

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

**TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI
URGANCH FILIALI**

KOMPYUTER INJINIRINGI FAKULTETI

DASTURIY INJINIRINGI KAFEDRASI

C++ DA DASTURLASH FANIDAN

Kurs ishi

**MAVZU: AVTOMASHINALAR MA'LUMOTLAR BAZASINI YARATISH VA
BOSHQARISH**

Bajardi:

911-14 guruh talabasi
Jabbarov Jamshid

Ilmiy rahbar:

Dasturiy injiniring kafedrası
assistenti
Kuryazov D.

MUNDARIJA.

KIRISH.

I. NAZARIY QISM......

1.1. “Avtomashinalar ma`umotlar bazasini yaratish va boshqarish” dasturi haqida umumiy ma`lumot.....

1.2. Visual Studio 2012 dasturda C++ dasturlash muhitining imkoniyatlari.....

II. ASOSIY QISM

2.1. Loyihaning qoyilishi.....

2.2. Dasturning funksional xarakteristikasi.....

2.3. “Avtomashinalar ma`umotlar bazasini yaratish va boshqarish ” dasturini Visual Studio 2012 dasturida C++ tilidagi yaratish.....

2.4. Foydalanuvchiga yo`riqnoma.....

XULOSA......

FOYDALANILGAN ADABIYOTLAR.

ILOVA......

KIRISH

Hozirgi kunda axborot va kompyuter texnologiyalari iboralari kundalik turmushda eng ko'p qo'llaniladigan tushunchalar desak mubolag'a bo'lmaydi. Chunki hayotning qaysi sohasini olmaylik, qanday amallarni bajarmaylik, albatta, axborotlar bilan ish ko'ramiz. Ya'ni axborotlardan foydalanish, axborot almashish, ularni uzatish, o'zlashtirish inson faoliyatining asosiy negizini tashkil etadi. Axborot texnologiyalari jahonda jadal sur'atlar bilan rivojlanib, takomillashib borayotgan sohadir. Insoniyat axborotlashtirish sohasida haqiqiy inqilobiy o'zgarishlar davrini boshidan kechirmoqda, buning natijasida esa umumjahon axborotlashgan hamjamiyati shakllanmoqda. Bu o'zgarishlar asosini axborotlashtirish, telekommunikatsiya va komp'yuter texnologiyalari konvergentsiyasi negizida axborotni uzatish hamda iste'molchiga etkazib berishning eng zamonaviy va samarali vositalarining dunyoga kelayotganligi tashkil etadi.

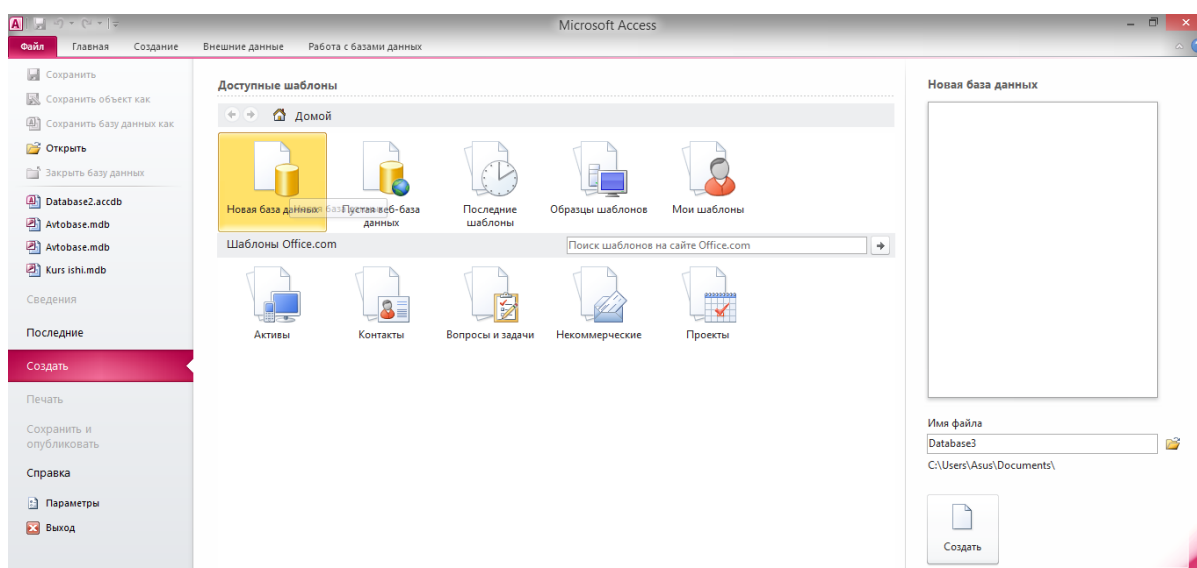
Bugungi kunda kompyuter inson faoliyatining turli sohalariga kirib bormoqda. Natijada dasturiy ta'minotga bo'lgan talab tobora kengayib bormoqda. Bu o'z navbatida dasturchilarga bo'lgan talabni oshiradi. Hozirgi kunda kompyuter va axborot texnologiyalari taraqqiyotida katta yutuqlarga erishilmoqda. Mamlakatimizda xalq xo'jaligi tarmoqlarida: ishlab chiqarish jarayonlari, ofis ishlari, bank operasialari, axborot almashinuv jarayonlari, milliy xizmat ko'rsatish muassasalari, ilmiy tadqiqot ishlari, texnikalarni avtomatik va masofadan boshqarish masalalari, ommaviy axborot vositalari, tibbiyot maskanlaridagi jarayonlarda kompyuterlar va axborot texnologiyalari muvaffaqiyatli qo'llanilmoqda. Hozirgi vaqtda bir qancha dasturlash tillari mavjud bo'lib, ular har xil sohaga doir masalalarni hal qilishda ishlatiladi.

I. NAZARIY QISM

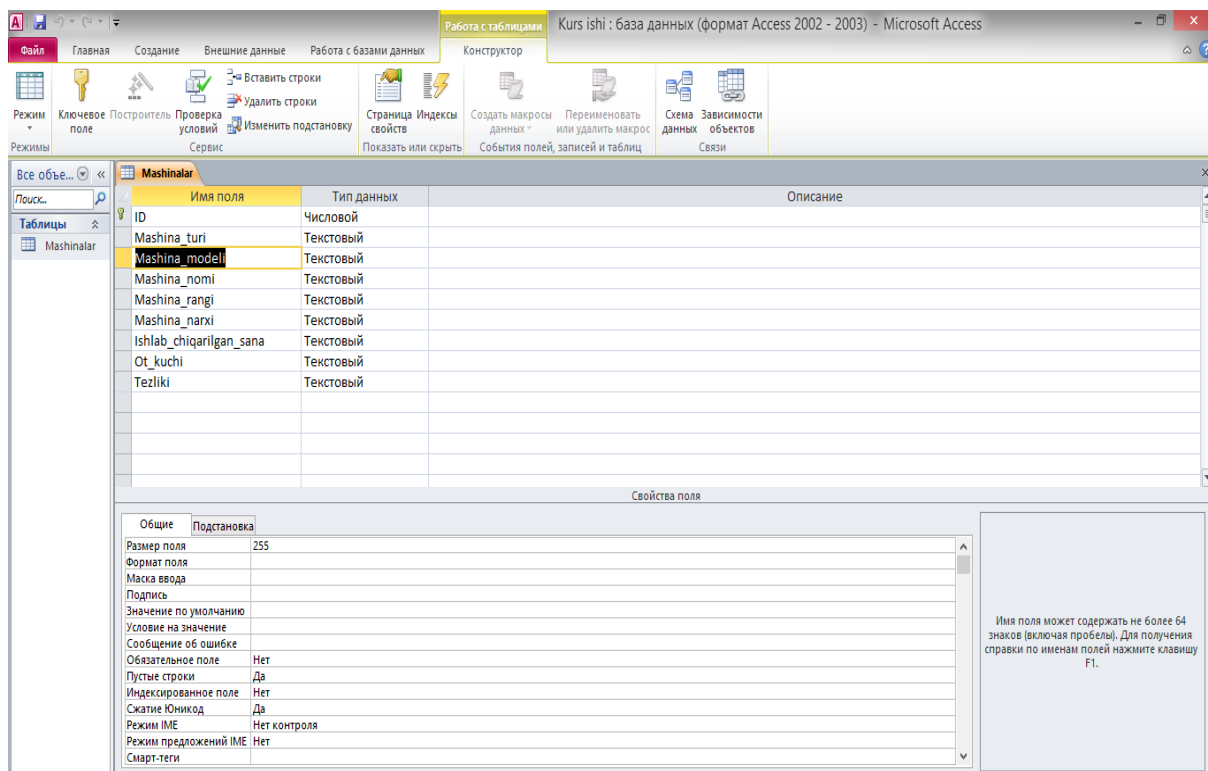
1.1. “Avtomobil ma`lumotlar bazasi” dasturi haqida umumiy ma`lumot.

Bu masalani yechishda Microsoft Access ma`lumotlar bazasini tanladim va unda berilgan mazkur ma`lumotlar ustida bajariladigan amallar algoritmini tuzdim. Visual Studio C++ texnologiyasi imkoniyatlaridan foydalangan holda dastur ko`rinishini yaxshilashga, foydalanish uchun qulayligini taminlashga erishdim. Dastur “Visual Studio C++”i orqali ishga tushadi. Biz baza bilan ishlash uchun Microsoft Accessga Visual Studio serveri orqali ulanib olamiz.

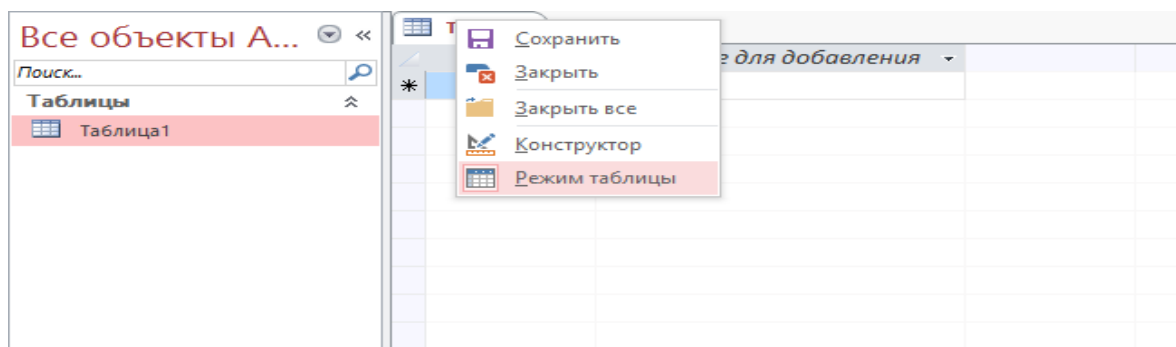
Ulangandan keyin bizda quyidagicha oyna bo`ladi:



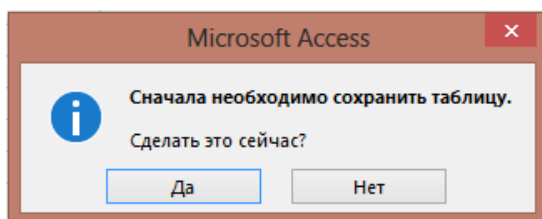
1.1-rasm. Bu yerdan Accesni nomini saqlagandan keyin quyidagi ish stoli paydo bo`ladi. Bu yerdan biz “создание” menyusi orqali “Конструктор таблиц” bo`limiga kiriladi.



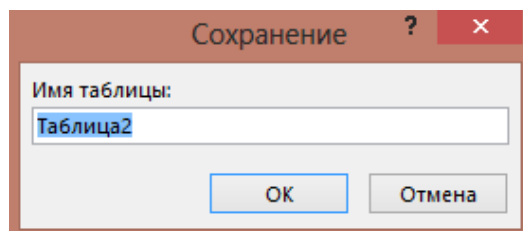
1.2.-rasm.



1.3.rasm. Paydo bo'lgan oynada biz yaratmoqchi bo'lgan jadvalning ustunlari va ustunlarning tipi yoziladi. Masalan:

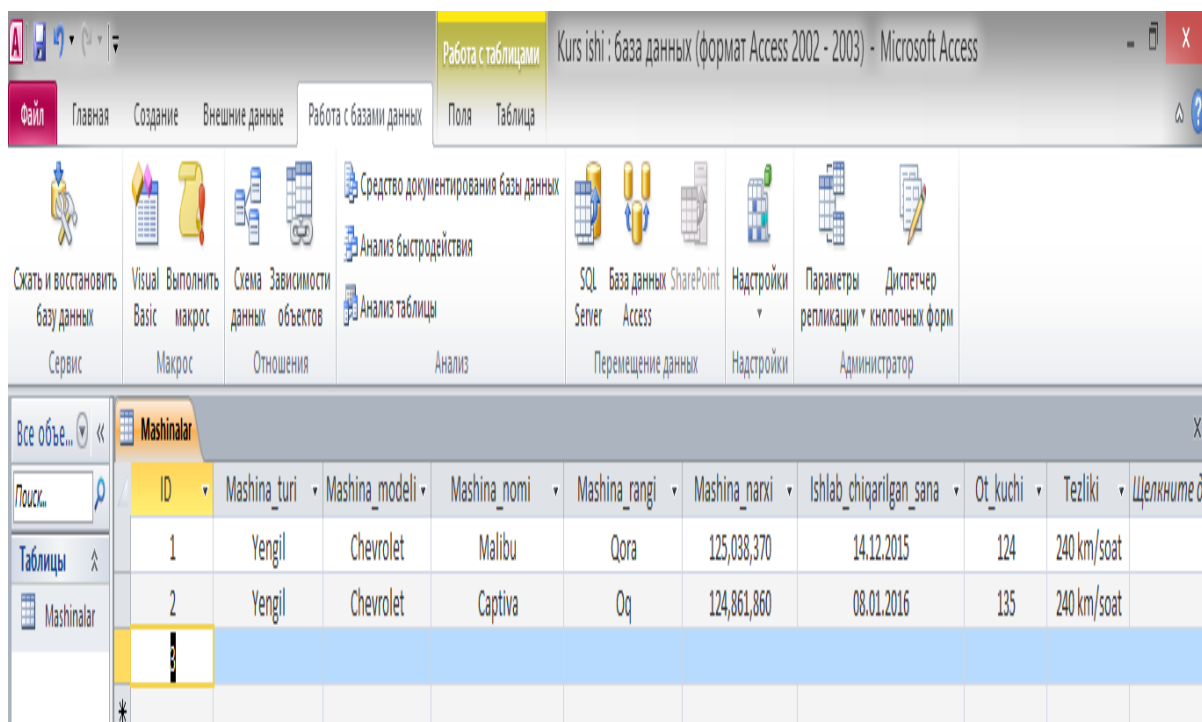


1.4-rasm. Endi biz "Таблица 2" ning ustida sichqonchanning o'ng tugmasi bosiladi va u yerdan "Режим таблиц" tanlanadi va ekranda quyidagi oyna paydo bo'ladi.

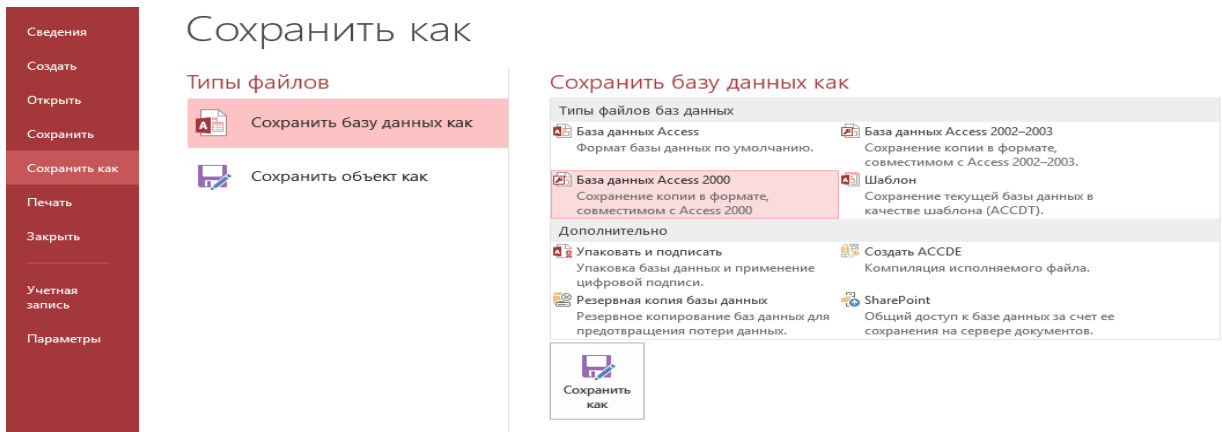


1.5-rasm. Bu yerdan “Да”ni tanlaganimizdan keyin yana kichkina oynacha paydo bo’ladi.

Bu bo’linga kirmoqchi bo’lganimizda kichik oyna hosil bo’ladi va undan “OK” tugmasi bosiladi. So’ngra yangi ochiladigan sahifamizni nomlab olamiz. Bu yerda biz nomlagan ustunlarimizning ichini ma’lumotlar bilan to’ldiramiz.



1.6.-rasm. Ma’lumotlar to’ldirib bo’lingandan keyin uni “файл” menyusidan “сохранить как ” belgisi tanlanadi va u yerdan “База данных Access 2000”



1.7.-rasmda ko`rsatilgandek tanlanadi va “сохранить как” tanladi va Accessning nomini yozib saqlab qo`yiladi. Saqlangandan so`ng access mdb faylga aylanadi.

1.2. Visual Studio 2012 dasturda C++ dasturlash muhitining imkoniyatlari

Dasturlash fanining rivojlanishi bevosita kompyuterlarning rivojlanishi bilan bog'liqdir. O'tgan asrning o'rtalarida kompyuterlarning yaratilishi, ular bilan parallel ravishda uning dasturiy ta'minotini ham rivojlanishiga sabab bo'ldi. Kompyuterlarning texnik ta'minotini rivojlanishi bilan, ular uchun turli dasturiy mahsulotlar yaratila boshlangan va yaratilmoqda, hamda kompyuterlarning texnik ta'minoti bilan parallel darajada rivojlanib bormoqda. Dasturiy ta'minotsiz kompyuterlar oddiy qutidan farqi qolmas edi.

Hozirgi vaqtda bir qancha dasturlash tillari mavjud bo'lib, ular har xil sohaga doir masalalarni hal qilishda ishlatiladi. C++ dasturlash tili visual dasturlash texnologiyasini qo'llash asosida turli darajadagi qiyinchilikdagi masalalarni tez hal qilish imkonini beradi.

Kurs ishining birinchi qismida Visual Studio 2012 dasturi imkoniyatlari haqida so'z boradi. Ikkinchi qismida C++ tili operatorlari tavsifi va ularning qo'llanilishi to'g'risida ma'lumot keltirilgan. Ushbu kurs ishida kalkulyator dasturini yaratish uchun kerakli operatorlar va Visual Studioda qo'llanilgan har bir komponentalar xususiyatlari haqida ma'lumot berilgan.

Visual Studio, **Microsoft** firmasining mahsuloti hisoblanadi. Bir necha yillardan beri dasturchilarning asosiy ish quroli bo'lib kelmoqda. Bu dastur orqali nafaqat C++ dasturlash tilida, balki boshqa tillar(C#, F#, FoxPro)da ham dastur tuzish mumkin bo'ladi. Shu sababli bu dastur biroz og'irroq hisoblanadi(katta hajmdagi tezkor xotirani talab qiladi). Visual Studio dasturini bir necha hil versiyalari mavjud bo'lib, hozirgi kunda **Visual Studio 2013** versiyasi chiqmoqda. Har bir versiyasiga bir necha modullar qo'shilib, dasturni biroz qiyinlashtirmoqda. Shularni inobatga olib, kompyuteringizni tuzilishiga qarab(tezkor xotira hajmi, protsessor,...), kerakli versiyani o'rnatishni maslahat beraman. Zamondan orqada qolmay deb, oxirgi versiyasini qo'yib, kompyuteringizni qiynab qo'ymang.

Visual Studio bu dastur yozish uchun **muhit**(joy) hisoblanadi. Bu muhitga, siz tuzgan dastur kodlarini kompyuter kodlarga o'tkazib beradigan **kompilyator**

birlashtirilgan bo‘ladi. Har bir dasturlash turi uchun alohida kompilyator mavjud bo‘lib, ishlatilgan dasturlash tiliga qarab kompilyator ishga tushadi. Visual Studio dasturini o‘rnatib, barcha kompilyatorlar ham avtomat o‘rnatiladi.

II. ASOSIY QISM

2.1. Loyihaning qoyilishi

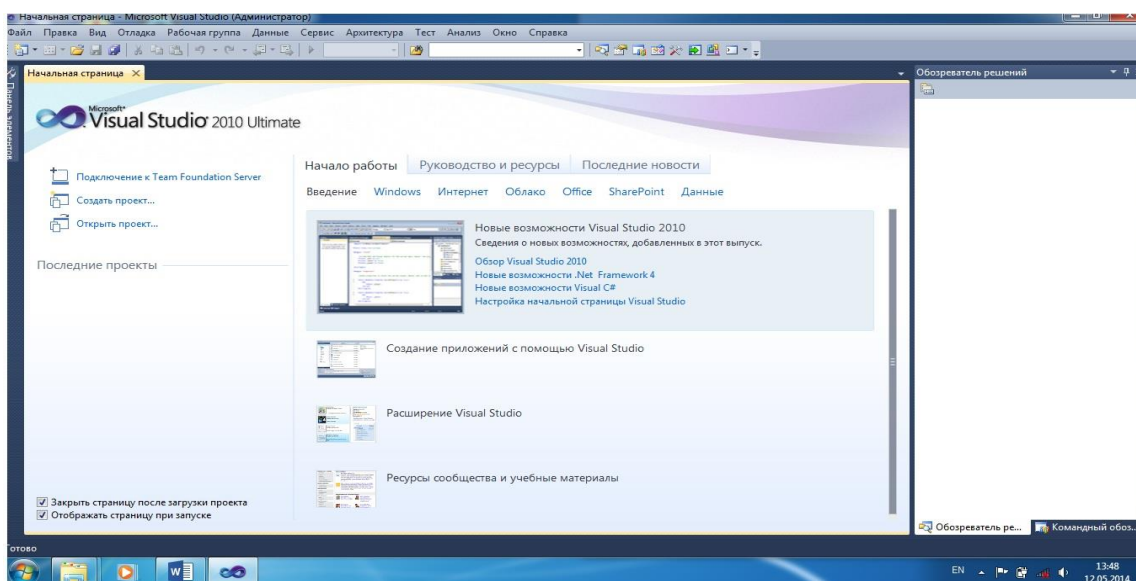
Hozirda kunda respublikamizda jahon mashinalari bilan bemalol raqobatlasha oldigan bir necha turdagi mashinalar ishlab chiqilmoqda. Shu sababli respublikamizni barcha viloyatlarida ko‘plab avtosalonlar ish yuritmoqda. Bu salonlarda O‘zbekistonda ishlab chiqarilayotgan o‘nlab turdagi mashinalardan yuzlab miqdorda keltirilmoqda.

Bizning vazifamiz shu avtosalonlar uchun malumotlar bazasini yaratishdan iborat. Bu bazamizda bo‘lishi kerak shu salondagi barcha mashinalar haqidagi malumot. Biz shu orqali barcha malumotlarni bilib olishimiz va yangi mashina kelsa bazaga u haqidagi malumotlarni kiritish kerak. Avtomobil saloni GM O‘zbekiston bilan shartnoma asosida ishlaydi. Mijozlarga o‘z mashinalariga ega bo‘lish imkoniyatini yaratib beradi. Unda quyidagi xizmatlar ko‘rsatiladi. Birinchi bo‘lib kirib kelgan mijozga mashinalar bilan tanishtiriladi. Ularning rusumlari, qulayliklari, narxi, imkoniyatlari va boshqalari ko‘rsatiladi. Mijoz mashinani tanlaganidan keyin u bilan shartnoma tuziladi. Shartnomada mashinaning narxi, rusumi, qanday rangda ekanligi va qachon avtosalonga yetib kelishi ko‘rsatiladi. Tuzilgan shartnoma GM O‘zbekistonga yuboriladi va shu asosida mashina ishlab chiqariladi va shu mashina shartnomadagi raqam asosida avtosalonga yuboriladi va mijoz o‘z mashinasini sozligini tekshirib olib ketishi mumkin.

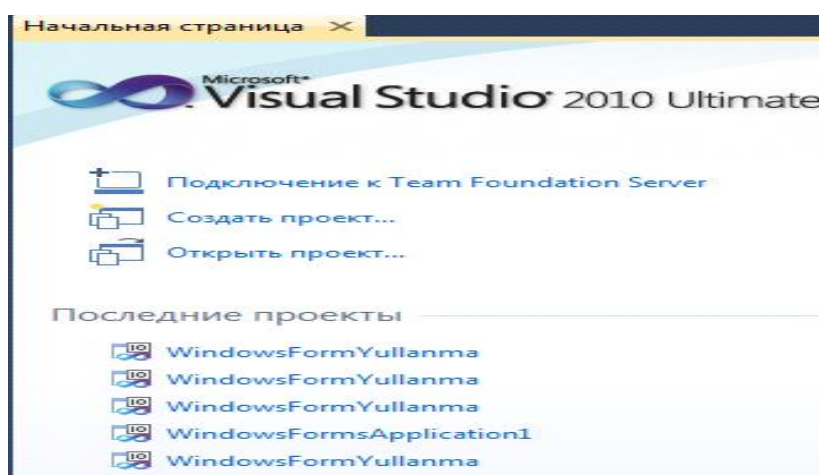
Microsoft Visual Studio haqida

Microsoft korporatsiyasi o‘zining oynalar bilan ishlovchi va dunyoda eng ko‘p tarqalgan va juda ham ko‘p foydalanuvchilar tomonidan ma‘qullangan operatsion tizimi Windows uchun ishlab chiqqan Visual Studio (VS) muhiti ham faqatgina oynalar orqaligina ishlaydi. Unda mavjud barcha dasturlash tillari oynalarga bo‘lingan va shu oynalar yordamida bir biriga bog‘langan. VS yuklash

uchun **пущ->program->Microsoft Visual Studio->Microsoft Visual Studio** ketma ketligini bajarish kerak.VS yuklanganda odatda quyidagi ko'rinishda bo'ladi. Birinchi navbatda,VS yuklanganda Start Page muloqot oynasi ochiladi.



2.1-rasm. MS Visual Studio muloqot oynasi.



2.2-rasm.Bosh sahifa oynasi

1.Recent Projects – avval ishlangan loyihalardan 6 tasi ekranda ko'rsatilgan bo'ladi, ularni ustiga sichqonchanning chap tugmasini ikki marotaba bosish orqali ishga tushirish mumkin, shu bo'limning pastki qismida Open: Projects va Create: Projects bandlari bo'lib, open bandi orqali xotirada mavjud bo'lgan loyihani qayta ochish imkonini beradi, create bandi orqali esa yangi loyiha yaratish mumkin.

2. Getting started – loyiha yaratish va shu muhit haqida ma'lumot olish imkonini beradi;

3. Visual Studio Headlines – bu bo'limda sizning shu muhitda ishlayotganingiz uchun uning mualliflari tomonidan bildirilgan minnatdorligi ko'rsatilgan;

4. Visual Studio Developer News – bu asosiy qismda VS ustida qanday o'zgarishlar olib borilayotgani yoki olib borilgani haqida ma'lumot beradi. Endi biz yangi foydalanuvchi bo'lganimiz uchun bu muloqot oynasini yopib, yangi oyna ochamiz. Buning uchun biz menyular paneli bilan tanishish kerak.

Menyular paneli bilan ishlash.

1. File

- **new** – yangi loyiha yaratish
- **open** – yaratilgan loyihani ochish
- **close** – joriy loyihani yopish
- **close Solution** – barcha loyihalarni yopish
- **save Selected Items** – belgilangan barcha loyiha qismlarini saqlash
- **save Selected Items As** - belgilangan qismlarni barchasini xohlagan tartibda saqlash
- **save All** – barcha dasturlarni saqlash
- **export Template** – dasturni arxivlash
- **page Setup** – sahifa sozlamasini o'zgartirish
- **print** – chop etish
- **recent files** – fayllarni qayta ochish
- **recent Projects** – loyihalarni qayta ochish
- **Visual Studio** dan chiqish

2. Edit

- **undo** – bajarilgan amallarni bir marotaba bekor qilish;
- **redo** – oxirgi bekor qilingan amalni qaytarish;
- **undo Last Global Action** – barcha qilingan amallarni bekor qilish;
- **redo Last Global Action** – bekor qilingan barcha amallarni qaytarish;
- **cut** – belgilangan qismni yoki komponentni qirqib, xotirada saqlab turish;
- **copy** – belgilangan qism yoki komponentning nusxasini xotirada saqlab turish;
- **paste** – xotiraga qirqib yoki nusxalab olingan qism yoki komponentni

belgilangan yoki ko'rsatilgan joyga qo'yish

- **select All** – barcha qismlarni belgilab olish;
- **Find and Replace** – ma'lum qism yoki komponentani butun muhit bo'ylab qidiradi va o'zgartiradi

3.View

- **open** – Joriy dasturni ochib beradi;
- **open With** –Joriy dasturni xohlagan dastur yordamida ishlov berish imkoniyatini yaratadi;
- **server Explorer** – Server oynasi bo'lib, unda asosiy komponentalar joylashadi;
- **solution Explorer** – Ob'yektlar oynasi;
- **class View** – Loyiha elementlarini ko'rsatib beradi, uning yordamida yangi yoki qo'shimcha ob'yektlarni qo'shish, olib tashlash, ko'rinishini, tartibini o'zgartirish mumkin;
- **code Definition Window** – Ob'yekt kodlarini ko'rsatib beruvchi oyna;
- **object Browser** – Visual Studiodagi barcha ob'yektlar xususiyatlarini va ularni qo'llash usullari beriladigan oyna;
- **Error List** – xatoliklar oynasi bo'lib, dastur kompilyatsiya qilinishidan oldin va keyin undagi xatoliklarni ko'rsatib beradi va unda namoyon bo'lgan har bir xatolikni ustida ikki marotaba sichqoncha chiqillatilsa xatolik bo'lgan joyga kursorni eltib qo'yadi;
- output** –Dastur o'z ishlashi davomida qanday fayllarga murojaat qilishi, qanday fayllarni hosil qilishi, qaysi cs yordamida hosil bo'lishi va qanday qiymat qaytarishini ko'rsatuvchi oyna;
- **properties Window** – xususiyatlar oynasi bo'lib, qaysi ob'yekt faol bo'lsa, shu ob'yektning xususiyatlarini;
- **toolbox** – komponentalar joylashgan oyna bo'lib, unda VS da ishlatish mumkin bo'lgan barcha ob'yektlar joylashtirilgan;
- **Find Result** – natijani topish oynasi bo'lib, uning yordamida dastur ishlatilganda berilgan qiymatlarda qanday natija olinishini ko'rsatib beradi;
- **toolbars** – panellar qo'yish yoki olib tashlash bandi bo'lib, unda barcha turdagi

panellarni izlash mumkin;

- **full Screen** – VS butun ekran hajmida kattalashadi;

4. Refactor

- **rename** – dasturni qayta nomlash;
- **extract Method** – metodni qayd etish;
- **extract Interface** – Interfeysni qayd etish;
- **remove Parameters**–parametrlarni qayta ko'chirish;
- **reorder Parameters** – parametrlarni qaytarish.

5. Project

- **add Windows Form** –Windows formasini qo'shadi;
- **add User Control** – foydalanuvchi boshqarishining boshqa usullarini qo'shadi;
- **add Component** – Yangi component qo'shadi (foydalanuvchi tomonidan tayyorlangan bo'lishi ham mumkin);
- **add Class** – yangi sinf(ma'lumotlar yoki formalar sinfi)ni qo'shadi;
- **add New Item** – yangi elementini qo'shadi (ob'yektning);
- **add Exiting Item** – yangi chiquvchi elementini qo'shadi;
- **show All Files** – barcha fayllarni namoyish etadi;
- **console - Properties** – Consolning xususiyatlari

6. Debug

- **windows** – Breakpoints,Output, Immediate Windows bandlarni o'z ichiga olgan bo'lib, bu oynalar yordamida ishlash imkonini beradi;
- **start Debugging** – Joriy loyihani komplyatsiya qilib, ishga tushiradi;
- start Without Debugging** – Dasturni komplyatsiya qilib, bir necha komponentalarni birgalikda ishga tushiradi;
- **new Breakpoints** - Yangi to'xtash nuqtalarini hosil qiladi.Bunda hosil bo'lgan nuqtalarni Shift+F5 tugmasi orqali bekor qilish mumkin;
- Delete All Breakpoints** – Barcha qo'yilgan to'xtash nuqtalarini bekor qilish uchun ishlatiladi.

7. Data

- Show Data Sources** – Shu loyiha bilan bog'langan barcha ma'lumotlar

bazalarini ko'rsatib beradi;

-Add New Data Sources – Yangi ma'lumotlar bazasini qo'shish uchun ishlatiladi;

-Schema Compare – Shu loyihada mavjud bo'lgan ma'lumotlar bazalari orasidagi barcha bog'liqliklarni ko'rsatib beradi;

-Preview Data – Shu loyihadagi barcha ma'lumotlar bazalarini qidirish uchun ishlatiladi;

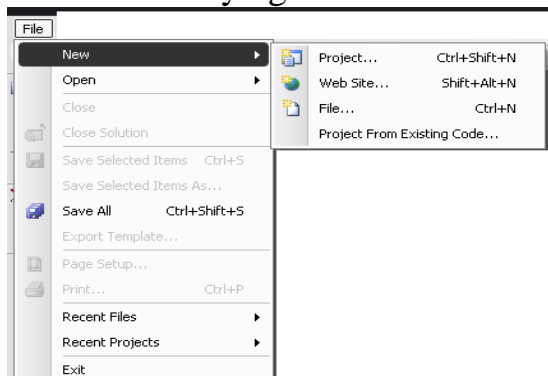
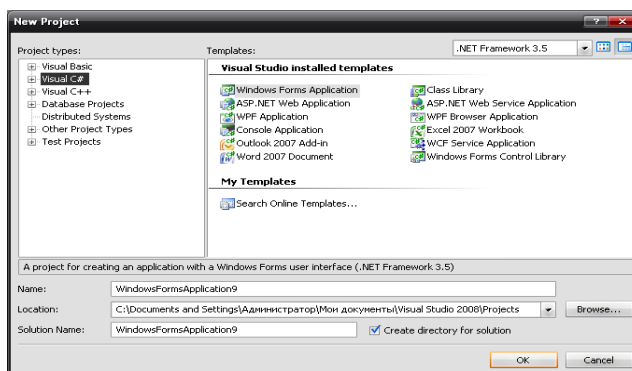
-Refactor – Refaktor bandi bilan bog'liqlik o'rnatadi.

8. Format

-Align – Formaning joylashish koordinatlarini o'rnatish imkonini beradi. Bunda uning chap, o'ng, o'rta, past va yuqoridan chegaralash mumkin;

-Make Same Size – Formaning o'lchamini berish mumkin. Unda balandligi, eni qalinligi va chegara qismini berish mumkin;

New- Yangi loyiha yaratish VS ning eng asosiy amallaridan biri. Bu menyuda yangi loyiha, web-sayt, file (o'zida kodlarni saqlab turuvchi *.cs fayl), oddiy fayl yaratish mumkin. VS da bir necha tillar jamlangan. Biz shu tillar orasidan hozirda dunyoning eng kuchli dasturlash tillaridan biri deb tan olinayotgan C# dasturlash tilidir.



2.3-2.4-rasm.

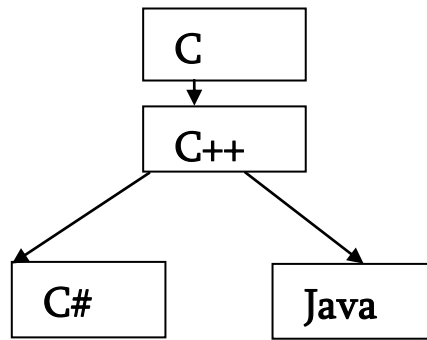
C++ tili o'zining interfeysi va sintaksisi uning yuqori sathli dasturlash tillariga kirishiga olib keladi. VS dagi C++ tili ham huddi shu vazifani bajarib beradi. Uning Console Application qismida qora

oynali muloqot oynasi yaratiladi va shu qora oyna dastur ishga tushganda paydo bo'ladi. C# tilining sintaksisi uning C# va C++ tillaridan "nusxa" ekanligini

bildiradi. Andres Xijisberg (C# tilining asoschisi) bu tilni yaratayotganda o'z oldiga quyidagi vazifalarni asosiy maqsadi sifatida ishlata boshladi. VS dagi C# tilida loyiha yaratish oynasi.

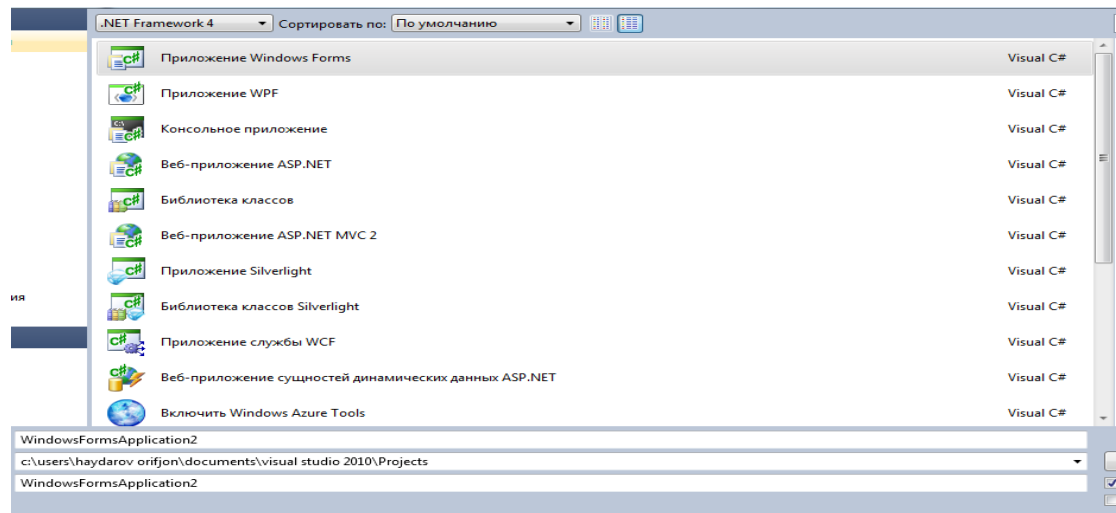
C# dasturlash tili .C# dasturi 90-yillarning oxirida ishlab chiqilib **Microsoft .NET** ning bir qismiga aylandi. Al`fa versiya sifatida 2000 yildan boshlab ishlatila boshladi .C # bosh arxitektori butun dunyo dasturchilari ichida birinchilar qatorida turgan va butun dunyo tomonidan tan olingan dasturlash tilidir. Asoschisi **Anders Hejlsberg** bo`ldi. Uning 1980 yillarda chiqarilgan Turbo Paskal dasturi orqali ham tanishimiz mumkin. C# bevosita C, C++ va Java bilan bog`liq. Chunki bu uchta til dasturlash olamida eng mashhur tillardir. Bundan tashqari profisanal dasturchilar C va C++ ni va juda ko`pchilik Java tilida ish yuritadi.

C# ning kelib chiqish genealogik daraxti quyidagicha bo'ladi.



2.5-rasm. C# ning kelib chiqishi.

Chizmada C va C++ tillari C# ning asosini tashkil qiladi. Lekin C# va Java o`zaro o`zgacha ravishda bog`langan. Ularning kelib chiqishi C va C++ bo`lsada o`zaro bir biridan farq qiladi . C# tili ham obe`ktga mo`ljallangan tillar sirasiga kiradi. VS 2010 da C# tili yordamida 17 xil loyiha yaratish mumkin.



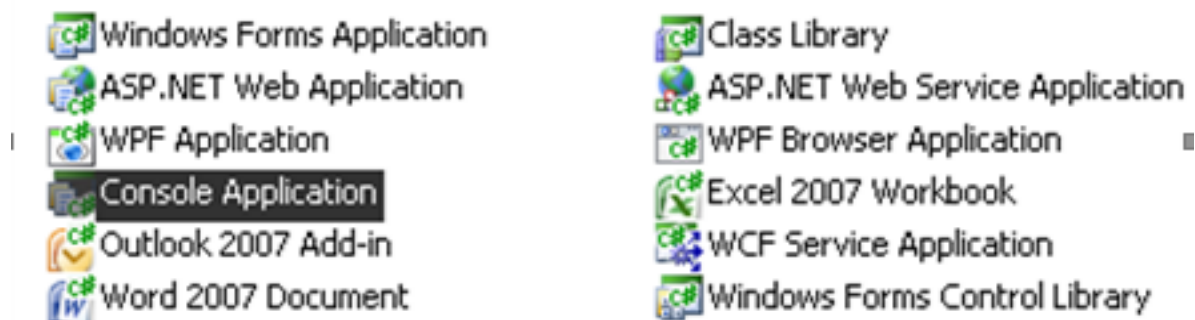
2.6-

rasm. MS VS 2013 da C# tili yordamida 17 xil loyiha yaratish jadvali

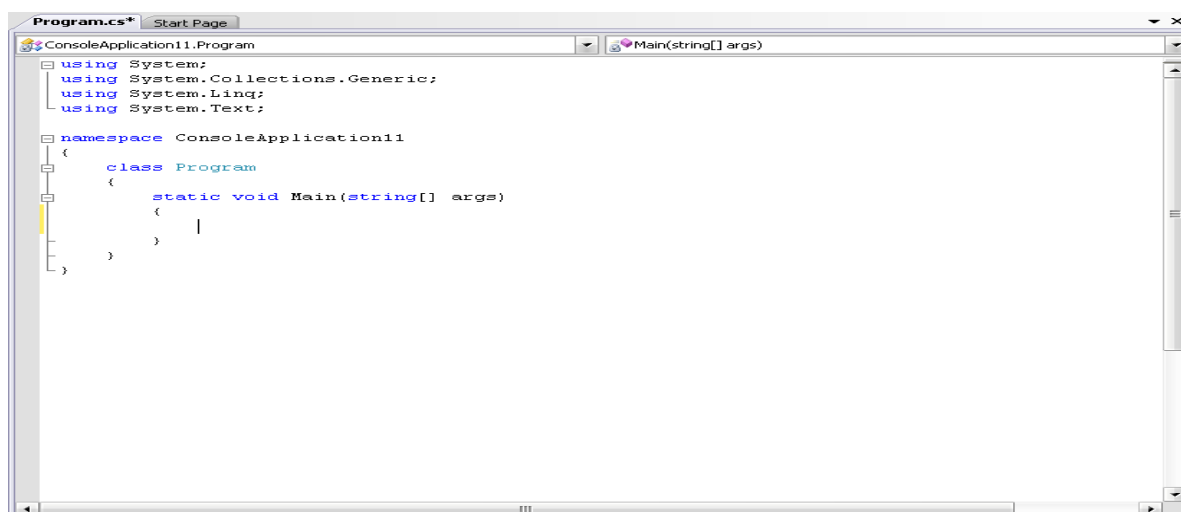
C# tili o`zining interfeysi va sintaksisi uning yuqori sathli dasturlash tillariga kirishiga olib keladi. Uning butun dasturlash tanasi bo`ylab boshqarishni EHM ga topshiradi va faqat bajarilishi lozim bo`lgan shartnigina dasturchi tomonidan yozilishi va shu bilan birga, u tomonidan qo`llaniladigan “aqlli” dasturlash (ya`ni har bir kodning boshlang`ich harfi yoki belgisi kiritilganida u kodning qolgan ismini o`zi namoyish etadi) ham bu dasturlash tilining keng ommaga ma`qul kelishini ta`minlab berdi. VS dagi C# tili ham huddi shu vazifani bajarib beradi. Uning Console Application qismida qora oynali muloqot oynasi yaratiladi va shu qora oyna dastur ishga tushganda paydo bo`ladi. Uning ko`rinishi avtomatik

tarzda VS tomonidan shakllantirilgan bo'lib u dasturchi tomonidan o'zgartirilishi mumkin emas. Consolda ishlatiladigan barcha komponentalar, ularning kodlari, parametrlar boshqa loyihalarda ham qo'llanilishi mumkin. Birinchi navbatda **New->Project->Visual C#->Console Application** buyruqlar ketma ketligi bajarishi kerak va quyidagicha bo'ladi.

Visual Studio installed templates



2.7-rasm Buyruqlar jadvali



```
Program.cs* Start Page
ConsoleApplication11.Program
Main(string[] args)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication11
{
    class Program
    {
        static void Main(string[] args)
        {
            |
        }
    }
}
```

2.2. Dasturning funksional xarakteristikasi

C++ da dastur birta yoki bir necha fayllardan iborat bo'ladi. Har bir fayl o'zida bir yoki bir necha nomlarni saqlab turishi mumkin. Har bir nom esa o'zida qo'yilgan nom yoki tipni, qaysiki sinflar, struktura, interfeys, hisoblash va delegate – funksional tiplardir. C# da yangi loyiha yaratish paytida, VS muhitidagi 10 ta loyiha tiplaridan biri tanlanadi. Bularga Windows Application, Class Library, ASP.NET Application va ASP.NET Web Service misol bo'ladi. Birortasi tanlansa avtomatik tarzda C# va C++ ning shu loyiha tipiga mos qolipi hosil bo'ladi.

1. Birinchi bo'lib C/C++ tillari oilasida ob'yektga yo'naltirilgan dasturlash tilini yaratish;
2. Shunday ob'yektga yo'naltirilgan dasturlash tilini yaratish kerakki, unda hamma narsa ob'yekt sifatida yaratilsin (o'zgaruvchilar, formalar, massivlar, sinflar) ;
3. C++ tilini osonlashtirish, lekin shunday yo'l bilanki, C++ tilining kuchi va konstruksiyalari saqlanib qolsin.

Bu tilning eng katta yangiligi uning ob'yektga murojaati bo'lib, komponentlar yangi loyihalar yaratishdagi tuzilmalarni tuzishdagi barcha muammolarni hal etadi. Komponentlar tuzilishi faqatgina dasturlash tiliga bog'liq bo'lib qolmasdan, balki, uning qanday platformaga ega ekanligiga ham bog'liq.

C# da dasturlashning asosiy qismi: C# sistemasi asosan quyidagi qismlardan iborat. Bular dasturni yozish redaktori, C# tili va standart kutubhonalardir. C# dasturi ma'lum bir fazalardan o'tadi. Birinchisi dasturni yozish va tahrirlash, ikkinchisi preprocessor amallarini bajarish, kompilyatsiya, kutubhonalardagi ob'ekt va funksiyalarni dastur bilan bog'lash (link), hotiraga yuklash (load) va bajarish (execute).

if strukturasi: Biz shartga ko'ra bir necha harakat yo'lidan bittasini tanlaymiz. Misol uchun

agar bolaning yoshi 7 ga teng yoki katta bo'lsa u maktabga borishi mumkin bo'lsin. Buni C# da if ni qo'llab yozamiz.

```
if (yosh >= 7)
```

```
maktab());
```

Bu yerda shart bajarilishi yoki bajarilmasligi mumkin. Agar yosh o'zgaruvchisi 7 ga teng yoki undan katta bo'lsa shart bajariladi va maktab() funksiyasi chaqiriladi. Bu holat true (to'g'ri) deyiladi. Agar yosh 7 dan kichik bo'lsa, maktab() tashlab o'tiladi. Yani false (noto'g'ri) holat yuzaga keladi. Biz shart qismini mantiqiy operatorlarga asoslanganligini ko'rib chiqqan edik. Aslida esa shartdagi ifodaning ko'rinishi muhim emas – agar ifodani nolga keltirish mumkin bo'lsa false bo'ladi, noldan farqli javob bo'lsa, musbatmi, manfiymi, true holat paydo bo'ladi va shart bajariladi. Bunga qo'shimcha qilib o'tish kerakki, C# da mahsus bool tipi mavjud. Bu tipdagi o'zgaruvchilarning yordamida bul (mantiqiy) arifmetikasini amalga oshirish mumkin. bool o'zgaruvchilar faqat true yoki false qiymatlarini olishlari mumkin.

if/else strukturasi: if ni qo'llaganimizda ifoda faqat shart haqiqat bo'lgandagina bajariladi, aks holda tashlanib o'tiladi. if/else yordamida esa shart bajarilmaganda (false natija chiqqanda) else orqali boshqa bir yo'ldan borishni belgilash mumkin. Misolimizni takomillashtirsak. Bola 7 yosh yoki undan katta bo'lsa maktabga, 7 dan kichkina bo'lsa bog'chaga borsin.

```
if (yosh >= 7)
```

```
maktab()); //nuqta-vergul majburiydir
```

```
else
```

```
bogcha());
```

Yuqorida if ga tegishli bo'lgan blok bitta ifodadan (maktab()) iborat. Shu sababli nuqta-vergul qo'yilishi shart. Buni aytib o'tishimizning sababi, masal Pascalda hech narsa qo'yilmasligi shart. C# da bitta ifoda turgan joyga ifodalar guruhini { } qavslarda olingan holda qo'ysa bo'ladi **switch strukturasi:** if-else-if yordami bilan bir necha shartni test qilishimiz mumkin. Lekin bunday yozuv nisbatan o'qishga qiyin va ko'rinishi qo'pol bo'ladi. Agar shart ifoda butun son tipida bo'lsa yoki bu tipga keltirilishi mumkin bo'lsa, biz switch (tanlash) ifodalarini ishlata olamiz. switch strukturasi bir necha case etiketlaridan (label) va majburiy

bo'lmagan default etiketidan iboratdir. Etiket bu bir nomdir. U dasturnig bir nuqtasiga qo'yiladi. Programmaning boshqa yeridan ushbu etiketga o'tishni bajarish mumkin. O'tish yoki sakrash goto bilan amalga oshiriladi, switch blokida ham qo'llaniladi. **while takrorlash strukturasi:** Takrorlash strukturasi bir ifoda yoki blokni ma'lum bir shart to'g'ri (true) bo'lishi davomida qaytarish imkonini beradi. Qaytarilayotgan ifoda shartga ta'sir ko'rsatishi kerak. Ma'lum bir vaqt o'tkandan keyin shart false ga o'zgartilishi kerak. Bo'lmasam while (davomida) tugatilmaydi. while faqat o'zidan keyin kelgan ifodaga ta'sir qiladi. Agar biz bir guruh amallarni qaytarmoqchi bo'lsak, ushbu blokni {} qavslar ichiga olishimiz kerak. Shart takrorlanuvchi blokning boshida tekshirilgani sababli, agar shart noto'g'ri bo'lib chiqsa, blokni hech ijro ko'rmasligi ham mumkin.

```
while (shart) {  
ifoda1;  
ifoda2;  
.... }
```

do/while takrorlash strukturasi:

do/while ifodasi while strukturasi o'xshashdir. Bitta farqi shundaki whileda shart boshiga tekshiriladi. do/while da esa takrorlanish tanasi eng kamida bir marta ijro ko'radi va shart strukturaning so'ngida test qilinadi. Shart true bo'lsa blok yana takrorlanadi. Shart false bo'lsa do/while ifodasidan chiqiladi. Agar do/while ichida qaytarilishi kerak bo'lgan ifoda bir dona bo'lsa {} qavslarning keragi yo'qdir. Quyidagicha bo'ladi:

```
do  
ifoda;  
while (shart);
```

Lekin {} qavslarning yo'qligi dasturchini adashtirishi mumkin. Chunki qavssiz do/while oddiy while ning boshlanishiga o'xshaydi. Buni oldini olish uchun {} qavslarni har doim qo'yishni tavsiya etamiz.

```
do {  
ifoda1;
```

ifoda2;

....

} while (shart);

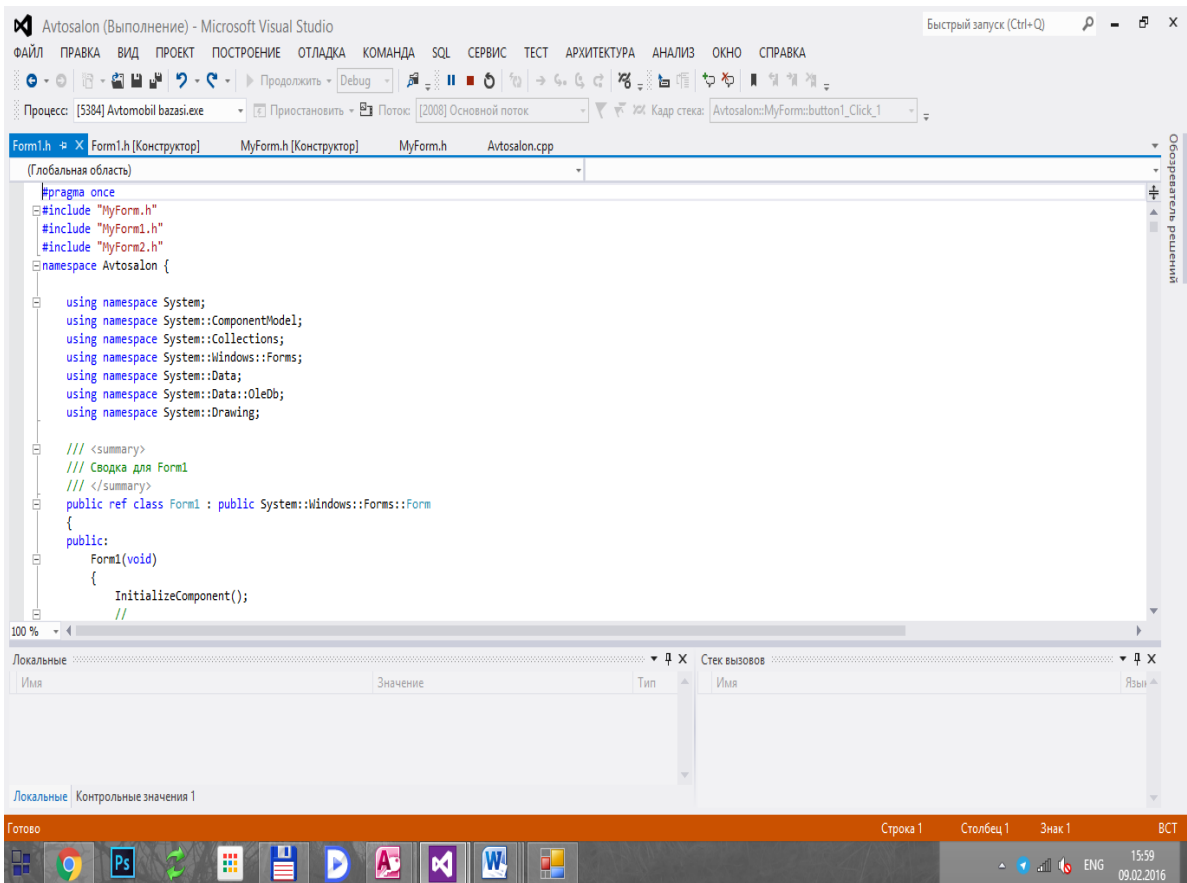
Ko'rsatkichlar va xotirani boshqarish: C++ da ko'rsatkichlar bilan ishlash asosiy o'rinlardan birida turadi. C# da oddiygina dastur tuzilishida ham xavfsiz kod yozish talab etiladi va ko'rsatkichlar va manzillar arifmetikasidan foydalanishni man etadi. Bu esa C# dasturchini ortiqcha xotirani boshqarish ishlari bilan shug'ullanishining oldinini olib, dasturning funksional qismiga ko'proq e'tibor berishini ta'minlaydi.

C# da C++ dagidek ko'p merosxo'rlik va shablon bilan ishlanmaydi, faqatgina interfeys uchungina ishlatilishi mumkin.

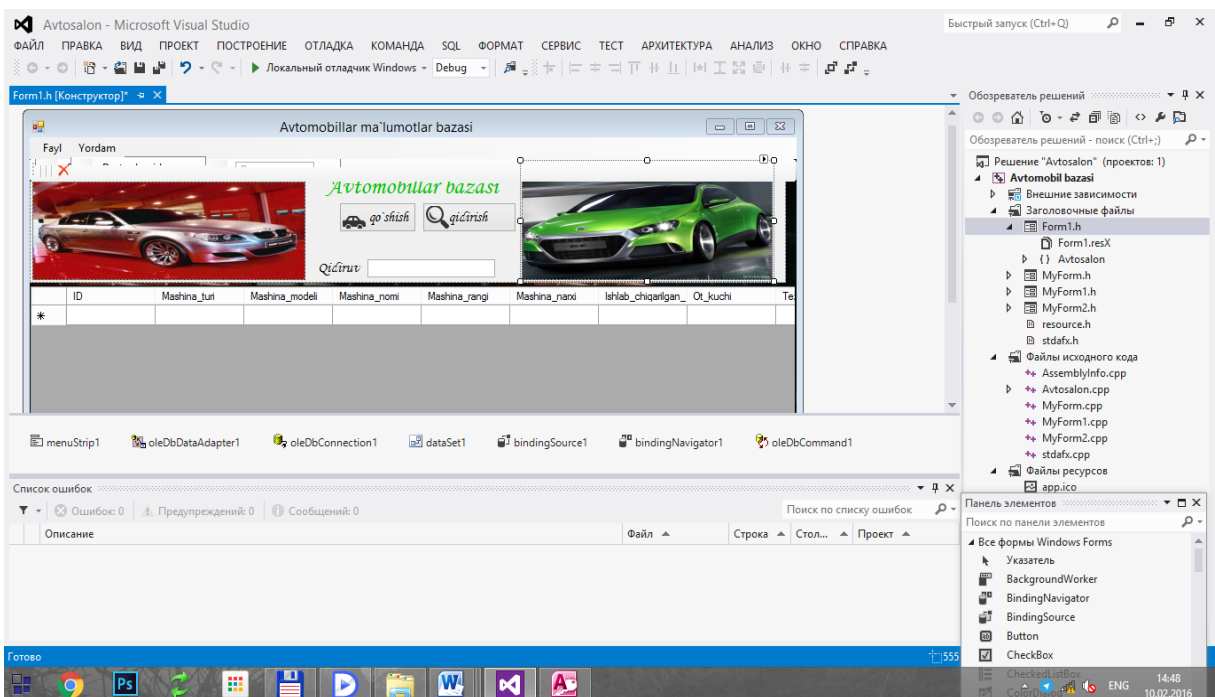
2.3. "Avtomobil ma'umotlar bazasi" dasturini Visual Studio 2012 dasturida C++ tilida yaratish



2.9.-rasm.Asosiy formani visual studioda ko'rinishi



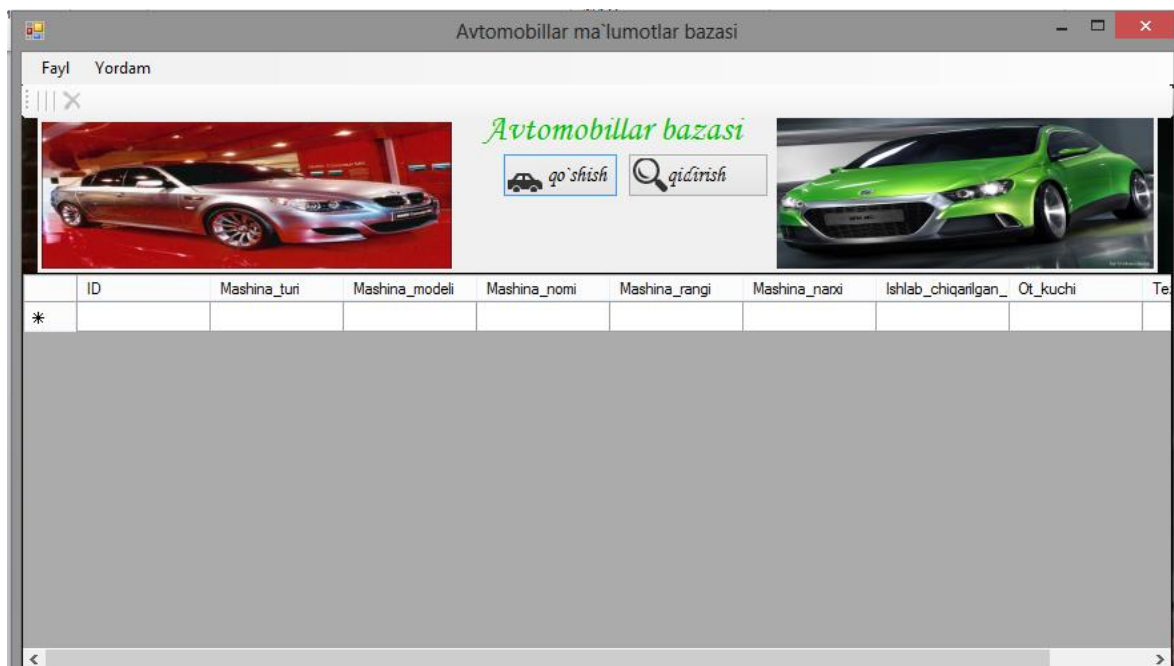
2.10.-rasm



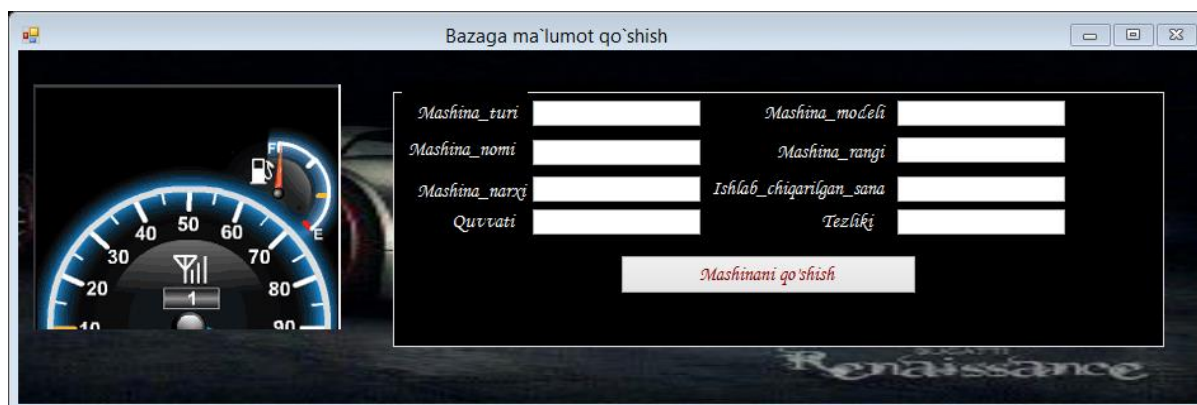
2.11-rasm

2.4. Foydalanuvchiga yo`riqnom

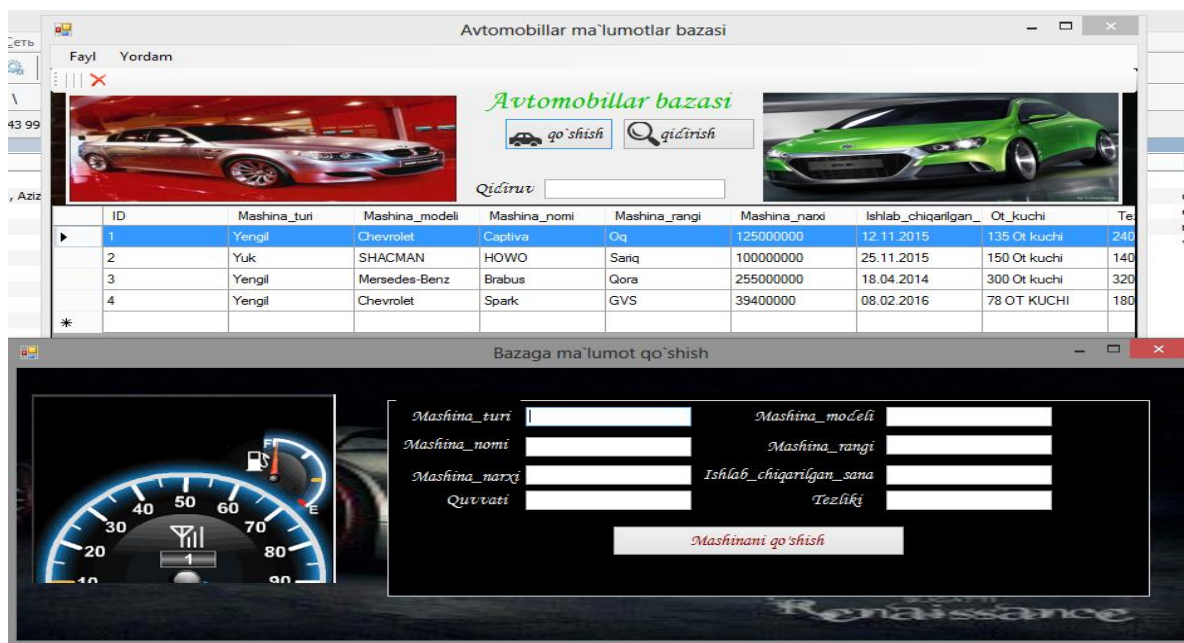
Bu dastur orqali foydalanuvchi avtomobillar ma'lumotlar bazasi haqida to'liq ma'lumotga ega bo'lishi mumkin. Masalan mashina turlari, mashina nomi, mashina modeli, mashina rangi, mashina narxi, mashinaning qachon ishlab chiqarilganliki mashinaning quvvat va mashinaning tezliki haqidagi ma'lumotlarga ega bo'lamiz.



3.1-rasmda Dasturning ko`rinishi



3.2-rasmda dasturga ma'lumot qo'shish mumkin bo'lgan oynasi



3.3-rasmda Dasturga ma'lumot qo'shish va ma'lumotlarni qidirishi oynasi ko'rsatilgan



3.4-rasmda Dastur haqida ma'lumot berilgan

XULOSA

Bu kurs ishini tayyorlash mobaynida olgan bilimlar hamda ko`nikmalarimizni yanada mustahkamladik. Amaliyotda qo`llab natijalarni ko`rdik. Visual Studio dasturlash tilini Microsoft Access ma`lumotlar bazasi bilan bog`lab, shu ma`lumotlar bazasi ustida amallar bajarishni o`rgandik. Va mana bu Avtomashinalar ma`umotlar bazasini yaratish va boshqarish dasturini qo`llab juda kata qulayliklar yaratdik Bu baza orqali avtosalon xodimlari ishini ancha osonlashtirishimiz mumkin. Xozirgi kunda axborotlashish juda tez tarzda rivojlanib bormoqda va hozirgi kunga kelib malumotlarni xajmi juda kattalashib ketdi. Biz bemalol XXI asr axborot asri desak bo`ladi. Buni isbotini biz xar javxada ko`rishimiz mumkin. Shu sababli bizni yurtimizda ham Axborot texnologiyalariga katta axamiyat qaratilmoqda. Biz bu imkoniyatlardan keng foydalangan xolda kelajak texnologiyalarini yaratib O`zbekiston nomini dunyoga mashxur qilamiz. O`ylaymanki qilgan ishlarimiz kelajakda yurtimizda dasturlash texnologiyasi rivojlanishiga o`z hissasini qo`shadi degan umiddaman.

FOYDALANILGAN ADABIYOTLAR

1. M.X.Hakimov,S.M.Gaynazarov. Berilganlar bazasini boshqarish tizimlari.
2. M.Z.Babamuxamedova A.K.Ergashev Operatsion tizimlar va axborotlarga birlamchi ishlov berish.
3. Kamilov M.M., Babamuhamedova M.Z. Kompyuter va avtomatlashtirilgan tizimlarning dasturiy ta'minoti .
4. Raxmonkulova I., G'ulomov S.S., Zayniddinov X.N. IBM PC shahsiy komyuteridan foydalanuvchilar uchun o'quv qo'llanma, 1994-yil.
5. <https://dastur.uz/>
6. <https://htmlbook.ru/>
7. <https://itportal.uz/>
8. <https://CodeForses.ru>
9. <https://GyberFrom.ru>

ILOVA

```
#pragma endregion
    private: System::Void button1_Click(System::Object^ sender,
System::EventArgs^ e) {
        MyForm ^f= gcnew MyForm();
        f->Show();
        f->Text="Bazaga ma`lumot qo`shish";
    }
private: System::Void button2_Click(System::Object^ sender,
System::EventArgs^ e) {
        dataSet1->Clear();
        oleDbDataAdapter1->Fill(dataSet1);
        label1->Visible=true;
        textBox1->Visible=true;
        textBox1->Focus();
        // pictureBox3->Visible=true;
    }
private: System::Void Form1_Load(System::Object^ sender,
System::EventArgs^ e) {
    }
private: System::Void textBox1_TextChanged(System::Object^ sender,
System::EventArgs^ e) {
        oleDbDataAdapter1->SelectCommand->CommandText =
"SELECT * FROM Mashinalar WHERE Mashina_nomi LIKE '%" +textBox1-
>Text+"%' ";
        //oleDbDataAdapter1->SelectCommand-
ExecuteNonQuery();
        DataTable ^dt= gcnew DataTable();
        oleDbDataAdapter1->Fill(dt);
        dataGridView1->DataSource = dt;
    }
private: System::Void dataGridView1_CellContentClick(System::Object^
sender, System::Windows::Forms::DataGridViewCellEventArgs^ e) {
        MyForm1^ inf = gcnew MyForm1;
```

```

        inf->Text      =      Convert::ToString(dataGridView1[1, e-
>RowIndex]->Value)+"      "+Convert::ToString(dataGridView1[1, e-
>RowIndex]->Value)+"      "+Convert::ToString(dataGridView1[2, e-
>RowIndex]->Value);
        inf->name->Text      +=
Convert::ToString(dataGridView1[1, e->RowIndex]->Value);
        inf->label2->Text      +=
Convert::ToString(dataGridView1[2, e->RowIndex]->Value);
        inf->label3->Text      +=
Convert::ToString(dataGridView1[3, e->RowIndex]->Value);
        inf->label4->Text      +=
Convert::ToString(dataGridView1[5, e->RowIndex]->Value);
        inf->label5->Text      +=
Convert::ToString(dataGridView1[8, e->RowIndex]->Value);
        inf->label6->Text      +=
Convert::ToString(dataGridView1[7, e->RowIndex]->Value);
        inf->Show();
    }
private: System::Void      chiqishToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
        this->Close();
    }
private: System::Void
dasturHaqidaToolStripMenuItem_Click(System::Object^      sender,
System::EventArgs^ e) {
        MyForm2 ^f = gcnew MyForm2();
        f->Show();
    }
private: System::Void      saqlashToolStripMenuItem_Click(System::Object^
sender, System::EventArgs^ e) {
        DataTable ^dt= gcnew DataTable();
        // OleDbDataAdapter1->DeleteCommand=;
        OleDbDataAdapter1->Update(dataSet1-
>Tables["Mashinalar"]);

```

```
    }  
private: System::Void panel1_Paint(System::Object^ sender,  
System::Windows::Forms::PaintEventArgs^ e) {  
    }  
private: System::Void saqlashToolStripMenuItem_Click_1(System::Object^  
sender, System::EventArgs^ e) {  
    OleDbDataAdapter1->Update(dataSet1-  
>Tables["Mashinalar"]);  
    dataSet1->Clear();  
    OleDbDataAdapter1->Fill(dataSet1);  
    }  
};  
}1
```