O’ZBEKISTON RESPUBLIKASI ALOQA, AXBOROTLASHTIRISH VA TELEKOMUNIKATSIYA DAVLAT QO’MITASI

TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI FARG’ONA FILIALI

”Axborot texnologiyalari” kafedrasi

”C++ da dasturlash”
fanidan

KURS ISHI

Bajardi: 612-12 Guruh talabasi

To’rayev Doston

Qabul qildi: Bazarbayev M.

Hashimov A

Farg’ona-2015

Mavzu: C++ dasturlash tilida Uchburchak uchlarining to'g'ri burchak koordinatalari

Reja:

I. Kirish_____

1 Dasturlash haqida

II. Asosiy qism _____

1. **C++ dasturlash tili haqida qisqacha ma'lumot.....**
2. **C++ dasturlash tilining massiv operatori Imkoniyatlari.....**
3. **C++ dasturlash tilida ishlash.....**

II. Dasturiy qism

1. **Dastur vazifasi.....**
2. **Dastur tana qismini ko'rinishi .**
3. **Dasturni C++ dasturlash tilida yaratish bosqichlari.....**

IV. Xulosa _____

V. Foydalanilgan adabiyotlar_____

Kirish

C++ dasturlash tili haqida so'z yuritishdan avval dasturlash haqida ma'lumotga ega bo'lishimiz kerak.

Dasturlash texnologiyasi bo'yicha qisqacha ma'lumot

Dasturlash– dastur yaratishga yo'naltirilgan amallar soxasidir.

Dastur – bu qo'yilgan masalani yechishga olib keluvchi kompyuter buyruqlarining ketma-ketligi.

Ilova – masalaning yechilishini kompyuterda dasturiy amalga oshirishdir.

Dasturiy ta'minot (DT)) – dasturiy maxsulotlar va ularga texnik xujjatlardir.

Dasturiy maxsulot (DM) – ma'lum masalani amalga oshirish uchun mo'ljallangan o/zoro bog'liq dasturlar kompleksi.

Dasturiy ta'minot sifatining xarakteristikalari:

- **Xujjatlashtirilganlik;**
- **Effektivlik;**
- **foydalanishga qulayligi;**
- **mobilligi;**
- **narxi va boshqalar.**

Dastur quyidagi talablarga javob berishi kerak:

1. **Texnik topshiriqqa asosan ishlashi;**
2. **Tezkorlik va xotira bo'yicha effektivlik;**
3. **Ishlatish va takomillashtirish uchun qulay bo'lishi;**
4. **Xatoliklarni aniqlashga moslashgan bo'lishi va boshqalar.**

Translyator dasturni dasturlash tilidan mashina kodlariga o'tkazib beradi.

Translyatorlarning turlari:

- **Interpretator – satrma–satr o'tkazadi va bajaradi. Interpretator – sekin ishlaydi lekin sozlashga qulay.**

- **Kompilyator – dasturni to'lig'icha o'tkazadi va keyin bajaradi. Kompilyator – sozlashga қulay emas, lekin tayyor dasturni tez bajaradi**

Algoritm va dasturlarni avtomatlashtirilmagan loyihalash tarkibi murakkab bo'lmagan nisbatan kichik dasturiy maxsulotlarni ishlab chiqishda ishlatiladi.

Avtomatlashtirilgan loyihalash katta firmalarda dasturiy maxsulotlarning ayrim klasslarini ishlab chiquvchilar kollektivlari tomonidan qo'llaniladi.

Algoritmlar va dasturlarni loyihalash dasturiy mahsulotlarning hayot siklidagi asosiy bosqichlardan biridir.

Avtomatlashtirilgan loyihalash - loyihalash ishlariga ketadigan sarflarni kamaytirish, ularni bajarish muddatlarini qisqartirish, har xil loyihalarda ko'plab marta nusxa olish mumkin bo'lgan tayyor ishlanmalardan foydalanish va dasturiy maxsulot ishlab chiqaruvchilarning katta kollektivining ishini muvofiqlashtirish zarurligi tufayli yuzaga kelgan

Tarkibiy loyihalash – loyihalash masalasini maqsadga yo'naltirilgan holda ayrim tashkil etuvchilarga bo'lib chiqishdir..

Tarkibiy loyihalash quyidagilarni o'z ichiga oladi:

- **yuqoridan pastga yo'naltirilgan loyihalash;**
- **modulli dasturlash;**
- **tarkibiy dasturlash.**

Yuqoridan pastga yo'naltirilgan loyihalash metodi ma'lumotlarni qayta ishlashning umumiy funktsiyasini sodda funktsional elementlarga bo'lib

chiqishni ko'zda tutadi.

Tarkibiy dasturlash dasturiy maxsulotning modulli tarkibiga va asosiy algoritmik tarkiblarga asoslangan.

Obhektga yo'naltirilgan loyihalash quyidagilarga asoslangan:

- obektlarning klasslarini ajratish;

- obektlarning xossalari va ularni o'zgartirish (qayta ishlash) metodlari;

- klasslarning ierarxiyasini hosil qilish, obektlarning xossalari va ularni qayta ishlash metodlarini meros qilish.

Har bir obekt ma'lumotlar va ularni qayta ishlash dasturini o'z ichiga oladi va ma'lum klassga mansub bo'ladi.

Obektga yo'naltirilgan loyihalashning asosiy maqsadi yuqoridan pastga loyihalashning quyidagi kamchiliklarini bartaraf qilishdir:

- ma'lumotlar strukturalariga e'tiborning yetarli emasligi;

- mahlumotlar strukturalari bilan ularni қayta ishlash jarayonlari orasidagi bog'lanishning kuchsiz bog'langanligi.

Obektga yo'naltirilgan loyihalash bir-biriga bog'langan ko'plab obektlarni o'z ichiga oluvchi dasturlarning hajmini va ularni tayyorlashga ketadigan mehnat sarflarini keskin kamaytirish imkoniyatini beradi.

Dasturiy maxsulotlarni yaratish bosqichlari

<u>1. Dasturlash uchun texnik topshiriqni tuzish</u>.

Ushbu bosqichda quyidagilar bajariladi:

Operatsion sistemaning turi (OS - MS DOS, Windows, Windows NT) tanlanadi,

dastur ishlashining tarmoq varianti zarurligi ko'rib chiqiladi,

dastur tuzishning zarurligi aniqlanadi,

masalani yechish usuli tanlanadi va algoritmi tuziladi,

foydalanuvchining interfeysiga talablar aniqlanadi.

<u>Texnik loyiha</u>

Ushbu bosqichda quyidagilar bajariladi:

Ma'lumotlarni qayta ishlashning batafsil algoritmi tuziladi,

Dasturiy ta'minotning tarkibi aniqlanadi,

Dasaturiy maxsulotning ayrim dasturiy modullardan tashkil topgan ichki tarkibi ishlab chiqiladi,

Dasturiy vositalarni ishlab chiqish vositalari tanlanadi.

<u>Ishchi xujjatlar (ishchi loyiha)</u>.

Ushbu bosqichda quyidagilar amalga oshiriladi:

Dasturiy modullar ishlab chiqiladi,

Dastur tayyorlanadi,

Dasturiy maxsulot sozlanadi,

Dasturiy modullar va vositalarning ishga yaroqligi sinab ko'riladi,

Dasturiy maxsulotning topshiriqqa mos kelishini tekshirib ko'rish uchun nazorat misoli tayyorlanadi.

Dasturiy maxsulotlarning tuzilishi

Dasturiy maxsulot o'zoro bog'langan dasturiy modullardan tashkil topgan bo'ladi.

Modul – dasturning mustaqil qismi bo'lib u boshqa modullardan avtonom ravishda berilgan funktsiyalarni ta'minlaydi va ma'lum vazifaga ega bo'ladi.

Modullarning quyidagi turlarini ko'rsatish mumkin:

***bosh modul* – dasturiy maxsulotni ishga tushirishni boshqaradi;**

***boshqaruvchi modul* – boshqa modullarni chaqirishni ta'minlaydi;**

***ishchi modullar* - qayta ishlash funktsiyalarini bajaradi;**

***servis modullar va bibliotekalar* – xizmat ko'rsatish funktsiyalarini amalga oshiradi.**

Har bir modul mustaqil saqlanuvchi fayl ko'rinishida bo'ladi.

Dasturning ichki tuzilishga ega bo'lishi uni loyihalash, dasturlash, sozlash va o'zgartirishlar kiritish uchun qulay bo'lishini ta'minlaydi.

Dialog rejim

Ko’pgina dasturiy maxsulotlar dialog rejimida ishlaydi.

Dialog rejimda foydalanuvchi tomonidan funktsiyalar ishga tushiriladi, obektlarning xossalari o’zgartiridadi va h.k.

Menyular ierarxik bo’lishi va ost menyularni o’z ichiga olishi mumkin.

Dialog tizimlarning tarkibi:

***Menyu* – foydalanuvchiga ro’yҳatdan algpternativ funktsiyalarni tanlash imkoniyatini beradi;**

Menyuda ichki (ost) meny u ham bo’lishi mumkin.

***Sўrov-javob amali*– ro’yxatdan tanlab olinishi mumkin bo’gan qiymatlar yoki Ҳa/Y o’q turdagi javoblar;**

Format bo’yicha so’rov – kalit so’zlar yoki iboralar yordamida.

Dialog jarayoni yaratilgan *stsenariyaga asosan boshқariladi.* Dialog jarayoni uchun quyidagilar aniqlanadi:

Dialogning boshlanish momenti;

Dialogni boshlovchi– foydalanuvchi yoki dasturiy maxsulot;

Dialogning parametrlari va mazmuni – axborotlar,

menyuning tarkibi sostav, ekran formalari;

Dasturiy maxsulotning dialog tugashiga reaktsiyasi.

Dialog jarayonni va foydalanuvchining interfeysini yaratish uchun dasturlarni ishlab chiqishning obektga yo’naltirilgan vositalaridan foydalaniladi. Ularning tarkibiga quyidagilar kiradi:

Menyu qurgich (bosh menyu va unga biriktirilgan ost menyularni yaratish uchun);

Ekran formalarining konstruktori (ekran orq ali kiritish formatlarini ishlab chiqish va mahlumotlarni taqrirlash uchun).

Dialog oynada quyidagi boshqarish elementlari bo’lishi mumkin:

Axborot matnlari;

Foydalanuvchining axborotlarini kiritish uchun maydonlar;

Tanlash uchun mumkin bo’lgan alternativlar ro’yxatlari;
tugmalar, ulab uzgichlar va h.k.

Foydalanuvchining grafik interfeysi

Grafik interfeys zamonaviy dasturiy maxsulotlar uchun zaruriy komponent bo’lib hisoblanadi va ochiluvchi menyular ko’rinishida quriladi. Grafik interfeys yordamida foydalanuvchi ekran formalari bilan ishlaydi. Ekran formalarida boshqarish elementlari, asboblar panellari va qayta ishlash komandalari bo’lishi mumkin.

Menyu dastur bajaradigan funktsiyalarga mos holda tushunarli punktlarga ega bo’lishi kerak.

Tarkibiy loyihalash va dasturlash

Tarkibiy loyihalash quyidagilarni o’z ichiga oladi:
Yuқoridan pastga loyiҳalash,
Modulli dasturlash,
tarkibiy dasturlash.

Yuqoridan pastga loyihalashda axborotlarni qayta ishlash funktsiy asi yuqoridan pastga yo’nalishda sodda funktsional elementlarga bo’lib chiqiladi. Natijada ilova algoritmining funktsional tuzilishi hosil hilinadi. Unda quyidagilar o’z aksini topadi:
maqsad-ost maqsad;
қo’yilgan maqsadni amalga oshirishni ta’minlaydigan ilovalarning tarkibi va o’zaro boғlanishlari;
axborotlarni qayta ishlash funktsiyalari.

Modul mantiqiy boғlangan elementlar majmuidir. Modul boshqa modullar yoki dasturlar foydalanishi uchun mo’ljallangan bo’ladi.
Modul tayyor dasturlarni saqlash uchun mo’ljallangan.

Modulning o’zi bajariluvchi dastur bo’lmaydi. Uning obektlarini boshqa dasturlar (protseduralar, funktsiyalar) ishlatadi.

Modulda bitta kirish va bitta chiqish bo'ladi. Kirishiga boshlanғich mahlumotlar beriladi, modul ularni qayta ishlaydi va natijalarni chiqishga uzatadi, ya'ni quyidagi printsipni amalga oshiradi

IPO (Input – process – Output) – kirish -jarayon- chiqish;

Har bir modul o'zidan yuqoridagi modul tomonidan chaqiriladi va ishini tugatgandan so'ng boshqarishni o'zini chaqirgan modulga topshiradi.

Algoritm – masalani elementar amallarga bo'lib yechish usulining formal tavsifi.

Algoritmik dasturlash- dasturni modullar ketma-ketligiga bo'lish. Bunda har bir modul bir yoki bir necha amalni bajaradi.

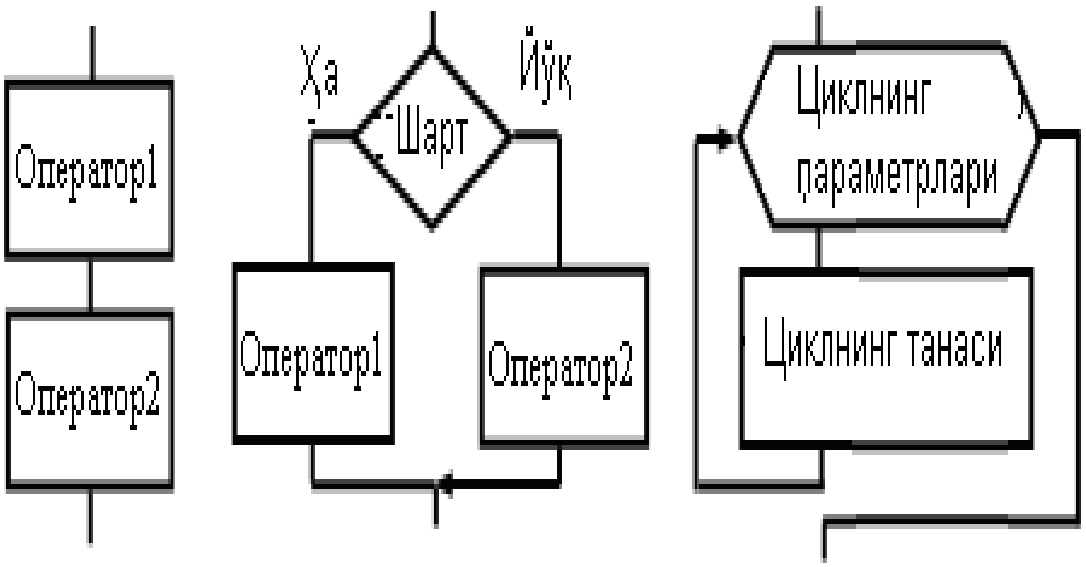
Tarkibiy dasturlash modulli tarkibga va boshqaruvchi tarkiblarga asoslangan.

Boshqaruvchi tarkiblarning turlari quyidagilar:

Ketma-ketlik;

Alternativa (tanlash sharti);

Sikl.



Dasturlashda dastur mukammal va qulay bo'lishi uchun dasturchi eng avvalo qulay va vazifaga mos dasturlash tilini tanlashi kerak. Qo'shimcha dasturlash tlini xisoblash sistemasi toki dasturchi foydalanuvchiga asosan tanlangan bo'ladi.

Agar dasturchi tanlashga ega bo'lib qolsa u albatta vazifaga mos eng yuqori dasturlash tilini tanlashi kerak. Agar tanlangan dasturlash tili berilgan vazifaga mos kelmasa dasturlashda va sozlashda muammolar kelib chiqishi mumkin. Tanlangan til loyihalashga ham ancha ta'sir ko'rsatadi.

Dasturlashda asosiy bosqichlardan biri Universallikdir. Universallik deb ya'ni ma'lumotlar turkumiga bog'liq bo'lmagan dasturlarni ataymiz.

Parametrlar sifatida kontaktlar o'rniga o'zgaruvchilardan foydalaniladigan dasturni universal dastur deymiz. O'zgaruvchilarni kontakt o'rnida qo'llanishi mumkin bo'lgan sharoitlarni ko'rsatib o'tamiz.

1. **Ro'yxatlar, massivlar va tablitsalar o'lchami.**
2. **Fizik konstanta va foizlar, oraliqlar.**
3. **"kiritish-chiqarish" moslamasini belgilanishi.**

Dasturlashni universalligi loyihalashda vaqtdan yutishi va undan foydalanishda keng sharoitlar ochib berishdir.

Universal protseduralar dastur kutubxonasiga keying dasturlarda ham foydalanish uchun kiritiladi.

Universallikni qidirishda asosan keyinchalik qo'llaniladigan modifikatsiya va variantlarni aniqlashda ham yordam beradi.

Kutubxonalar.

Dasturni samaradorligini oshirish, sozlash va tekshirishni yengillashtirsh xamda dastur yaratishni qisqartirish uchun dastur kutubxonalari qo'llaniladi. Kutubxona uchue mo'lljallangan kichik dasturlar katigoriyasiga dasturlash tiliga ega bo'lgan funktsiya yoki kichik dasturlarni o'z ichiga oladi. Bu katigoriyalarga barcha standart funktsiyalar (sin, cos va xakazo), kiradi.

Ikkinchi funktsiya va kichik dasturlar manbaisi sizning hisob-lash sistemangizdan tashkarida joylashgan. Foydalunuvchiga u yoki bu sistemada bajarilgan dasturlar to'g'ri keladi. Kutubxona dasturlarda foydalanish sabablaridan biri – ishonchlikni oshirish va dasturlashdagi qiyinchiliklarni kamaytirishdir. kutubxona dasturlarining uchinchi manbaisi bu dasturlar va hisoblash mashinalari to'g'risidagi kitoblardir.

"Kiritish - chiqarish" formatlari.

Ma'lumotlarning "kiritish - chiqarish" formatlari loyihalash etapining bir qismi hisoblanadi. Kiritish formatlari foydalanuvchi

uchun eng yuqori qulaylik va eng kam xatoliklar bo'lishini hisobga olgan xolda bajariladi. CHiqish maolumoti tashqi manbalar ta'sirisiz solishtirilishi kerak. U uz ichiga:

1. **CHiqish yezuvini solishtirish.**
2. **Yezuv funktsiyasi yoki tavsifi.**
3. **Sana.**
4. **Varaklarning tartib raqamini olishi kerak.**

Yaxshi o'ylab topilgan chiqish formatlari tuzuvchiga sozlashda va sinashda katta yordam beradi.

Maqsadni aniqlash.

Loyihalash vaqtida dasturchi o'z oldiga ma'lum bir aniq maqsadlar qo'yishi kerak. Quyida ularning bir nechasi keltirilgan.

1. **Eng yuqori ishonchlilik.**
2. **Ma'lum bir ishning ayrim hajmini aniq bir sanaga bajarish.**
3. **Loyihalash uchun eng oz vaqt sarflash yoki tan - narxini or-zonligi.**
4. **Foydalanishdagi qulayligi va oddiyligi.**
5. **Samaradorligi (xotira xajmi yoki tez ishlashi).**
6. **Takommilashtirishni hisobga olish.**
7. **Universallik.**

Ayrim xollarda bitta dastur ustidan Ishlayotgan ikkita dastu chi har xil maqsadlarni ko'zlab ishlashadi.

Biri xotira xajmini kichraytirishga xarakat qilsa, ikkinchisi loyihalash uchun sarf - xarajatni kamaytirishni hisoblaydi. Maqsadlarni oldindan va aniq qo'ying.

Dasturiy taominlash sistemasidan foydalanish vaqtida yangidan yangi maqsadlar yuzaga keladi. Yangi maqsadlarga asosan tayer dasturni o'zgartirish - pogonaga (tiqishtirish) deb ataladi.

Dasturga yangi funktsiyalar va eskisini yangilashda pog'onaga qo'llaniladi.

Murakkablik.

Xammaga ma'lumki, dasturda 2 ta asosiy parametrlar: xotira va sanalari bor.

Yahna murakkablik parametrini ham hisobga olish kerak. Dastur qanchalik murakkab bo'lsa, uni tekshirish, nazorat qilish, sinash, tartiblash shunchalik qiyin bo'ladi. Murakkablik parametrlarini chegaralaganimizda biz boshqarishni ancha yengillashtiramiz. Xozirgi vaqtda murakkablikni chegaralovchi usullar va metodlar bor. Murakablikni boshqarish metodlari nima? Ular qanday bajaradi? Dasturlashda ularni qanday qo'llaniladi.

Sifatni oshirishning 2 ta yo'li bor: biri kattik va chuqur tekshirishlar o'tkazish, ikkinchisi har bir loyihalash jarayonida sistemalarga eng yuqori ishonchlilik bilan taominlash.

Murakkablikni boshqarish metodida asosan jarayon yoki tarkib bir necha parametrlarga bo'linadi va aniq bir funktsiyani bajarish uchun foydalaniladi . Bunday metodlar barcha Ishlab chiqarishda, administrativ ishlarda va tashkilotlarda foydalaniadi. Murakkablikni boshqarish dasturlashdagi asosiy funktsiyalardan biridir.

Asosiy qism

Barcha dasturlash tillarini o'z vazifasiga va qulayliklariga ko'ra ishlatiladi. Hozirgi paytda dasturlash tillaridan eng keng

tarqalganlarini siz yuqoridagi jadvalda ko'rdingiz. Ularning eng yuqorisida turgan Java dasturlash tili hozirgi paytda eng ko'p tarqalgan dasturlash tilidir. Bunga sabab ushbu dasturlash tilida dastur juda mukammal va puxta tuziladi. Dasturchiga juda ko'p qulayliklari bor. Java dasturlash tili bilan bir qatorda hozir yana ko'p tarqalgan ko'plab dasturchilar foydalanayotgan dasturlash tillaridan yana biri bu C++ dasturlash tilidir.

C++ tili C tiliga asoslangan ushbu tilni 1980-yillar boshida Bjarne Stroustrup tomonidan C ga asoslangan holda tuzildi. Boshida Straustrup yangi tilni " C with Class " deb nomladi. Keyinchalik 1983 – yilda bu nom C++ ga o'zgardi. C++ oddiy C dasturlash tiliga yangi imkoniyatlarni, funksiyalarni, ob'ektga yo'naltirlganlik xususiyatlarini olib kirdi. C++ dasturlash tili C dasturlash tili asosida qurilgan va uning barcha imkoniyatlarini o'zida saqlaydi. Qisqacha qilib aytadigan bo'lsak C++ da yozilgan dastur kodida C ning kodlarini qo'llash mumkin. Dress Libertining fikricha C++ va Java dasturlash tillari bir-biriga o'xshash "Bulardan birini o'rgangan dasturchi ikkinchisini 90% o'zlashtiradi" deb yozadi Liberti o'zining "C++ за 30 день" nomli kitobida. Umuman C++ dasturlash tilini o'rgangan odam Java va C# dasturlash tiullarini 90% o'zlashtiradi. C++ dasturlash tili C dasturlash tili imkoniyatlarini oshirish maqsadida ishlab chiqilgan. Bu tillar asosan tizimli programmalashda juda qo'l keladi. Java dasturlash tili esa C va C++ dasturlash tillarida bor bo'lgan afzallik va kamchiliklarni inobatga olgan holda ishlab chiqilgan yangi til hisoblanadi. Dastlab Javada tuzilgan dastur, har qanday platformada ishlashda maqsad qilingan holda ishlab chiqilgan (1991 yillarda) internet rivojlanishi bilan, Javani ishlab chiqaruvchilar guruhi asosiy e'tiborni webga qaratishadi. Ko'p mutaxassislar fikricha internetning rivojlanishi bir vaqtga to'g'ri kelgani, Javani muvaffaqiyatiga sabab

bo’lgan. Ya’ni Java web texnalogiyalarda juda keng qo’llaniladi.keyinchalik Microsoft firmasi tomonidan javaga raqobatchi sifatida C# dasturlash tili (2001 yili) ishlab chiqildi. C# dasturlash tli Delphi, C++ va Javada bor bo’lgan afzallik va kamchiliklarni inobatga olgan holda ishlab chiqilgan. Ba’zi mutaxassislar C# dasturlash tilini kelajak texnalogiyalari tili deb hisoblashadi. Shunday bo’lishiga qaramasdan hozircha C++ va Java dasturlash tillari o’z mavqeini yo’qotgani yo’q. umuman olganda har bir dasturlash tiilini o’ziga yarasha afzallik va kamchiliklari mavjud va ular ishlatiladigan o’rni bor. C++ juda ko’p qo’shimchalarni o’z ichiga olgan, lekin eng asosiysi u ob’ektlar bilan dasturlashga imkon beradi. Dasturlarni tez va sifatli yozish kunda katta ahamiyat kasb etmoqda. C++ dasturlash tili funksiya ob’ektlarning juda boy kutubxonasiga ega. Ya’ni C++ dasturlash tilida dasturlashni o’rganish ikki qismga bo’linadi. Birinchisi bu C++ tilini o’zini o’rganish, ikkinchisi esa C++ ning standart kutubxonasidagi tayyor ob’ekt va funksiyalarni qo’llashni o’rganishdir.

MASSIVLAR

Bu qismda dasturdagi ma'lumot strukturalari bilan tanishishni boshlaymiz.

Dasturda ikki asosiy tur ma'lumot strukturalari mavjuddir. Birinchisi statik, ikkinchisi dinamikdir. Statik deganimizda hotirada egallagan joyi o'zgarmas, dastur boshida beriladigan strukturalarni nazarda tutamiz. Dinamik

ma'lumot tiplari dastur davomida o'z hajmini, egallagan hotirasini o'zgartirishi mumkin.

Agar struktura bir hil kattalikdagi tiplardan tuzilgan bo'lsa, uning nomi massiv (array) deyiladi. Massivlar dasturlashda eng ko'p qo'laniladigan ma'lumot tiplaridir. Bundan tashqari strukturalar bir necha farqli tipdagi o'zgaruvchilardan tashkil topgan bo'lishi mumkin. Buni biz klas (Pascalda record) deymiz. Masalan bunday strukturamiz ichida odam ismi va yoshi bo'lishi mumkin.

Bu bo'limda biz massivlar bilan yaqindan tanishib o'tamiz. Bu bo'limdagi massivlarimiz C uslubidagi, pointerlarga (ko'rsatkichlarga) asoslan strukturalardir. Massivlarning boshqa ko'rinishlarini keyingi qismlarda o'tamiz.

Massivlar hotirada ketma-ket joylashgan, bir tipdagi o'zgaruvchilar guruhidir.

Alohida bir o'zgaruvchini ko'rsatish uchun massiv nomi va kerakli o'zgaruvchi indeksini yozamiz. C/C++ dagi massivlardagi elementlar indeksi har doim noldan boshlanadi. Bizda char tipidagi m nomli massiv bor bo'lsin. Va uning 4 dona elementi mavjud bo'lsin. Shemada bunday ko'rsataylik:

m[0] -> 4

m[1] -> -44

m[2] -> 109

m[3] -> 23

Ko'rib turganimizdek, elementga murojaat qilish uchun massiv nomi va [] qavslar ichida element indeksi yoziladi. Bu yerda birinchi element qiymati 4, ikkinchi element - 1 nomerli indeksda -44 qiymatlari bor ekan. Ohirgi element indeksi n-1 bo'ladi (n - massiv elementlari soni).

[] qavslar ichidagi indeks butun son yoki butun songa olib keluvchi ifoda

bo'lmog'i lozim. Masalan:

...

```
int k = 4, l = 2;
m[ k-l ] = 77; // m[2] = 77
m[3] *= 2; // m[3] = 46
double d = m[0] * 6; // d = 24
cout << m[1]; // Ekranda: -44
```

...

Massivlarni ishlatish uchun ularni e'lon qilish va kerak bo'lsa massiv elementlarini initsalizatsiya qilish kerak. Massiv e'lon qilinganda kompilyator elementlar soniga teng hajmda hotira ajratadi. Masalan yuqorida qo'llanilgan char tipidagi m massivini e'lon qilaylik.

```
char m[4];
```

Bu yerdagi 4 soni massivdagi elementlar miqdorini bildiradi. Bir necha massivni e'londa bersak ham bo'ladi:

```
int m1[4], m2[99], k, l = 0;
```

Massiv elementlari dastur davomida initsalizatsiya qilishimiz mumkin, yoki boshlang'ich qiymatlarni e'lon vaqtida, {} qavslar ichida ham bersak bo'ladi.

{} qavslardagagi qiymatlar massiv initsalizaytsiya ro'yhati deyiladi.

```
int n[5] = {3, 5, -33, 5, 90};
```

Yuqorida birinchi elementning qiymati 3, ikkinchiniki 5 ... ohirgi beshinchi element qiymati esa 90 bo'ldi. Boshqa misol:

```
double array[10] = {0.0, 0.4, 3.55};
```

Bu yerdagi massiv tipi double bo'ldi. Ushbu massiv 10 ta elementdan iboratdir.

{} qavslar ichida esa faqat boshlangich uchta element qiymatlari berildi.

Bunday holda, qolgan elementlar avtomatik tarzda nolga tenglashtiriladi. Bu
yerda aytib o'tishimiz kerakki, {} qavslar ichida berilgan boshlangish qiymatlar soni massivdagi elementlar sonidan katta bo'lsa, sintaksis hatosi vujudga keladi. Masalan:

```
char k[3] = {3, 4, 6, -66, 34, 90}; // Hato!
```

Uch elementdan iborat massivga 6 dona boshlangich qiymat berilyapti, bu hatodir. Boshqa misolni ko'rib chiqaylik:

```
int w[] = {3, 7, 90, 78};
```

w nomli massiv e'lon qilindi, lekin [] qavslar ichida massivdagi elementlar soni berilmadi. Bunday holda necha elementga joy ajratishni kompilyator {} qavslar ichidagi boshlangich qiymatlar miqdoriga qarab biladi. Demak, yuqoridagi misolda w massivimiz 4 dona elementdan iborat bo'ladi.

E'lon davridagi massiv initsalizatsiya ro'yhati dastur ijrosi vaqtidagi initsalizatsiyadan ko'ra tezroq ishlaydigan mashina kodini vujudga keltiradi.

Bir misol keltiraylik.

```
// Massivlar bilan ishlash.
# include <iostream.h>
# include <iomanip.h>
const int massiv = 8; // massiv kattaligi uchun konstanta
int k[massiv];
char c[massiv] = {5,7,8,9,3,44,-33,0}; // massiv initsalizatsiya ro'yhati
int main()
{
for (int i = 0; i < massiv; i++) {
k[i] = i + 1; // dastur ichida inisalizatsiya
}
```

```
for (int j = 0; j < massiv; j++) {
cout << k[j]
<< setw(4)
<< c[j]
<< endl;
}
return (0);
}
```

Ekranda:

1 5

2 7

3 8

4 9

5 3

6 44

7 -33

8 0

Yuqorida <iomanip.h> faylini dasturimizga kiritdik. Bu e'lon faylida standart kirish/chiqish oqimlari bilan ishlaydigan buyruqlar berilgan. Dasturimizda qo'llanilgan setw() manipulyatori chiqish oqimiga berilayatgan ma'lumotlarning eng kichik kengligini belgilaydi, biz setw() parametrini 4 deb berdik, demak c[] massivi elementlari 4 harf kenglikda ekranga bosiladilar. Agar kenglik kamlik qilsa, u kattalashtiriladi, agar bo'sh joy qolsa, elementlar chapga yondashilib yoziladi. Biz <iomanip.h> va manipulyatorlarni keyinroq to'la ko'rib chiqamiz.

Misolimizda massiv nomli konstantani qo'lladik. Uning yordamida massiv

chegaralarini va for strukturasidagi chegaraviy qiymatlarni berdik. Bunday
o'zgarmasni qo'llash dasturda hatoga yo'l qo'yishni kamaytiradi. Massiv chegarasi o'zgarganda, dasturning faqat bir joyiga o'zgarish kiritiladi.
Massiv hajmi e'lonida faqat const sifatli o'zgaruvchilar qo'llanilishi mumkin.
Massivlar bilan ishlaganda eng ko'p yo'l qoyiladigan hato bu massivga
0 dan kichkina va (n-1) dan (n: massivdagi elementlar soni) katta indeks bilan
murojaat qilishdir. Bunday hato dastur mantig'i hatosiga olib keladi.
Kompilyator bu turdagi hatolarni tekshirmaydi. Keyinroq o'zimiza yozgan massiv klaslarida ushbu hatoni tekshiriladigan qilishimiz mumkin.
10 ta sonning tushish ehtimilini ko'rsaturvchi dastur yozaylik.

```
// Ehtimollar va massivlar
# include <iomanip.h>
# include <iostream.h>
# include <time.h>
# include <stdlib.h>
int main ()
{
const int massivHajmi = 10;
int m[massivHajmi] = {0}; // hamma 10 ta element
// 0 ga tenglashtirildi
srand( time(NULL) );
for(int i = 0; i < 1000; i++) {
++m[ rand() % 10 ];
}
for(int j = 0; j < massivHajmi; j++) {
```

cout << j << setw(4) << m[j] << endl;

}

return (0);

}

Ekranda:

0 96

1 89

2 111

3 97

4 107

5 91

6 100

7 118

8 99

9 92

Ko'rib turganimizdek, sonlarning tushish ehtimoli nisbatan tengdir. Albatta,

bu qiymatlar dasturning har yangi ishlashida o'zgaradi.

++m[rand() % 10];

Yozuvi bilan biz massivning rand() % 10 indeksli elementini birga oshirmoqdamiz. Bunda rand () % 10 ifodasidan chiqadigan qiymatlar [0;9] ichida

yotadi.

Satrlar, yani harflar ketma-ketligi ("Toshkent", "Yangi yilingiz bilan!"...) C/C++ da char tipidagi massivlar yordamida beriladi. Bunday satrlar bilan islovlar juda tez bajariladi. Chunki ortiqcha tekshirishlar bajarilmaydi. Bundan tashqari C++ da ancha rivojlangan String klasi mavjuddir, u oddiy char

bilan berilgan satrlardan ko'ra qulayroqdir. Lekin ushbu klas ko'proq joy egallaydi va massivli satrlardan ko'ra sekinroq ishlaydi. String klasini keyingi qismlarda o'tamiz. Qolaversa, satrlar bilan ishlash uchun biz o'zimiz

klas yoki struktura yozishimiz mumkin. C dan meros bo'lib qolgan satrlar ustida amallar bajarish uchun biz dasturimizga <string.h>

(yangi ismi <cstring>) e'lon faylini kiritishimiz kerak. Ushbu e'lon faylida berilgan funksiyalar bilan keyingi bo'limda ishlaymiz.

Harflar, yani literalar, aytib o'tganimizdek, C++ da char tipi orqali beriladi. Literalar apostroflarga ('S', '*' ...) olinadi. Satrlar esa qo'shtirnoqlarga olinadi. Satrlar e'loniga misol beraylik.

```
char string[] = "Malibu";
char *p = "Ikkinchi!?!";
```

Satrlarni yuqoridagi ikkita yo'l bilan initsalizatsiya qilsa bo'ladi.

Satrlar ikkinchi uslubda e'lon qilinganda, yani pointer mehanizmi qo'llanilganda, ba'zi bir kompilyatorlar satrlarni hotiraning konstantalar saqlanadigan qismiga qo'yadi. Yani ushbu satrlarga o'zgartirish kiritilishi ta'qiqlanadi. Shu sababli satrlarni hardoim o'zgartirish imkoni bo'lishi uchun ularni char tipidagi massivlar ko'rinishida e'lon qilish afzaldir.

Satrlar massiv yordamida berilganda, ularning uzunligi noma'lumdir.

Shu sababli satrning tugaganligini bildirish uchun satr ohiriga mahsus belgi

nol literasi qo'yiladi. Uning dastursa belgilanishi '\0' ko'rinishga ega.

Demak, yuqorida berilgan "Malibu" satriga ohiriga '\0' belgisi qo'shiladi, yani

massivning umumiy uzunligi "Malibu":6 + '\0':1 = 7 ta char tipidagi elementga

teng bo'ladi. Satrlarni massiv initsalizatsiya ro'yhati ko'rinishida ham

bersak bo'ladi:

```
char c[6] = {'A', 'B', 'C', 'D', 'E' , '\0'};
...
cout << c;
...
```

Ekranda:

```
ABCDE
```

Biz cout bilan c ning qiymati ekranga bosib chiqardik. Aynan shunday klaviaturadan ham o'qib olishimiz mumkin:

```
char string[80];
cin >> string;
```

Eng muhimi satr bilan '\0' belgisi uchun yetarli joy bo'lishi kerak.

Satrlar massiv yordamida berilganligi uchun, alohida elementlarga indeks orqali yetishish mumkin, masalan:

```
char k[] = "Bahor keldi, gullar ochildi.";
for (int i = 0; k[i] != '\0'; i++)
if ( (i mod 2) == 0 ) // juft sonlar uchun haqiqat bo'ladi
cout << k[i] << " ";
```

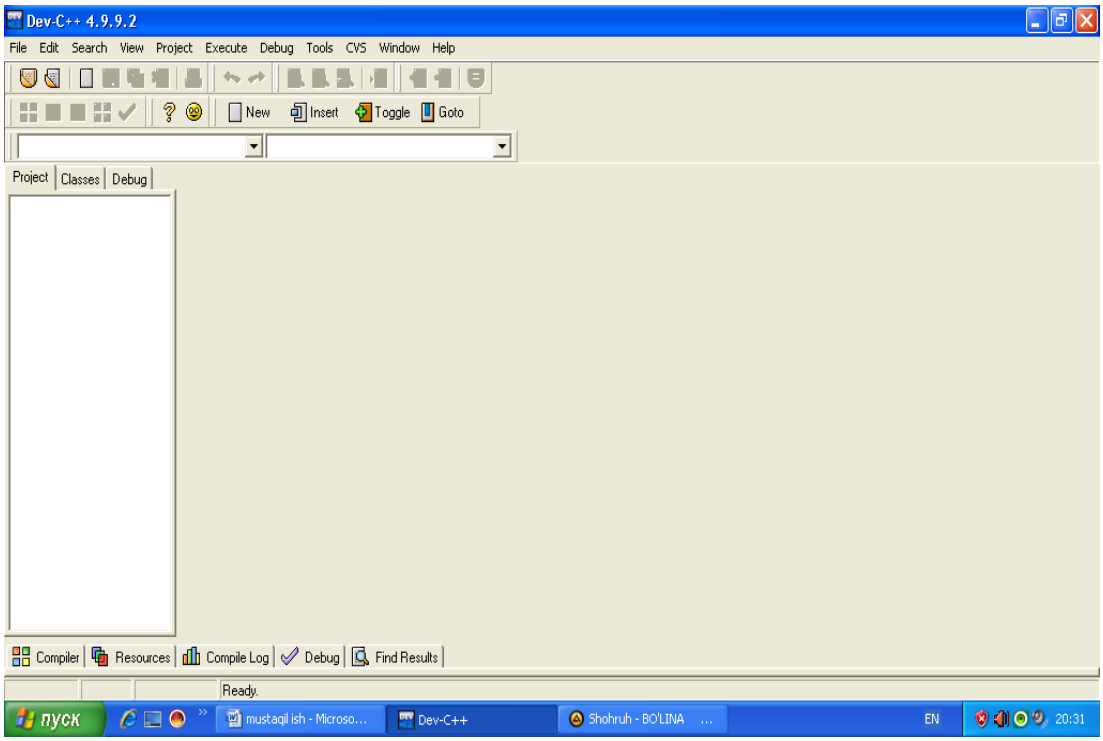
Ekranda:

```
B h r k l i u l r o h l i
```

Yuqoridagi misolda, sikl tugashi uchun k[i] element '\0' belgiga teng bo'lishi kerak.

C++ dasturlash tilida ishlash

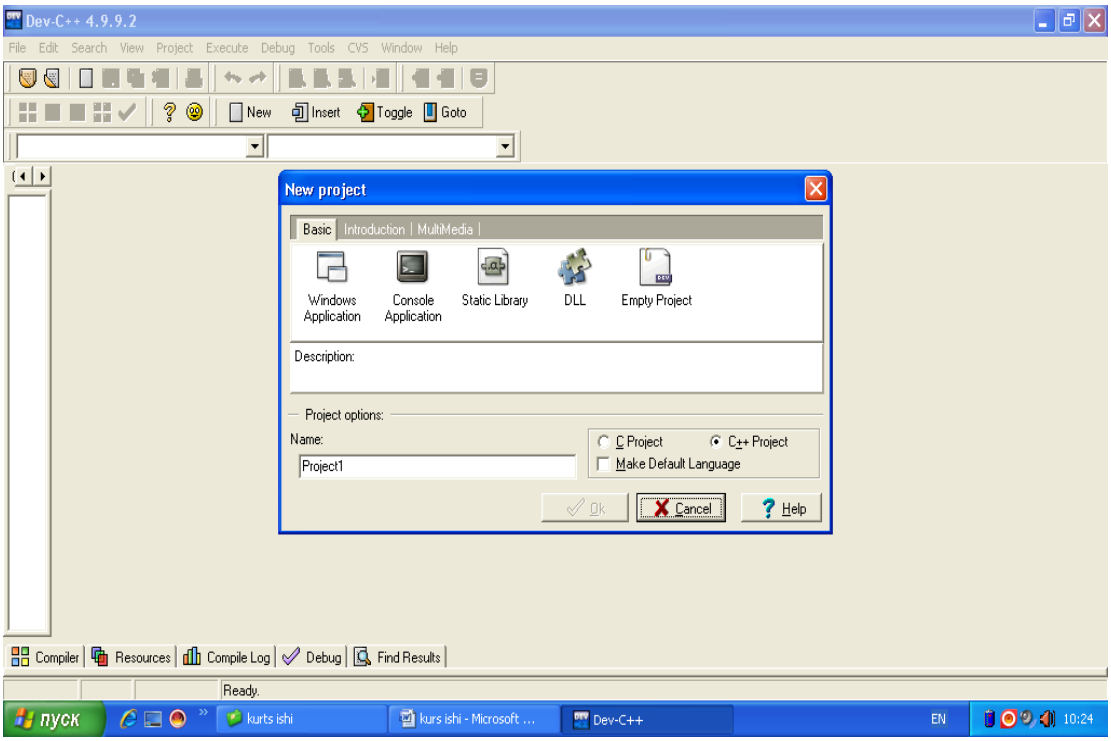
Biz C++ dasturini ishga tushirganimizda quyidagi oynaga ega bo’lamiz.



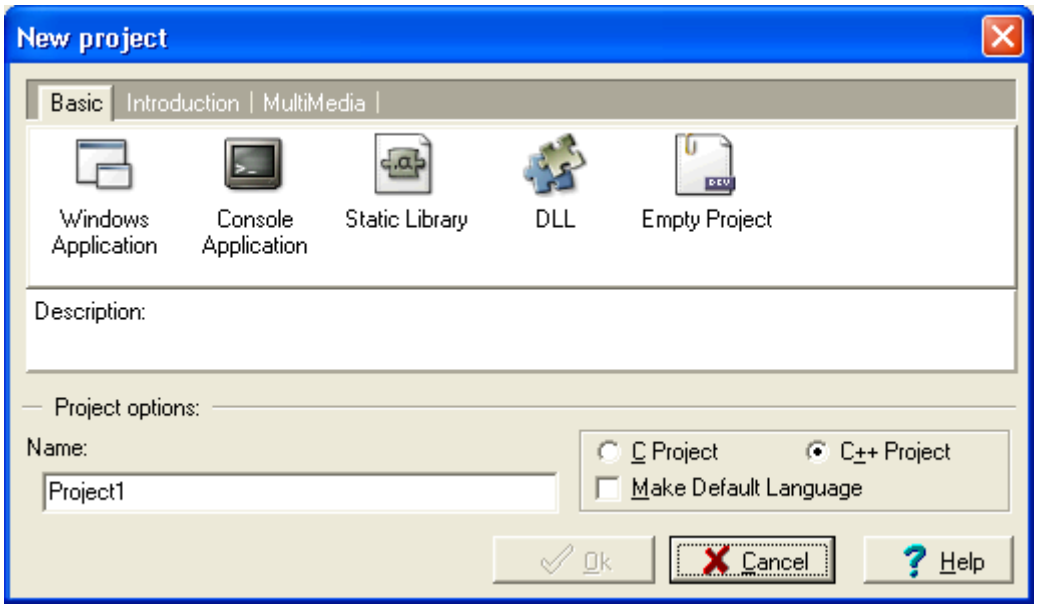
Biz ko’rib turgandek bu dasturni ishchi oynasi hisoblanadi. U quyidagi qismlardan tashkil topgan.

1. **Menyular qatori**
2. **Uskunalarning ikki yoki undann ko’p paneli**
3. **Formulalar qatori**

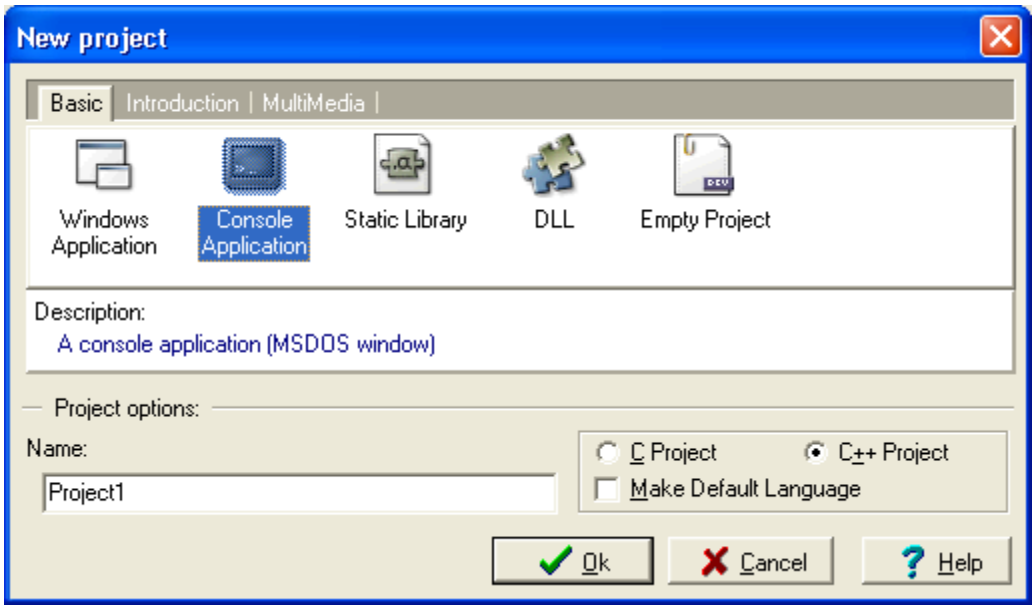
Dasturda ishlashimiz uchun bizga dasturning asosiy ya’ni tana qismini tuzish uchun bizga asosiy ishchi oyna kerak biz uni menyular qatorining birinchi menyusi bo’lgan <File> mrnyusida sichqonchani chap tugmasini bir marta bosamiz va <New> undan <Project> ga sichqonchani chap tugmasini bir marta bosamiz quyidagi oynaga ega bolamiz.



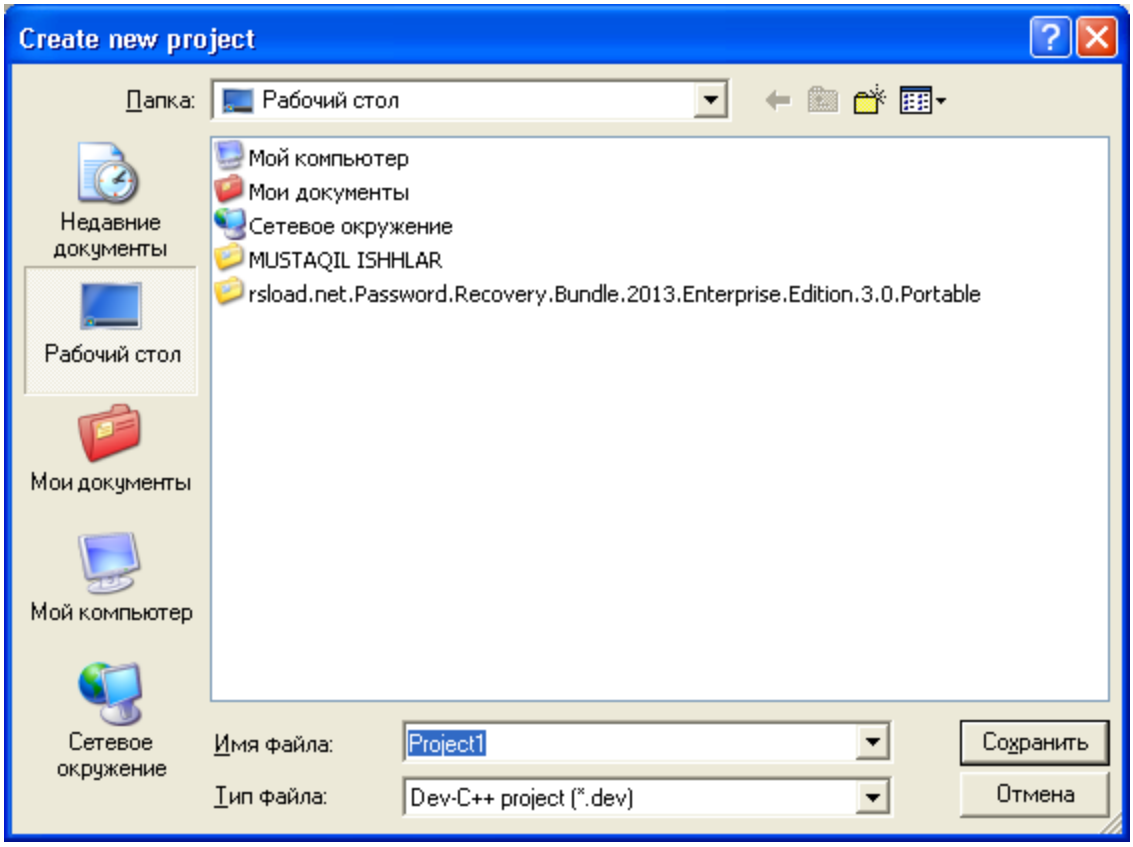
Bizga kerak oyna



Biz < Concole Application > ni tanlaymiz

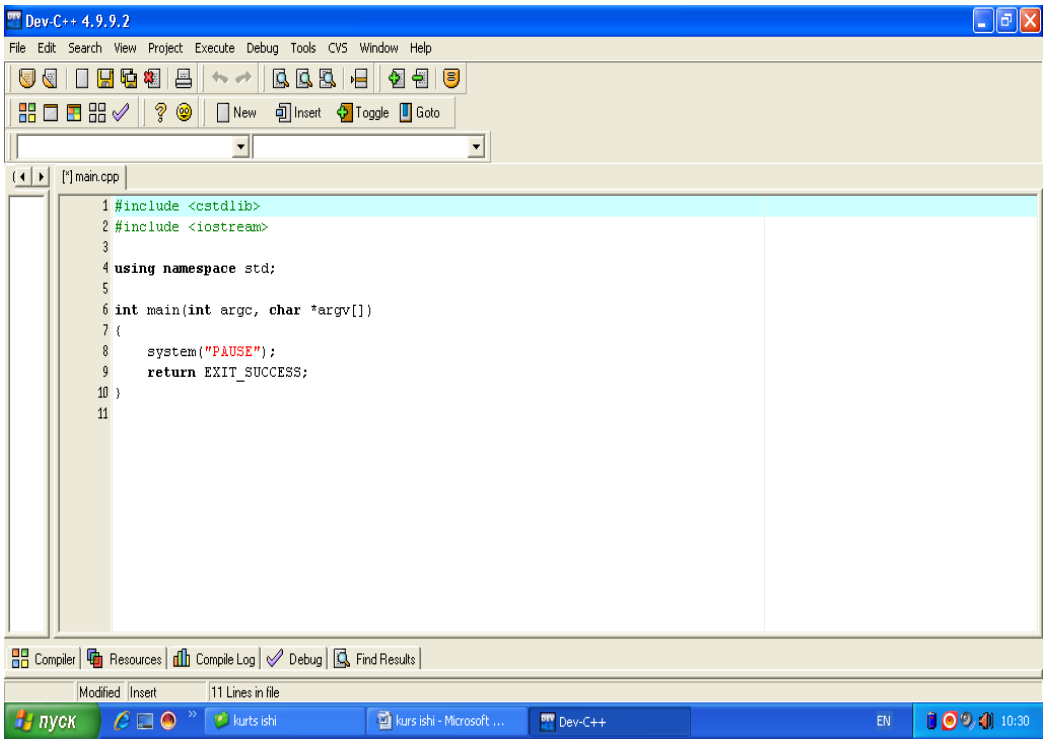


< Ok > bosganimizdan so'ng



Oynasi chiqadi. Ushbu oyna bizga tuzmoqchi bo'lgan dasturimiz qismlarini qayerga saqlashni va qanday nom bilan nomash imkoniyatlarini beradi.

< Сахранит > bossak



oynasi chiqadi.

Ushbu oynaning eng yuqori qismida dastur kutubxonalarini chaqirib olinadi

Biz tuzmoqchi bo'lgan dastur juda sodda bo'lib bizga uni tuzishda C++ matematik kutubxonasidan foydalanamiz.

Ko'rib turganingizdek biz C++ dasturlash tini ishga tushirib oldik.

{ belgisidan keyin dasturni kiritamiz.

Dasturning vazifasi

Biz tuzadigan dasturning vazifasi quyidagicha.

C++ dasturlash tilida Uchburchak uchlarining to'g'ri burchak koordinatalari X1. Y1, X2. Y2, X3, Y3 va nuqtaning X ,Y koordinatalari berilgan. Nuqta uchburchakda yotish – yotmasligini aniqlaydigan dastur tuzish. Xisoblash xatoliklari e'tiborga olmasin.

Dasturning umumiy ko’rinishi

```
 //Muallif: Goyibnazarov
//Maqsad: Nuqta uchburchakka tegishli yoki tegishli emasligini aniqlash
//Sana: 25.01.2014

//================================================
===Kutubxonalar e'loni
#include <iostream>
#include <cmath>
#include <cstdlib>

using namespace std;
//ikki nuqta orasidagi masofa
float uzunlik(float x1, float y1, float x2, float y2){
	return sqrt(pow((x2 - x1),2) + pow((y2 - y1),2));
};
//yuza hisoblash uchun Geron formulasi
float yuza(float x1, float y1, float x2, float y2, float x3, float y3){
	float a, b, c;
	float p;
	//ikki nuqta orasidagi masofa
	a = sqrt(pow((x2 - x1),2) + pow((y2 - y1),2));
	b = sqrt(pow((x2 - x3),2) + pow((y2 - y3),2));
	c = sqrt(pow((x3 - x1),2) + pow((y3 - y1),2));
```

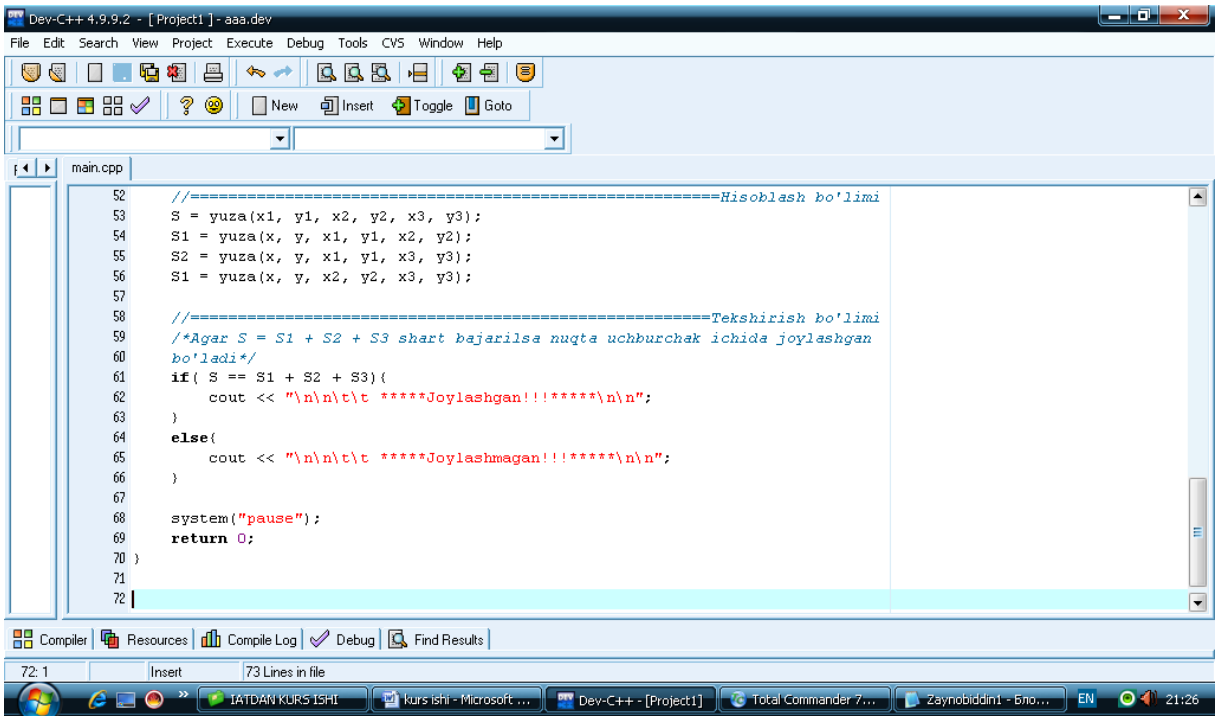
```
    p = (a + b + c) / 2;// yarim pirimetr
    return sqrt(p * (p - a) * (p - b) * (p - c));
};

int main(){
    float x1, x2, x3, y1, y2, y3;//uchburchak uchlari
    float x, y;// qayerda joylashgani aniqlanadigan nuqta
    float S;// uchburchak yuzasi
    float S1, S2, S3; /*Agar nuqta uchburchak ichida bo'lsa nuqta va uchburchak
    uchlarini tutashtirishimizdan uchta uchburchak hosil bo'ladi*/

    Nishon:
    //==================================Ekran orqali kiritiladigan o'zgaruvchilar
    cout << "Uchburchak uchi koordinatalarini kiriting:\n";
    cout << "x1 = "; cin >> x1; cout << "y1 = "; cin >> y1;
    cout << "x2 = "; cin >> x2; cout << "y2 = "; cin >> y2;
    cout << "x3 = "; cin >> x3; cout << "y3 = "; cin >> y3;


    //=============================================================Uchburchak mavjudmi
```

```
if( ((x1 == x2) && (x2== x3)) || ((y1 == y2) && (y2== y3))){
        cout << "\n\n\t\t Bunday uchburchak mavjud emas\n\n";
        goto Nishon;
}
//==================================Mavjud uchburchak uchun nuqta kiritiladi
cout << "Tekshiriladigan nuqta koordinatalarini kiriting:\n";
cout << "x = "; cin >> x; cout << "y = "; cin >> y;

//==========================================================Hisoblash bo'limi
S = yuza(x1, y1, x2, y2, x3, y3);
S1 = yuza(x, y, x1, y1, x2, y2);
S2 = yuza(x, y, x1, y1, x3, y3);
S1 = yuza(x, y, x2, y2, x3, y3);

//==========================================================Tekshirish bo'limi
/*Agar S = S1 + S2 + S3 shart bajarilsa nuqta uchburchak ichida joylashgan
bo'ladi*/
if( S == S1 + S2 + S3){
        cout << "\n\n\t\t *****Joylashgan!!!*****\n\n";
```

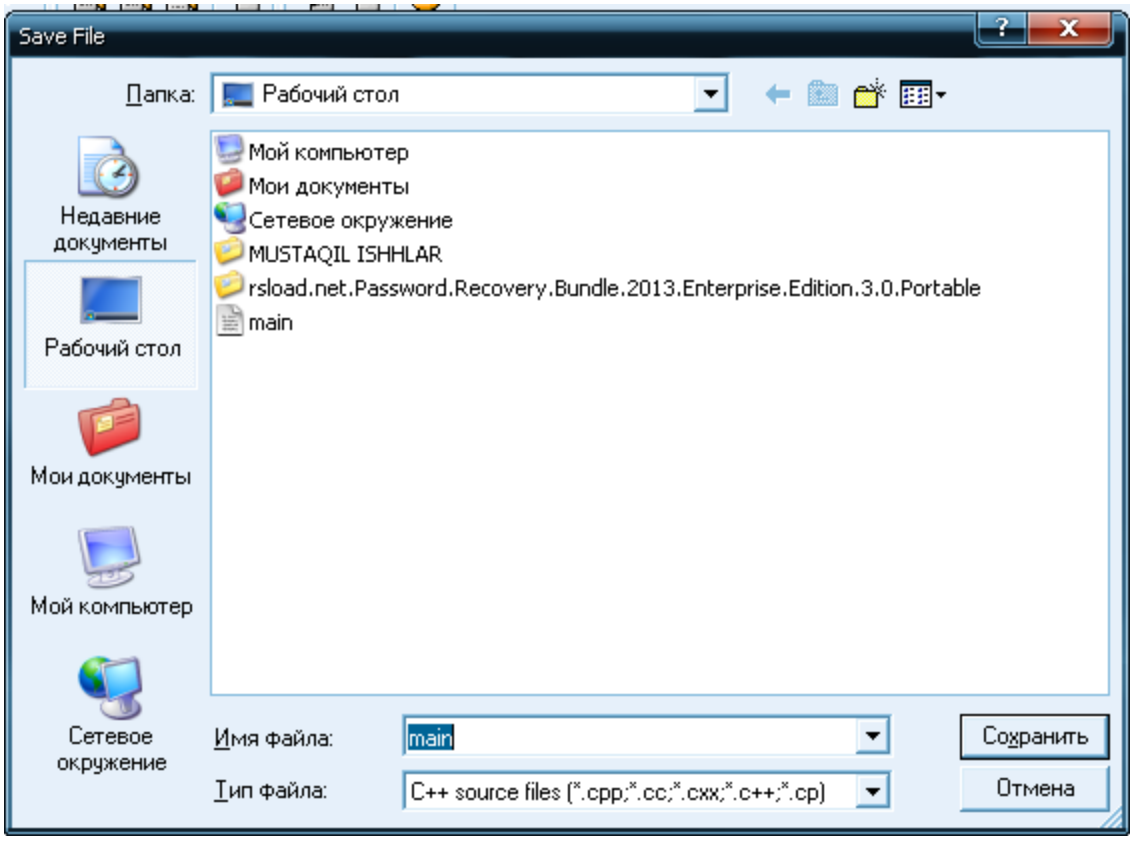
```
    }
    else{
        cout << "\n\n\t\t *****Joylashmagan!!!*****\n\n";
    }

    system("pause");
    return 0;
}
```

Dasturni C++ dasturlash tilida yaratish uchun yuqorida aytib o'tilganidek ishlarni bajaramiz va dasturimizni tana qismini { belgisidan keyin kiritamiz.

Dasturni kiritgandan keyin

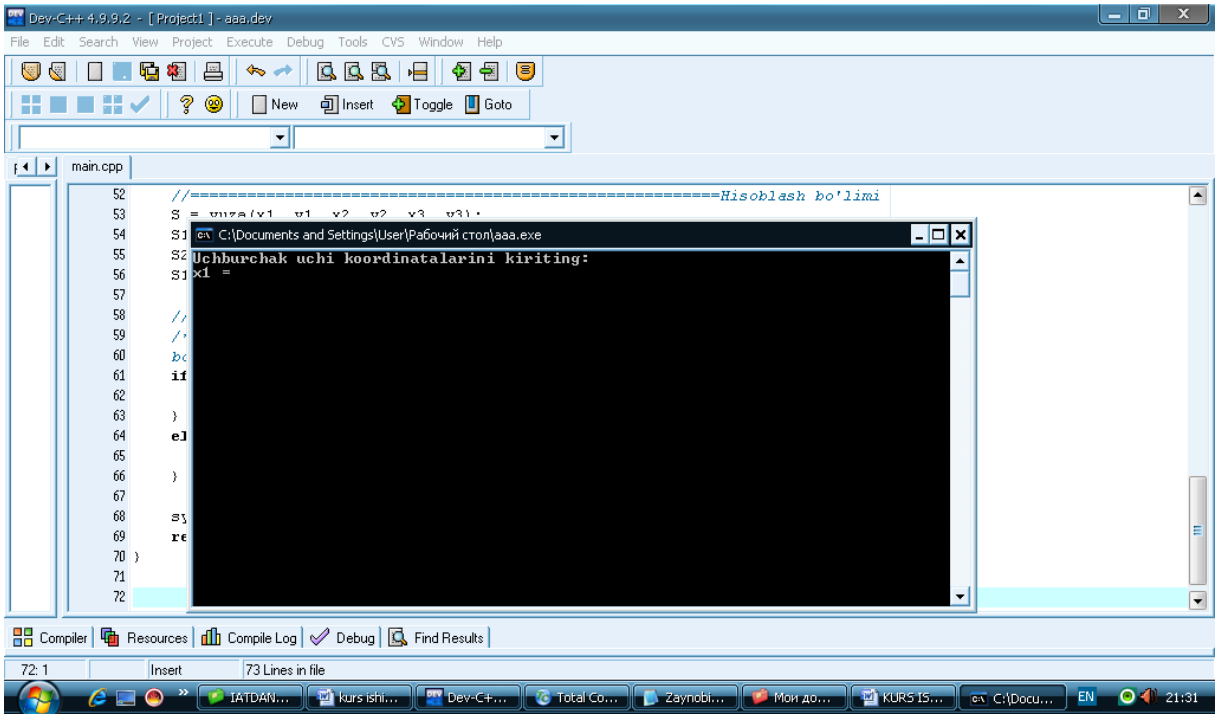


F9 tugmasini bosamiz va

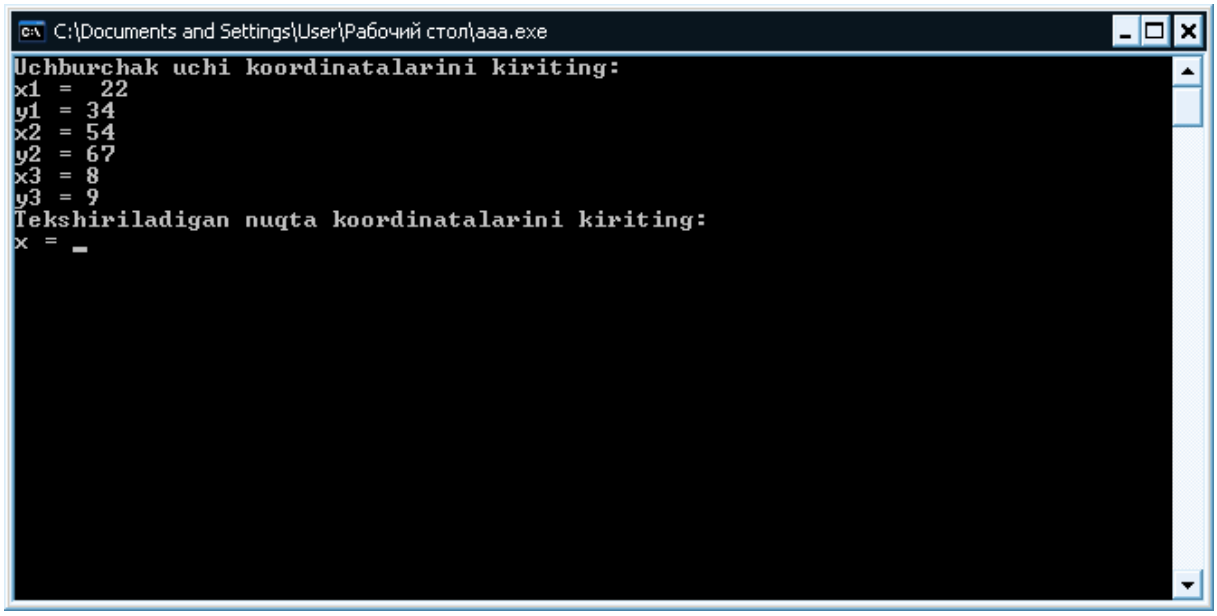


Oyna chiqadi dasturning asosiy qismini qayerga saqlashni va unga qanday nom berishni tanlab <<Enter>> tugmasisi bosiladi.

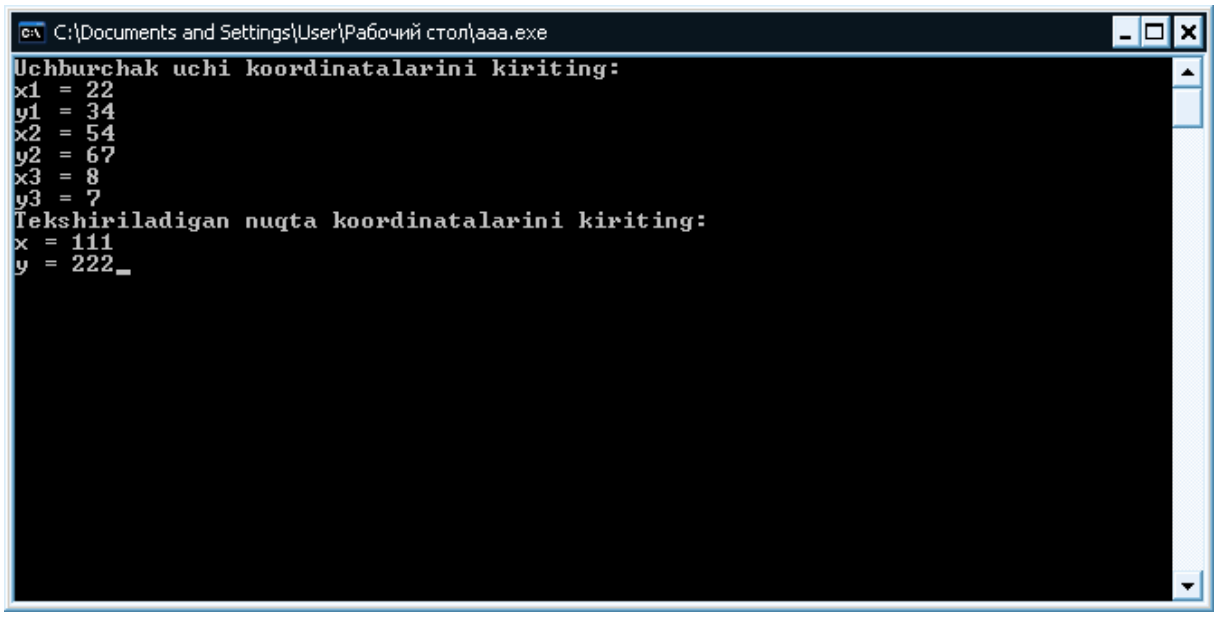
Agar dasturda xatoliklar yo'q bo'lsa ushbu oyna chiqadi.



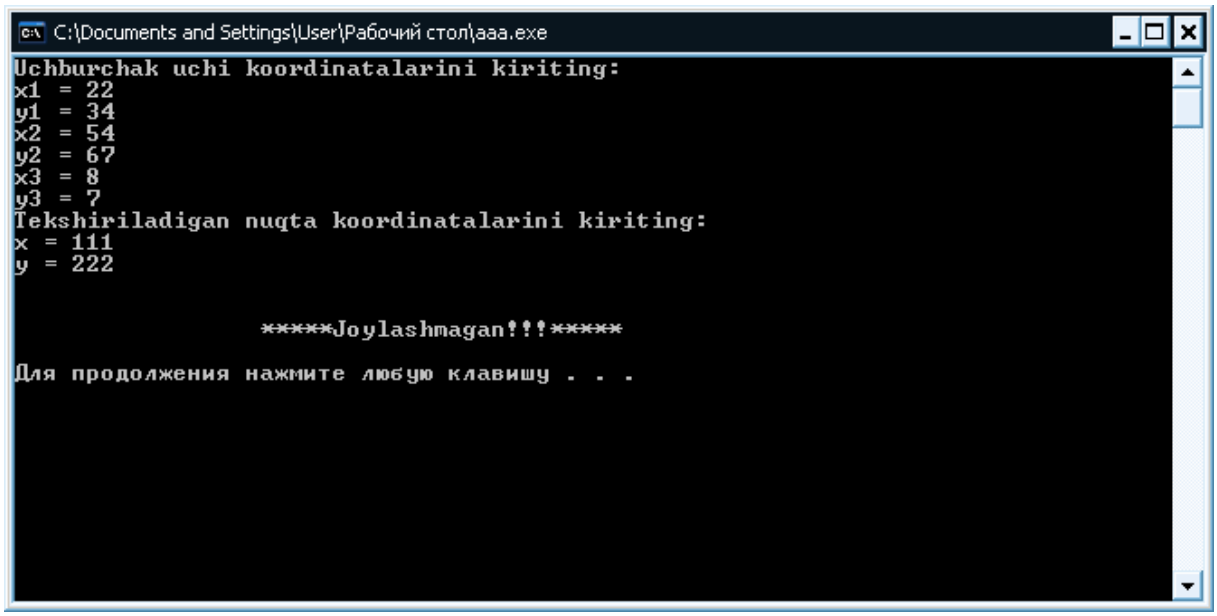
So’ng ushbu oynaga so’ralgan koordinatalarni kiritamiz.



Endi tekshirilayotgan nuqta koordinatalarini kiritamiz.



So’ng <<Enter>> tugmasi bosiladi.



Ko’rinib turgandek ushbu nuqta koordinatalari uchburchakka tegishli emas.

Xulosa

Men ushbu Kurs ishini bajarish mobaynida eng avvalo dasturlash, haqida u nima o'zi dasturlashning qanday talablari va imkoniyatlari borligi haqida, songra dasturlash tillari va ularning bir-biridan farqlari va o'xshashlik tomonlari, kamchilik va afzalliklari haqida kerakli ma'lumotlarga ega bo'ldim.

Qolaversa men ushbu dasturni C++ dasturlash tilida tuzish davomida C++ dasturlash tili haqida o'z bilimlarimni yanada mustahkamlab oldim va qo'shimcha ma'lumotlarga ega bo'ldim.

Men tuzgan dasturda C++ dasturlash tilining massiv operatoridan foydalanganim tufayli men massivlar haqida ham ma'lumotlarga ega bo'ldim.

Foydalanilgan adabiyotlar

C++ dasturlash tili haqida ma’ruza 2011.

C++ dastur uz.

Dasturlash tillari haqida ma’ruza 2012.

Dasturlash haqida ma’ruza 2013.