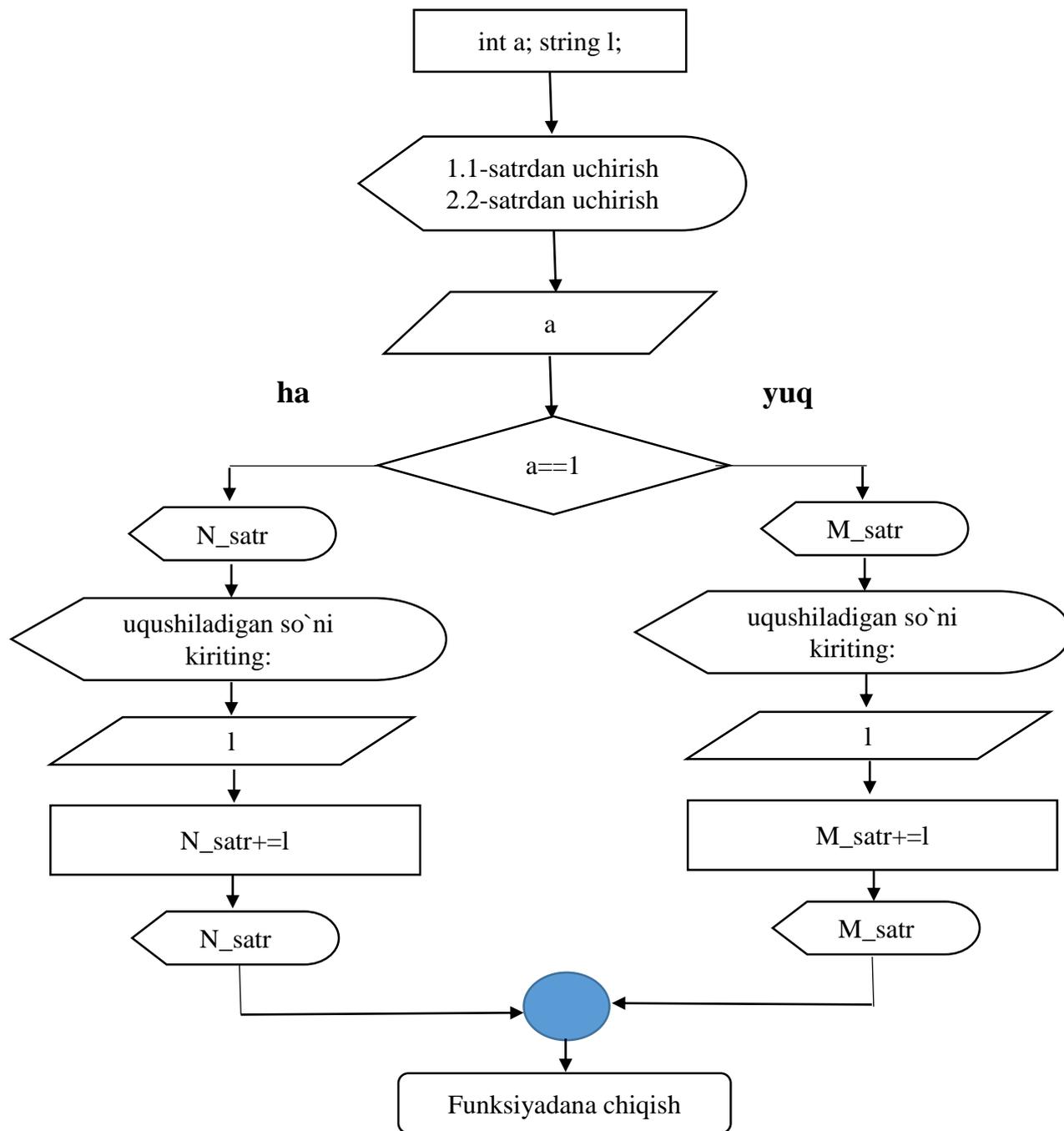
//solishtirishvoid Satr::s_s()



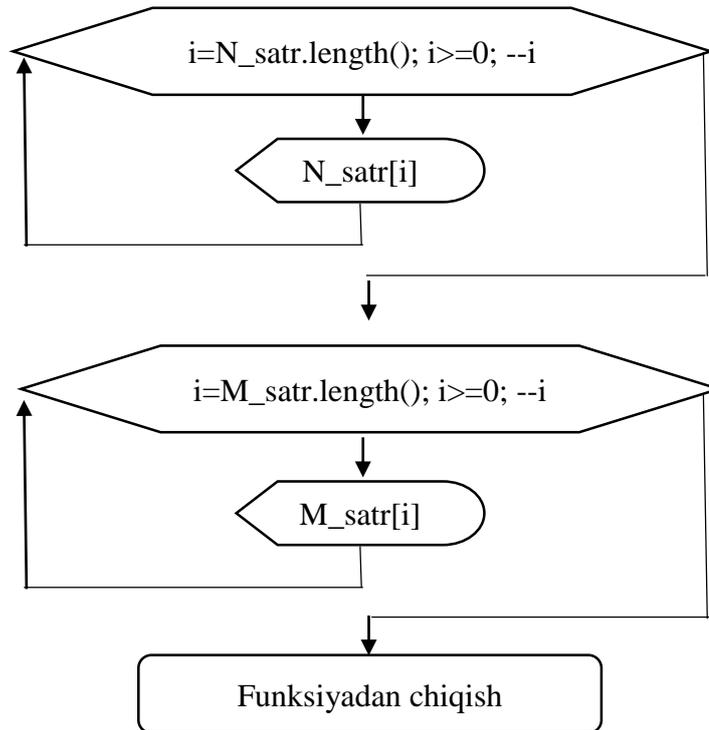
void Satr::s_uc()



/ulashvoid Satr::s_u()



//teskarilashvoid Satr::s_t()



4.3. Algoritm dasturiy kodi va natijalar

```
#include<iostream>
#include<string.h>
using namespace std;
class Satr
{
public:
    Satr();
    ~Satr();
    void s_uc();
    void s_t();
    void s_u();
    void s_s();
private:
    string N_satr, M_satr;
};
Satr::Satr()
{
    string R, K;
    cout<<"1-satrn kiriting: "; cin>>R;
```

```

        cout<<"\n2-satrni kiriting: "; cin>>K;
        N_satr+=R; M_satr+=K;
    }
    Satr::~~Satr( ) { }
    void Satr::s_uc()
    {
        int a, l, r;
        cout<<"1.1-satrdan uchirish\n";
        cout<<"2.2-satrdan uchirish\n";
        cin>>a;
        if(a==1)
        {
            cout<<N_satr<<"\n";
            for(int i=0; i<N_satr.length(); i++)
                cout<<i;
            cout<<"\n";
            cout<<"uchirish boshlanadigan simvol tartib raqami: "; cin>>l;
            cout<<"uchirish tugaydigan simvol tartib raqami: "; cin>>r;
            N_satr.erase(l,r);
            cout<<"\n"<<N_satr<<"\n";
        }
        else if(a==2)
        {
            cout<<N_satr<<"\n";
            for(int i=0; i<M_satr.length(); i++)
                cout<<i;
            cout<<"\n";
            cout<<"uchirish boshlanadigan simvol tartib raqami: "; cin>>l;
            cout<<"uchirish tugaydigan simvol tartib raqami: "; cin>>r;
            M_satr.erase(l,r);
            cout<<"\n"<<M_satr<<"\n";
        }
    }
    void Satr::s_t()
    {
        for(int i=N_satr.length(); i>=0; --i)
            cout<<N_satr[i];
        cout<<"\n";
        for(int i=M_satr.length(); i>=0; --i)
            cout<<M_satr[i];
        cout<<"\n";
    }
    void Satr::s_u()
    {

```

```

int a; string l;
cout<<"1.1-satrdan uchirish\n";
cout<<"2.2-satrdan uchirish\n";
cin>>a;
if(a==1)
{
    cout<<N_satr<<"\n";
    cout<<"qushiladigan so`ni kiriting: "; cin>>l;
    N_satr+=l;
    cout<<N_satr<<"\n";
}
else if(a==2)
{
    cout<<M_satr<<"\n";
    cout<<"qushiladigan so`ni kiriting: "; cin>>l;
    M_satr+=l;
    cout<<M_satr<<"\n";
}
}
void Satr::s_s()
{
    if(N_satr==M_satr) cout<<"satrlar o`zaro teng \n";
    else cout<<"satrlar o`zaro teng emas \n";
}
int main()
{
    Satr Frisky;
    string k;
    cout<<"[exit]-chirish.\n";
    while(true)
    {
        cout<<"\t\tMenu\n";
        cout<<"U.satrlarni ulash:\n";
        cout<<"I.satrni biror qismini uchirish:\n";
        cout<<"T.satrni teskarilash:\n";
        cout<<"S.satrlarni solishtirish:\n";
        cin>>k;
        if(k=="exit") break;
        else switch(k[0])
        {
            case 'U': Frisky.s_u(); break;
            case 'I': Frisky.s_uc(); break;
            case 'T': Frisky.s_t(); break;
        }
    }
}

```

```
        case 'S': Frisky.s_s(); break;
    }
}
system("pause");
return 0;
}
```

V. XULOSA

Kurs ishimni tayorlash davomida EHM ning kelib chiqishi uning utmishi to'g'risida anchagina tushinchaga ega bo'ldim. Dastrulash tillari to'g'risida malumotga egabulub gina qolmay manashu kurs ishimning mavzusida o'zimga kerak bo'lgan va asosiy zarur bo'lgan tushinchalarga ega bo'ldim. C++ tilining ahamiyatini, uning qo'llanish usullarini unda foydalaniladigan tushinchalarhaqida umumiy tushinchalarga ega bo'dim.

Xulosa qilib aytadigan bulsam bu kurs ishimni bajarish davomida ko'p narsalarni o'rgandim jumladan satirlar bilan ishlash ular ustida amallar bajarish va h k. Satirlar bilan ish uchun ko'p funksiyalar bilan tanishdim shulardan satri uzunligini. Aniqlashfunksiyalari, satrlarni nusxalash ,satrlarni ulash ,satrlarni solishtirish, satridagi harflar registrini almashtirish, satrida belgini izlash,satr qismini boshqa satrga qo'shish funksiyalari bilan tanishib ularni chuqurroq urgandim ular ustida turli dasturlar tuzdim va kurs ishimda keltirib utdim C++ tilida satirlar bilan ishlash osonligini bilib xamda tushinib oldim Sinflar C++ ning sinflar (class) bilan ishlashni ularda dasturda foydalanishni urganib oldim. Sinflarda vorislik ,polimorfizm tushinchalari bilan ham tanishdim. Kurs ishimda keltirgan barcha dasturlarimni DEV C++ muhitida tuzganman.

VI. ADABIYOTLAR

1. StenliLippman. YazikprogrammirovaniyeS++. Bazovoykurs. Vilyams - M.: 2014.
2. Sidxarma Rao. OsvoysamostoyatelnoS++ za 21 den. Vilyams - M.: 2013.
3. Nikita Kultin. Microsoft Visual C++ vzadachaxiprimerax. BXV-Peterburg - Peterburg.:2010.
4. B. Straustrup. YazikprogrammirovaniyaS++. Spetsialnoyeizdaniye.-M.:OOO «Binom-Press», 2006.-1104 s.
5. PavlovskayaT.A. S++. Programmirovaniyena yazikevisokogourovnya – SPb.: Piter. 2005.- 461 s.
6. PodbelskiyV.V. YazikS++.- M.; Finansiistatistika- 2003 562s.
7. PavlovskayaT.S. ShupakY.S. S/S++. Strukturnoyeprogrammirovaniye. Praktikum.-SPb.: Piter,2002-240s
8. PavlovskayaT.S. ShupakY.S. S++. Obyektno- oriyentirovannoye programm-mirovaniye. Praktikum.-SPb.: Piter,2005-265s
9. GlushakovS.V., Koval A.V., SmirnovS.V. Yazikprogrammirovaniya S++: Uchebniykurs.- Xarkov: Folio; M.: OOO «Izdatelstvo AST», 2001.-500s.
10. SH.F. Madraximov, S. M. GaynazarovS++ tilida programmalash asoslari. T. 2009.

Internet saytlari.

1. www.ziyonet.uz
2. www.tuit.uz
3. www.tuitkf.uz
4. www.Refiratlar .uz
5. Kitoblar.uz