

**O'ZBEKISTON RESPUBLIKASI OLIIY VA O'RTA MAXSUS TA'LIM
VAZIRLIGI**

BUXORO DAVLAT UNIVERSITETI

FIZIKA-MATEMATIKA FAKULTETI

“Amaliy matematika va axborot texnologiyalari” kafedrası

Haydarov Orifjon Rustamovich

YO'LLANMALAR NAZORATI VA QAYT QILISH DASTURI

**5480100-“Amaliy matematika va informatika” ta'lim yo'nalishi bo'yicha
bakalavr darajasini olish uchun**

BITIRUV MALAKAVIY ISHI

**«Ish ko`rildi va himoyaga
ruxsat berildi »**

**Ilmiy rahbar: _____ dots. B.B. Mo'minov
“ ____ ” _____ 2014 y.**

**Kafedra mudiri :
_____ dots.T.B. Boltayev
« ____ » _____ 2014 y.**

**Taqrizchi:

« ____ » _____ 2014 y.**

«Himoya qilishga ruxsat berildi »

Fakultet dekani _____ prof. Sh.M. Mirzayev

“ ____ ” _____ 2014 y.

Buxoro-2014

MUNDARIJA

KIRISH	3
I BOB. Yo'llanma dasturini yaratishning nazariy asoslari	
1.1 Microsoft Visual Studio haqida.....	7
1.2 C# dasturlash tili haqida.....	12
II BOB. Yo'llanma dasturini yaratish.	
2.1 Ma'lumotlar ba'zasini boshqarish tizimi SQL Server 2008 haqida...	19
2.2 Yo'llanma dastur algoritimini yaratish.....	25
2.3 Yaratilgan dasturdan foydalanish.....	36
XOTIMA	46
ADABIYOTLAR	47

KIRISH

Hozirgi kunda kompyuter texnologiyalari kirib bormagan soha deyarli uchramaydi. Kompyuter texnologiyalardan nafaqat hisoblash ishlarini olib borish uchun emas, balki hayotga tadbiq qilinadigan elektron darsliklar, chizma va video tasmlarni qayta ishlovchi, katta hajmli ma'lumotlarni o'zida saqlovchi dasturlar yaratish uchun ham foydalaniladi. So'nggi yillarda kompyuter va uning dasturiy ta'minotiga bo'lgan talab va qiziqishlar ortib bormoqda. Bu esa o'z navbatida dasturchidan katta izlanish va mahoratni talab qiladi.

Prezidentimiz I. A. Karimovning "Yoshlarimiz bizdan ko'ra kuchli bilimli dono va albatta baxtli bo'lishlari kerak", "O'zbekiston kelajagi yoshlar qo'lida" degan so'zlari yoshlarga bo'lgan e'tiborni va katta ishonchni ko'rsatadi.

Prezident I.A.Karimov tashabbusi bilan qabul qilingan "Ta'lim to'g'risida"gi qonun va Kadrlar tayyorlash Milliy dasturi ta'limning Yangi milliy modelini yaratilishiga asos bo'ldi. Milliy modelning asosiy maqsadi ta'lim sohasini isloh qilish, mutaxassislar tayyorlashning yangi tizimini joriy etish va ta'lim sifatini hozirgi zamon talablari darajasiga ko'tarishdir. Ushbu maqsadni bitta qonun yoki farmonlar bilan amalga oshirib bo'lmaydi. U mashaqqatli mehnat va izlanishlarni talab qiladi. Bu muammoga hukumatimiz tomonidan alohida e'tibor bilan qaralmoqda. Respublikamiz Prezidenti I.A.Karimovning 2001 yil Oliy Majlisning 5-sessiyasida so'zlagan nutqida axborot texnologiyalari va kompyuterlarni jamiyat hayotiga, kishilarning turmush tarziga, maktab va oliy o'quv yurtlariga jadallik bilan olib kirish g'oyasi ilgari surilgan edi. Prezidentimiz tashabbusi bilan O'zbekiston Respublikasi Vazirlar Mahkamasining 2001-yil 23-maydagi 230-sonli «2001—2005 yillarda kompyuter va axborot texnologiyalarini rivojlantirish», shuningdek, «Internet»ning xalqaro axborot tizimlariga keng kirib borishini ta'minlash dasturini ishlab chiqishni tashkil etish chora-tadbirlari to'g'risida»gi Qarorlari qabul qilindi. Oradan bir yil o'tgach, 2002-yil 30-mayda O'zbekiston Respublikasi Prezidentining «Kompyuterlashtirishni yanada rivojlantirish va uning ijrosini amalga oshirish yuzasidan Vazirlar Mahkamasining 2002-yil 6-iyundagi «2002—2010 yillarda kompyuterlashtirish va axborot-

«kommunikatsiya texnologiyalarini rivojlantirish dasturi» to'g'risidagi Qarori e'lon qilindi.

Ta'lim sifatini oshirish bugungi kunning asosiy vazifalaridan biridir. Chunki bugungi kunda insoniyat rivojlanishi shunday bosqichga etib bordiki, bu bosqichda inson aql-zakovati, bilimi va uning mutaxassis sifatidagi malakasi hal qiluvchi ro'l o'ynaydigan bo'lib qoldi.

Ta'lim sifatini oshirish o'quv dargohlarini zamonaviy o'quv rejalari, dasturlar, o'quv jihozlari bilan ta'minlash, yangi darsliklar va qo'llanmalarni yaratish, ilg'or pedagogik texnologiyalarni joriy etishni taqazo etadi.

Albatta, jihozlar, darsliklar, texnologiyalar ham ushbu vazifani to'liq bajarilishiga olib kela olmaydi. Chunki ta'lim sifati ko'p jihatdan ta'lim beruvchining shaxsiga va malakasiga bog'liqdir.

Ta'lim jarayonini tashkil etish, uni etarli materiallar bilan ta'minlash, o'quv jihozlaridan maqsadli foydalanish, o'quvchi fikrini darsga doimiy jalb etish o'qituvchidan katta mahorat, bilim va ko'nikma talab etadi.

Yangi axborot – kommunikatsion texnologiyalari hozirgi vaqtda eng dolzarb mavzulardan biri bo'lib kelmoqda, sababi har bir sohani o'rganish, izlanish va tajriba orttirish uchun turli usullardan foydalanish kerak bo'ladi. Shuning uchun yangi axborot – kommunikatsion texnologiyalardan foydalanish maqsadga muvoffiqdir.

Hozirgi zamon mutaxasislari, faoliyat doiralari qanday bo'lishidan qat'iy nazar informatika bo'yicha keng ko'lamdagi bilimlarga, zamonaviy hisoblash texnikasi, informatsion aloqa va kommunikatsiya tizimlari, orgtexnika vositalari va ulardan foydalanish borasida yetarli malakalarga ega bo'lishi, hamda yangi information texnika va texnologiya asoslarini uning ertangi kuni, rivoji to'g'risidagi bilimlarni o'zida mujassamlashtirgan bo'lishi kerak. Zamonviy hisoblash texnikasi va informatsion texnologiyalarning kun sayin rivojlanib, jamiyatning esa tobora informatsiyalashib borishi sababli, uzluksiz ta'lim tizimining o'rta va yuqori bosqichlariga informatika, ishlab chiqarish va boshqarish jarayonlarini kompyuterlashtirish bo'yicha bir qator o'quv fanlari

kiritilgan. Yigirmanchi asr o'rtalariga kelib tezkor mashina mexanizmlaridan foydalana boshlandi. Murakkab texnika va texnologiyalar o'ylab topildi. Ko'pgina masalalarni hal qilish jarayonida axborot hajmi behisob bir majmuaga aylandi, hamda bu axborotlarni yig'ish va uzatish vositalarini yaratish, ularni vaqtida qayta ishlab, boshqarish uchun zarur bo'lgan choralarni belgilab chiqish kerak bo'ladi. Ko'pchilik vazifalarni bajarishda boshqarish jarayonini takomillashtirish, axborot tizimini joriy etish, mutaxassislarni kompyuterda ishlashga o'rgatish muhim ahamiyatga ega.

Bitiruv malakaviy ish mavzuning dolzarbligi: Mavzuning dolzarbligi shundan iboratki, Visual Studio muhitida loyiha fayllarini yaratishi, C# dasturlash tilida ishlashini mukammallashtirish, korxona ishchi xodimlariga yo'llanma berish jarayonini o'rganib chiqish va uni ishini avtomatlashtirish. Berilganlar bazasini boshqarish tizimini o'rganish va unda loyihasini shakllatirish. Hozirgi paytda axborot texnologiyalariga juda e'tibor kattadir. Shuning uchun biz korxona ishini osonlashtirish uchun yo'llanmalar nazorati va qayt qilish dasturini bajarishni oldimizga maqsad sifatida qo'yildi.

Tadqiqot ob'yekti va predmeti: Korxonaning kasaba uyushmasini ish faoliyati bilan tanishish. Visual Studio muhiti bilan ishlash. C# dasturlash tili va unda loyihalar yaratsih. Berilganlar bazasini boshqarish tizimlari bilan tanishish. SQL so'rovlar tili bilan ishlash va uni mukammal o'rganish. SQL Server muhiti bilan tanishish. SQL server 2008 da dasturni bazasini loyihasini shakllantirish. So'rovlar bilan ishlash va undan C# tiliga foydalanish.

Ishning maqsadi va vazifalari: Mavzu bo'yicha yetarli adabiyotlarni jamlash. Microsoft Visual Studio 2010 muhiti bilan ishlashni o'rganish. C# dasturlash tili bilan ishlash va undan foylana olish. SQL so'rovlar tilini o'rganish. Microsoft SQL Server 2008 muhitidan foydalanish. Jadval orasidagi bog'lanishlarni va normal formalarni o'rganish. SQL so'rovlarini yoza olish va undan C# dastrulash tiliga qo'llash. Ishchi xodimlarga beriladigan yo'llanmalarni qayd qilish va nazorat qilish uchun berilganlar bazasini va foydalanuvchi interfeysini yaratishdan iborat.

Tadqiqot usuli va uslubiyoti: C# dasturlash tilini imkoniyatlarini ko'rsatish, zamonaviy texnologiyalardan foydalanish. Zamonaviy dasturlash usullaridan foydalanish. Berilganlar bazasini boshqarish tizimlarini imkoniyatlaridan foydalanish.

Olingan asosiy natijalar: Microsoft Visual Studio 2010 muhiti imkoniyatlarini ko'rsatish. C# dasturlash tilini imkoniyatlarini namoyish qilish. C# dasturlash tilining obyektlariga hodisalarni qo'shish va undan foydalanish. Visual Studio muhitini C# tilida, yo'llanmalar nazorati va qayt qilish dasturi yaratish.

Natijalarning ilmiy yangiligi va amaliy ahamiyati: Yangi chiqayotga dasturlash tillari bilan tanishish va undan foydalanishni o'rganish. Bu dasturni yaratish uchun Microsoft Visual Studio muhitidagi C# tilini imkoniyatlarini ko'rsatish. O'rgangan bilimlar asosida yo'llanmalar nazorati va qayt qilish dasturini yaratish. Ushbu dasturni korxonaning kasaba uyushmasi uchun mo'ljallangan bo'lib bu dasturdan foydalanish juda katta ahamiyatga ega.

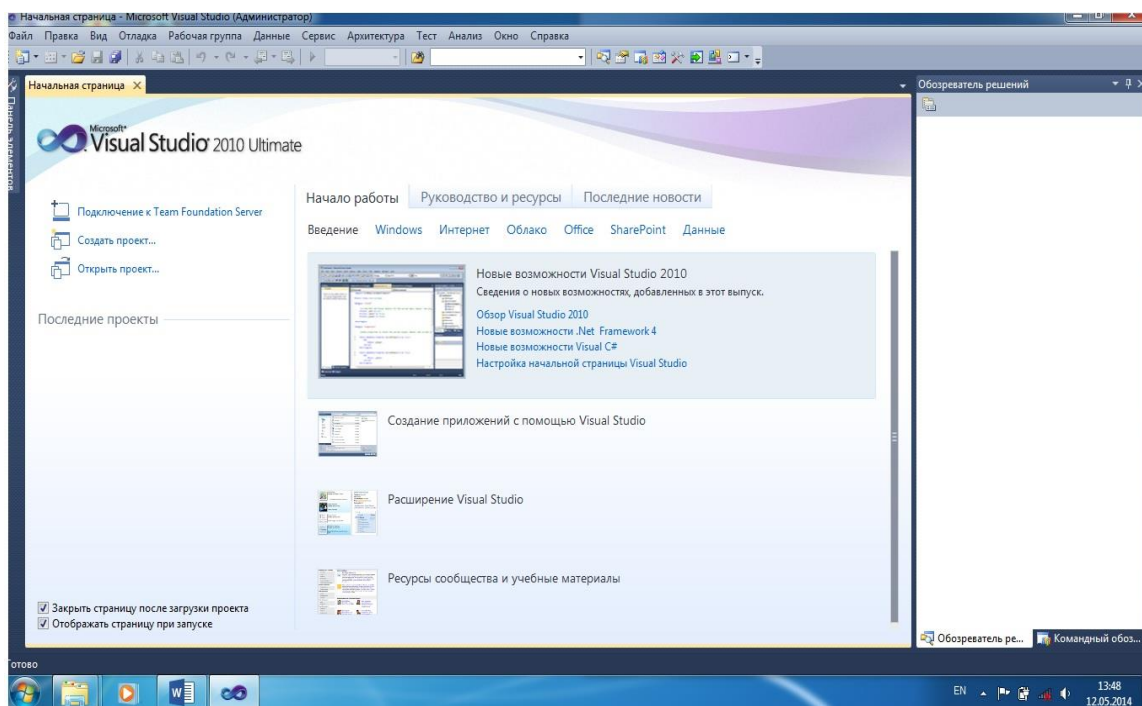
Tadbiq etish darajasi va iqtisodiy samaradorligi. Qo'llanish sohasi. Xulosa va takliflar: Tashkilotlarga yo'llanma berishi uchun qayd qilish dasturini zarurligi sababli ko'rsatilgan tashkilotga shu dastur tadbiq qilinadi. Bu dasturdan foydalanishda ishchi xodimning vaqti, va qog'oz variantdan chiqishi, bir necha yillik hujjatlarni bir zumda bila olishi iqtisodiy samaradorligidir. Korxonani kasaba uyushmalarini ish jarayonini osonlashtirish uchun qo'llanilsa maqsadga muvofiq bo'ladi. Bu dasturni tegishli korxonalarga amalga qo'llanilishini taklif etaman.

Ishning hajmi va tuzilishi: Kirish, 2 ta bob, har bir bobning qisqacha xulosasi, foydalanilgan adabiyotlar ro'yxati va xotimalardan iborat holda bayon qilingan. I bob Yo'llanma dasturini yaratishning nazariy asoslari bo'lib bunda Microsoft Visual Studio, C# dasturlash tili, Ma'lumotlar Ba'zasini Boshqarish (MBB) tizimlari haqida ma'lumot keltirilgan. II bob Yo'llanma dasturini yaratish bo'lib bunda Yo'llanma dasturi uchun texnik loyiha tayyorlash, Dastur algoritimini yaratish, Yaratilgan dasturdan foydalanish haqida ma'lumot keltirilgan bo'lib 47 betdan iborat.

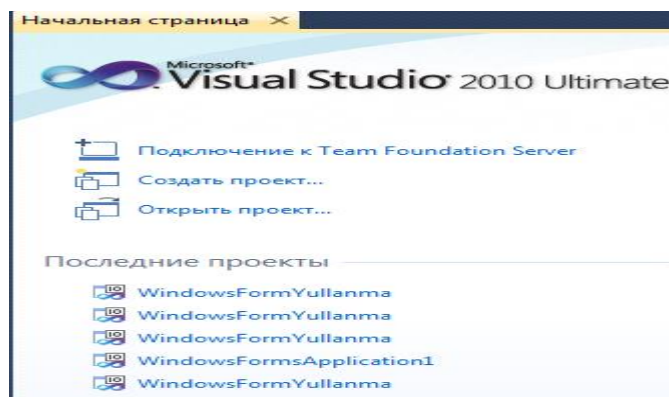
I BOB. Yo'llanma dasturini yaratishning nazariy asoslari

1.1 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 dastrulash tillari oynalarga bo'lingan va shu oynalar yordamida bir biriga bog'langan. VS yuklash uchun **пуск->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.



1.1.1-chizma. MS Visual Studio muloqot oynasi.



1.1.2-chizma Bosh sahifa oynasi

1.Recent Projects – avval ishlangan loyihalardan 6 tasi ekranda ko'rsatilgan bo'ladi, ularni ustiga sichqonchaning 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 komponentni 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, Immedite 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

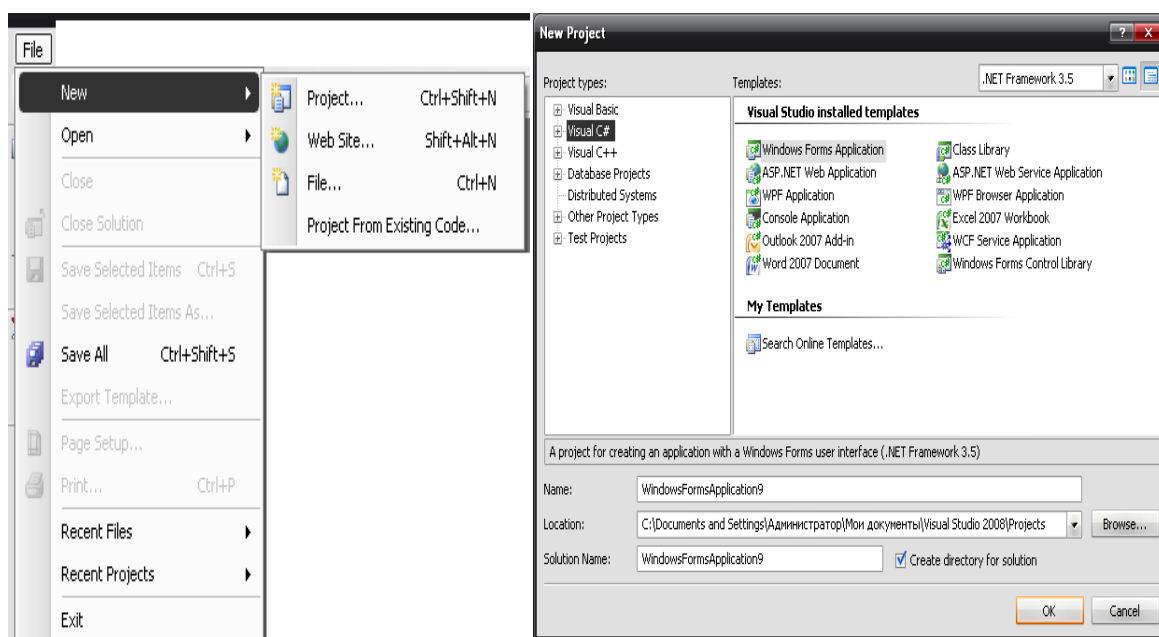
- **Show Data Sources** – Shu loyiha bilan bog'langan barcha ma'lumotlar bazalarini ko'rsatib beradi;
- **Add New Data Sources** – Yangi ma'lumotlar bazasini qo'shish uchun ishlatiladi;
- **Schema Compare** – Shu loyihada mavjud bo'lgan ma'lumotlar bazalari orasidagi barcha bog'liqliklarni ko'rsatib 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. 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.



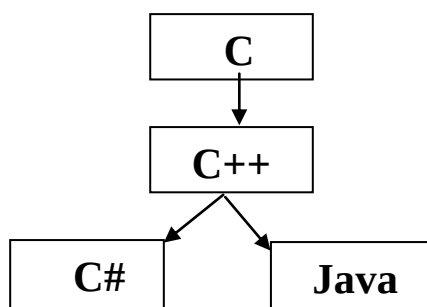
1.1.3-chizma.VS dagi C# tilida loyiha yaratish oynasi.

1.2. 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.

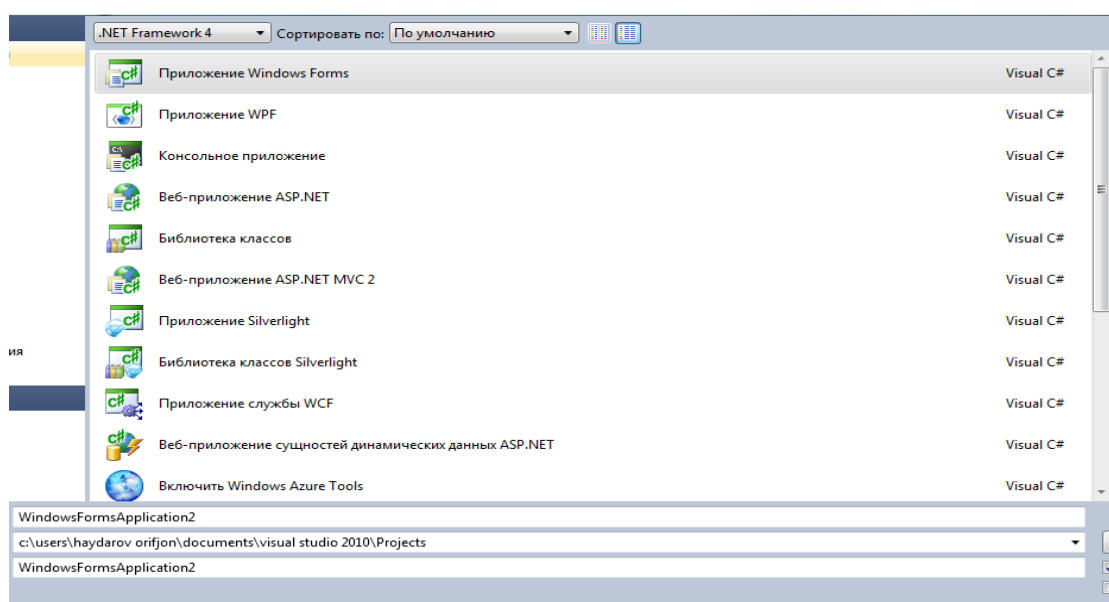
C# bevosita C, C++ va Java bilan bog`liq. Chunki bu uchta til dasturlash olamida eng mashhur tillardir. Bundan tashqari profisanal dasturchilar C va C++ ni va juda ko`pchilik Java tilida ish yuritadi.

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



1.1.4-chizma. C# ning kelib chiqishi.

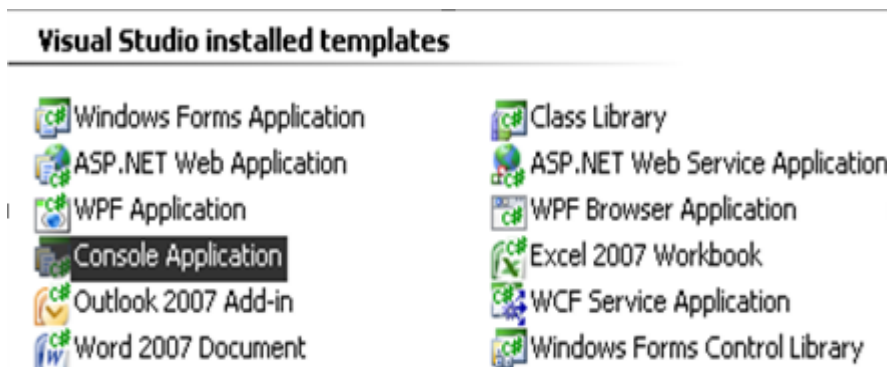
Chizmada C va C++ tillari C# ning asosini tashkil qiladi. Lekin C# va Java o'zaro o'zgacha ravishda bog'langan. Ularning kelib chiqishi C va C++ bo'lsada o'zaro bir biridan farq qiladi. C# tili ham obe'ktga mo'ljallangan tillar sirasiga kiradi. VS 2010 da C# tili yordamida 17 xil loyiha yaratish mumkin.



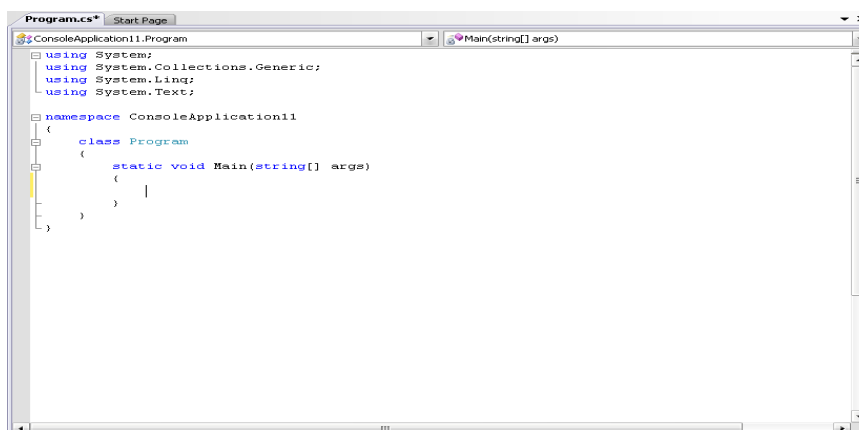
1.1.5-chizma.MS VS 2010 da C# tili yordamida 17 xil loyiha yaratish jadvali

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

qismini 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.



1.1.6-chizma Buyruqlar jadvali



1.9-chizma. Muloqot oynasi.

Dastur tuzilishi: 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 preprosessor 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 strukturalasi: 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
    bog'cha();
```

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 strukturalasi: 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 strukturalasi 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.

Bitiruv malakaviy ishining birinchi bobi yuqorida ta'kidlab o'tilganidek, "Yo'llanma dasturini yaratishning nazariy asoslari" deb, nomlanadi. Bu bobning o'zi ham ikki qismga bo'lingan bo'lib, birinchi qismi "Microsoft Visual Studio 2010 haqida" deb nomlanadi. Bu qismda Microsoft Visual Studio haqida, menyular paneli bilan ishlash, hamda yangi loyiha yaratish imkoniyatlari bayon etilgan. Ikkinchi qismi esa "C# dasturlash tili haqida" deb nomlanib unda dastur tuzilishi, Visual Studio 2010 da C# tili yordamida loyiha yaratish, dastur tuzilishi, C# dasturlash tilining strukturalarining imkoniyatlari ko'rsatilgan.

II BOB Yo'llanma dasturini yaratish.

2.1 Ma'lumotlar ba'zasini boshqarish tizimi SQL Server 2008 haqida

Ma'lumotlar bazasi dunyosi tobora yagona bo'lib bormoqda. Bu jarayon har xil kompyuter muxitlarida faoliyat ko'rsatuvchi axborot tizimlarini yaratishda qo'llanuvchi yagona standart til yaratishni talab qiladi. Standart til bir komandalar to'plamini bilgan foydalanuvchilarga ularni shaxsiy kompyuter tarmoq ishchi stantsiyasi yoki katta EXM da ishlashlaridan qat'iy nazar ma'lumotni yaratish, izlash va uzatishga imkon beradi.

Berilganlar bazasini tashkil etuvchi elementlar turli ko'rinishda bo'lishi mumkin. Eng ko'p tarqalgan va amaliyotda qo'llaniladigan ma'lumotlar matnli fayllar hisoblanadi. Chunki matnli fayllar orqali turli axborotlarni ifodalash va kompyuter xotirasida saqlash mumkin.

Kompyuterdagi axborot tizimlarining ko'rinishlaridan biri berilganlar bazasi hisoblanadi. Oddiy fayllardan farqli ravishda berilganlar bazasi kompyuter xotirasida joylashgan axborotlarni izlash va saralashni amalga oshirish imkoniyatiga ega.

MS SQL Server BBT Microsoft firmasi mahsuloti. MS SQL Server – 2008 ga XML va HTTP protokoloni qo'llash kiritilgan. U katta hajmli berilganlar bazasi bilan ishlashga, yaxshi himoya vositalariga, keng imkoniyatlarga va minglab foydalanuvchilarga bir vaqtning o'zida xizmat qilish quvvatiga hamda har qanday web xizmatlarni qo'llash imkoniga ega.

SQL (Structured Query Language, odatda "sikvel" deyiladi) ma'nosi *Tarkiblangan so'rovlar tili*. Bu relyatsion ma'lumotlar bazalarida ishlashga imkon beradigan tildir. Bu til ifodalarining xususiyati shundan iboratki ular ma'lumotlarni qayta ishlash protseduralariga emas natijalariga yo'naltirilgandir. SQL o'zi ma'lumotlar qayerda joylashgani, qanday indekslar va hatto amallarning eng effektiv ketma ketligini qo'llash kerakligini aniqlaydi.

SQL tilining o'zi IBM kompaniyasida MBBT DB2 yaratish jarayonida ishlab chiqilgan va keng ko'lamda RISC protsessorli mashinalarda UNIX tizimlar

asosida, hamda meynfreymlarda, superkompyuterlar asosida qurilgan katta hisoblash tizimlarida qo'llanilgan.

Shu bilan birga mustaqil bo'lmagan PL/SQL, va Transact-SQL kabi ichki dasturlash tillariga inkapsulyatsiya qilinadi. 1986 yilda, ANSI (American National Standart Institute) SQL tilining rasmiy standartini ishlab chiqdi, 1992 yil bu standart kengaytirildi. Butun til 30 ga yaqin operatorlarga ega bo'lib, ba'zi versiyalarida sal ko'proq, ba'zilarida sal kamroq. Har qanday MB har xil ob'ektlarga ega, Ya'ni jadvallar, protseduralar, funktsiyalar, tasavvurlar, ketma ketliklar va xokazo.

"Klient-Server" texnologiyasiga ko'ra, foydalanuvchi EXM (Klient) lar so'rovlari maxsus ma'lumotlar serverlarida (Server) qayta ishlanadi, foydalanuvchi EXM larga faqat so'rovni qayta ishlash natijalari qaytariladi.

Tabiiyki Server bilan muloqot qilish uchun yagona til kerak va bunday til sifatida SQL tanlandi. Shuning uchun hamma zamonaviy relyatsion MBBT versiyalari (DB2, Oracle, Ingres, Informix, Sybase, Progress, Rdb) va hattoki norelyatsion MBBT versiyalari (masalan, Adabas) "Klient_Server" texnologiyasi va SQL tilidan foydalanadilar.

SQL tilida Ma'lumotlarni jadval ko'rinishda tasvirlashga yo'naltirilgan amallar kontseptsiyasi ko'p bo'lmagan (30 dan kam) ifodalardan iborat kompakt til yaratishga iborat bo'ladi.

Ikki xil SQL mavjud: **Interaktiv** va **Joylashtirilgan**. Ko'p xollarda ikkala forma bir xil ishlaydi, lekin ikki xil foydalaniladi:

Interaktiv SQL ma'lumotlar bazasi o'zida faoliyat ko'rsatadi va buyurtmachi foydalanishi uchun chiqish xosil qilish uchun ishlatiladi. SQL bu formasida, siz komanda kiritsangiz, u darov bajariladi, va siz darhol natijani (agar u mavjud bo'lsa) ko'rishingiz mumkin.

Joylashtirilgan SQL boshqa tilda yaratilgan dasturga joylashtirilgan SQL komandalardan iborat.

SQL Interaktiv, va joylashtirilgan formalarida ko'p sonli guruxlar yoki sub

bo'limlar mavjud. Ular ANSI tomonidan e'tiborga olingan va kontseptual darajada foydali, lekin ko'pchilik SQL dasturlar ularni aloxida qayta ishlamaydi, shuning uchun ular aslida SQL komandalarining funktsional kategoriyalaridir.

- **DDL** (*Ma'lumotlarni Ta'riflash Tili*) - ANSI da Sxemani ta'riflash tili, ob'ektlarni(jadvallar, indekslar, tasavvurlar va xokazo) yaratuvchi komandalardan iborat.

- **DML** (*Ma'lumotlarni O'zgartirish Tili*) - bu ixtiyoriy daqiqada jadvallarda qanday qiymatlar saqlanishini aniqlovchi komandalar majmuasidir.

- **DCD** (*Ma'lumotlarni Boshqarish Tili*) foydalanuvchiga ma'lum ob'ektlar ustida ma'lum ta'sir o'tkazishga ruxsat berish yoki bermaslikni aniqlovchi vositalardan iborat.

SQL Standarti **ANSI** tomonidan aniqlangan va hozirda **ISO** tomonidan qabul qilingan. Lekin kommertsial ma'lumotlar bazalari dasturlari ANSI ni ogoxlantirmasdan SQL ni kengaytiradilar, ya'ni foydali hisoblagan har xil xossalar qo'shadilar.

SQL tilida ma'lumotlar turlari. SQL tilida quyidagi asosiy ma'lumotlar turlari ishlatilib, ularning formatlari har xil MBBT lar uchun farq qilishi mumkin:

INTEGER - butun son (odatda 10 tagacha qiymatli raqam va ishora).

MALLINT- "qisqa butun" (odatda 5 tagacha qiymatli raqam va ishora).

DECIMAL(p,q) - o'nli son, p raqam va ishoradan iborat ($0 < p < 16$). O'nli nuqtadan so'ng raqamlar soni q orqali beriladi ($q < p$, agar $q = 0$ bo'lsa, tashlab yuborilishi mumkin).

FLOAT- haqiqiy son 15 ta qiymatli raqam va butun darajadan iborat. Daraja MBBT tipi bilan aniqlanadi (masalan, 75 yoki 307).

CHAR(n) - uzunligi o'zgarmas, n ga teng bo'lgan simvolli qator ($0 < n < 256$).

VARCHAR(n) - uzunligi o'zgaruvchi, n simvoldan oshmagan simvolli qator ($n > 0$ va har xil MBBT larda har xil lekin 4096 dan kam emas).

DATE-maxsus komanda orqali aniqlanuvchi formatdagi sana; sana maydonlari bizning eramizdan oldin bir necha mingyilliklardan boshlanuvchi va bizning eramiz beshinchi- o'ninchi mingyilligi bilan cheklangan haqiqiy sanalarni o'z ichiga olishi mumkin.

TIME-maxsus komanda orqali aniqlanuvchi formatdagi vaqt (ko'zda tutilgan bo'yicha hh.mm.ss).

DATETIME- sana va vaqt kombinatsiyasi.

MONEY- maxsus komanda orqali aniqlanuvchi formatdagi pul.

Jadvallarni yaratish. Jadvallar CREATE TABLE komandasi bilan yaratiladi. Bu komanda qatorlarsiz bo'sh jadval yaratadi. CREATE TABLE komandasi jadval nomini va jadval o'zini ma'lum tartibda ko'rsatilgan ustunlar nomlari ketma – ketligi ta'rifi ko'rinishida aniqlaydi. U ma'lumotlar tiplari va ustunlar o'lchovini aniqlaydi. Har bir jadval juda bo'lmaganda bitta ustunga ega bo'lishi kerak.

Misol uchun o'quvchilar jadvalini yaratishni ko'rib chiqamiz:

CREATE TABLE Salepeople

**(SNum integer,
SName char (10),
City char (10),
Comm decimal);**

Jadvallarni o'chirish. Jadvalni o'chirish imkoniga ega bo'lish uchun, jadval egasi (Ya'ni yaratuvchisi) bo'lishingiz kerak. Faqat bo'sh jadvalni o'chirish mumkin. Qatorlarga ega bo'lgan, to'ldirilgan jadvalni o'chirish mumkin emas, Ya'ni jadval o'chirishdan oldin tozalangan bo'lishi kerak. Jadvalni o'chirish komandasi quyidagi ko'rinishga ega:

DROP TABLE < table name >;

Masalan: **DROP TABLE Salepeople;**

Jadvalni yaratilgandan so'ng o'zgartirish. Jadvalni o'zgartirish uchun ALTER TABLE komandasidan foydalaniladi. Bu komanda jadvalga Yangi

ustunlar qo'shish, ustunlarni o'chirish, ustunlar kattaligini o'zgartirish, hamda cheklanishlarni qo'shish va olib tashlash imkoniyatlariga ega. Bu komanda ANSI standarti qismi emas, shuning uchun har xil tizimlarda har xil imkoniyatlarga ega. Siz INSERT komandasidan bir jadvaldan qiymat tanlab, so'rov bilan ishlatish uchun, ikkinchisiga joylashishda foydalanishingiz mumkin. Buning uchun siz VALUES ifodasini /mos so'rovga almashtiringiz kerak:

```
INSERT INTO Londonstaff  
SELECT * FROM Salespeople  
WHERE City = 'London';
```

Satrlarni o'chirish. Satrlarni jadvaldan DELETE komandasi bilan o'chirish mumkin. U aloxida qiymatlarni emas faqat satrlarni o'chiradi. DELETE quyidagi formatga ega:

```
DELETE FROM <table name | view name>  
[WHERE search-condition];
```

Masalan, o'quvchilar jadvalidagi hamma satrlarni o'chirish uchun, quyidagi shartni kiritish mumkin:

```
DELETE FROM Salepeople;
```

Ma'lum satrlarni o'chirish uchun predikatdan foydalaniladi.

UPDATE komandasi yordamida jadval o'zgartirish bajariladi. Bu komandada UPDATE ifodasidan so'ng jadval nomi va SET ifodasidan so'ng ma'lum ustun uchun o'zgartirish ko'rsatiladi. UPDATE ikki formatga ega. Ulardan birinchisi:

```
UPDATE <table name | view name>  
SET column = expression [, column = expression] ...  
[WHERE search-condition]
```

bu erda expression - bu ustun | ifoda | konstanta | o'zgaruvchi.

Ikkinchi variant:

```
UPDATE <table name>  
SET column = expression, ...  
[ FROM table-list ]
```

[**WHERE search-condition**]

SELECT ifodasi o'z ichiga albatta **FROM** ifodasini olishi kerak. Qolgan ifodalar kerak bo'lsa ishlatiladi.

SELECT ifodasidan so'ng so'rovda qaytariluvchi ustunlar ro'yxati yoziladi.

FROM ifodasidan so'ng so'rovni bajarish uchun jadvallar nomi yoziladi.

WHERE ifodasidan so'ng agar ma'lum satrlarni qaytarish lozim bo'lsa, izlash sharti yoziladi.

GROUP BY ifoda guruxlarga ajratilgan natijaviy so'rov yaratishga imkon beradi.

HAVING ifodasidan guruxlarni qaytarish sharti yoziladi va **GROUP BY** bilan birga ishlatiladi.

ORDER BY ifodasi ma'lumotlar natijaviy to'plamini tartiblash yo'nalishini aniqlaydi.

Jadvallarni o'zi bilan jamlash. Jadvallarni o'zi bilan jamlash uchun har bir satrning o'zi yoki boshqa satrlar bilan kombinatsiyasini xosil qilishingiz mumkin. So'ngra har bir satr predikat yordamida baxolanadi. Bu turdagi jamlash boshqa turdagi jamlashdan farq qilmaydi, farqi ikki jadval bir xildir. Jadvallarni jamlashda qaytariluvchi ustun nomlari oldiga jadval nomi qo'yiladi. Bu ustunlarga so'rovlarda murojaat qilish uchun har xil nomlarga ega bo'lishi kerak. Buning uchun vaqtinchalik nomlar Ya'ni psevdonimlar qo'llanadi. Ular so'rov FROM jumlasida jadval nomidan so'ng bo'shlik qo'yib yoziladi.

Foydalanuvchilar. SQL muxitida har bir foydalanuvchi maxsus identifikatsion nom, murojlat identifikatoriga (ID) ega. Ma'lumotlar bazasiga yuborilgan komanda ma'lum foydalanuvchi bilan yoki boshqacha aytganda maxsus murojaat identifikatori bilan bog'lanadi. SQL ma'lumotlar bazasida ID ruxsat – bu foydalanuvchi nomi va SQL komanda bilan bog'langan murojaat identifikatoriga ilova qiluvchi maxsus kalit so'z USER dan foydalanishi mumkin.

Registratsiya bu kompyuter tizimiga kirish huquqini olish uchun foydalanuvchi bajarishi kerak bo'lgan protseduradir. Bu protsedura foydalanuvchi

bilan qaysi murojaat ID si bog'lanishini aniqlaydi. Odatda har bir ma'lumotlar bazasidan foydalanuvchi o'zining ID siga ega bo'lishi kerak va registratsiya jarayonida haqiqiy foydalanuvchiga aylanadi. Lekin ko'p masalalarga ega foydalanuvchilar bir necha murojaat ID lari bilan registratsiyadan o'tishlari, yoki bir necha foydalanuvchi bitta murojaat ID sidan foydalanishlari mumkin.

Imtiyozlar. Har bir foydalanuvchi SQL ma'lumotlar bazasida nima qilish mumkinligini ko'rsatuvchi imtiyozlarga egadir. Bu imtiyozlar vaqt o'tishi bilan o'zgarishi ya'ni eskilari o'chirilib yangilari qo'shilishi mumkin. SQL imtiyozlari bu ob'ekt imtiyozlaridir. Bu shuni bildiradiki foydalanuvchi berilgan komandani ma'lumotlar bazasining biror ob'ekti ustida bajarishi mumkin. Ob'ekt imtiyozlari bir vaqtning o'zida foydalanuvchilar va jadvallar bilan bog'liq. Ya'ni imtiyoz ma'lum foydalanuvchiga ko'rsatilgan jadvalda, asos jadvalda yoki tasavvurda beriladi. Ixtiyoriy turdagi jadvalni yaratgan foydalanuvchi shu jadval egasidir. Bu shuni bildiradiki foydalanuvchi bu jadvalda hamma imtiyozlarga ega va imtiyozlarini shu jadvalning boshqa foydalanuvchilariga uzatishi mumkin.

2.2. Yo'llanma dasturning algoritimini yaratish

Ushbu yo'llanma dasturining algoritmini C# dasturlash tili orqali bayon qilamiz. Dasturda foydalanilgan asosiy kutubxonalar ro'yxati:

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Windows.Forms;
```

Qidiruv oynasi bilan ishlash uchun yozilgan algoritm:

```
find f00 = new find();  
f00.ShowDialog();  
if (f00.FindText != "")  
{ dataTable1BindingSource2.Filter = String.Format("tashkilot like '%{0}%",  
f00.FindText);
```

```

funkka(); }
else
MessageBox.Show("Qidirish bekor qilindi.", "xabar", MessageBoxButtons.
OK, MessageBoxIcon.Information); }

```

Bu algoritmdagi funkka() funksiyasining algoritmi quyidagicha:

```

private void funkka()
{
if (dataGridView1.RowCount <= 0)
{
MessageBox.Show("Qidirish - bo`yicha qidirish bajarildi, ammo
ma`lumot topilmadi ? ", "Info...", MessageBoxButtons.OK,
MessageBoxIcon.Information);
dataTable1BindingSource2.RemoveFilter();}
}

```

Dasturning asosiy interfeysi yuklangandagi C# tilida yozilgan algoritm:

```

private void Form1_Load(object sender, EventArgs e)
{
this.asosiyTableAdapter.Fill(this.databaseYullanmaDataSet.asosiy);
this.dataTable1TableAdapter.Fill(this.databaseYullanmaDataSet.DataTable1
);
Form2 f2 = new Form2();
f2.ShowDialog();
if (f2.test)
{ }
else { this.Close();
}
}

```

Add formani chaqirish algoritmi:

```

private void toolStripButton6_Click(object sender, EventArgs e)
{
add add1 = new add();
add1.ShowDialog();
this.dataTable1TableAdapter.Fill(this.databaseYullanmaDataSet2.DataTable
}

```

Datagrid da ma`lumotlarni to`liq yuklash algoritmi :

```

private void toolStripButton9_Click(object sender, EventArgs e)
{
this.dataTable1TableAdapter.Fill(this.databaseYullanmaDataSet2.DataTable
1);
}

```

Formani yopish algoritmi:

```
private void toolStripButton12_Click(object sender, EventArgs e)
{
    this.Close();
}
```

Ma'lumotlarni o'chirish algoritmi:

```
private void toolStripButton7_Click(object sender, EventArgs e)
{
    int count = int.Parse(dataGridView1.SelectedRows[0].Cells[10].Value.
ToString());
    add updateform = new add();
    updateform.editor = false;
    updateform.id = count.ToString();
    updateform.data1 = dataGridView1.SelectedRows[0].Cells[1].Value.ToString();
    updateform.data2 = dataGridView1.SelectedRows[0].Cells[2].Value.ToString();
    updateform.data3 = dataGridView1.SelectedRows[0].Cells[3].Value.ToString();
    updateform.data4 = dataGridView1.SelectedRows[0].Cells[4].Value.ToString();
    updateform.data5 = dataGridView1.SelectedRows[0].Cells[5].Value.ToString();
    updateform.data6 = dataGridView1.SelectedRows[0].Cells[6].Value.ToString();
    updateform.data7 = dataGridView1.SelectedRows[0].Cells[7].Value.ToString();
    updateform.data8 = dataGridView1.SelectedRows[0].Cells[8].Value.ToString();
    updateform.ShowDialog();
    this.asosiyTableAdapter.Fill(this.databaseYullanmaDataSet.asosiy);
    this.dataTable1TableAdapter.Fill(this.databaseYullanmaDataSet2.DataTable
1);
}
```

Yozuvlarning birinchisiga kelish algoritmi:

```
private void toolStripButton1_Click(object sender, EventArgs e)
{
    int SelectedRowIndex = dataGridView1.SelectedRows[0].Index;
    if (dataGridView1.SelectedRows.Count > -1)
    {
        if (SelectedRowIndex != 0)
            dataGridView1.CurrentCell = dataGridView1.Rows[0].Cells[0];
        else
            MessageBox.Show("Birinchi yozudasiz !!!", "info", MessageBoxButtons.
OK, MessageBoxIcon.Question);
    }
}
```

Yozuvlarning oxirgisiga kelish algoritmi:

```

private void toolStripButton4_Click(object sender, EventArgs e)
{
    int SelectedRowIndex = dataGridView1.SelectedRows[0].Index;
    if (dataGridView1.SelectedRows.Count > -1)
    {
        if (SelectedRowIndex == dataGridView1.Rows.Count - 1)
            MessageBox.Show("Oxirgi yozudasiz !!!", "info", MessageBoxButtons.OK,
                MessageBoxIcon.Question);
        else
            dataGridView1.CurrentCell = dataGridView1.Rows[SelectedRowIndex +
1].Cells[0]; } }

```

Hisobot formasiga murojat qilish algoritmi :

```

private void toolStripButton10_Click(object sender, EventArgs e)
{
    int count = int.Parse(dataGridView1.SelectedRows[0].Cells[10].Value.
ToString());
    hisobot h1 = new hisobot();
    h1.ID = count;
    h1.report = 0;
    h1.ShowDialog();
}

```

Parametrli hisobotlar ni chaqirish algoritmi:

```

private void yilToolStripMenuItem1_Click(object sender, EventArgs e)
{
    Form7 f7 = new Form7();
    f7.index = 1;
    f7.ShowDialog();
    if (f7.str != "")
    {
        hisobot h2 = new hisobot();
        h2.resultText = f7.str;
        h2.report = 1;
        h2.ShowDialog();
    }
    else
        MessageBox.Show("Hisobot tayyorlash bekor qilindi.", "xabar",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
}

```

Parametrli qidirish oynasini chaqirish algoritmi:

```
private void romgohToolStripMenuItem1_Click(object sender, EventArgs e)
{
    find f01 = new find();
    f01.ShowDialog();
    dataTable1BindingSource2.Filter = String.Format("nomi like '%{0}%",
    f01.FindText);
    funkka();
}
```

Ma`lumotlarni Id bo`yicha tartiblash algoritmi:

```
private void button1_Click(object sender, EventArgs e)
{
    if (comboBox1.SelectedIndex > -1)
    {
        int id = int.Parse(comboBox1.SelectedValue.ToString());
        string pass = usersTableAdapter.ScalarQuery(id);
        if (textBox1.Text == pass)
        {
            this.Close();
            test = true;
        }
        else
        {
            MessageBox.Show("Kalit so`z xato ???.", "Xabar",
            MessageBoxButtons.OK,
            MessageBoxIcon.Information);
        }
    }
}
```

Ma`lumotlar qo`shish algoritmi:

```
private void button1_Click(object sender, EventArgs e)
{
    if ((textBox1.Text == "") || (textBox2.Text == ""))
    {
        MessageBox.Show("Kerakli ma`lumotni yozing !!!.", "Xabar",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        usersTableAdapter.Insertlogin(textBox1.Text, textBox2.Text);
        this.Close();
    }
}
```

```

MessageBox.Show("Amal bajarildi.", "Xabar", MessageBoxButtons.OK,
MessageBoxIcon.Information);
}
}

```

Ma'lumatlarni yangilash algoritmi:

```

private void button1_Click(object sender, EventArgs e)
{
if (comboBox1.SelectedIndex > -1)
{
int id = int.Parse(comboBox1.SelectedValue.ToString());
usersTableAdapter.UpdateQuery(comboBox1.Text, textBox2.Text, id);
this.Close();
MessageBox.Show("Amal bajarildi..", "Xabar", MessageBoxButtons.OK,
MessageBoxIcon.Information);
}
this.usersTableAdapter.Fill(this.databaseYullanmaDataSet.users);
}

```

Malumotlarni o'chirish algoritmi:

```

private void button1_Click(object sender, EventArgs e)
{
if (comboBox1.SelectedIndex > -1)
{
usersTableAdapter.DeleteQuery(int.Parse(comboBox1.SelectedValue.ToStri
ng()));
this.Close();
MessageBox.Show("Amal bajarildi..", "Xabar", MessageBoxButtons.OK,
MessageBoxIcon.Information);
}
this.usersTableAdapter.Fill(this.databaseYullanmaDataSet.users);
}

```

Parametrlarni qidirish formasiga berish algoritmi:

```

private void button1_Click(object sender, EventArgs e)
{
if (index == 1) { str = comboBox1.Text; }
if (index == 2) { str = comboBox1.Text; }
if (index == 3) { str = comboBox1.SelectedValue.ToString(); }
this.Close();
}

```

Yangi oromgoh qo'shish algoritmi:

```

private void button1_Click(object sender, EventArgs e)
{
    if ((textBox1.Text == "") || (textBox2.Text == ""))
    {
        MessageBox.Show("Kerakli ma`lumotni yozing !!!.", "Xabar",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else {
        oromgohTableAdapter.InsertQuery(textBox1.Text, textBox2.Text);
        this.Close();
        MessageBox.Show("Amal bajarildi.", "Xabar", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
    this.oromgohTableAdapter.Fill(this.databaseYullanmaDataSet.oromgoh);
}

```

Oromgohlarni tanlash va yangilash algoritmi:

```

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    int id = int.Parse(comboBox1.SelectedValue.ToString());
    textBox1.Text = oromgohTableAdapter.ScalarQuery1(id);
}

private void button2_Click(object sender, EventArgs e)
{
    if (comboBox1.SelectedIndex > -1) {
        int id = int.Parse(comboBox1.SelectedValue.ToString());
        oromgohTableAdapter.UpdateQuery(comboBox1.Text, textBox1.Text, id);
        this.Close();
        MessageBox.Show("Amal bajarildi.", "Xabar", MessageBoxButtons.OK,
        MessageBoxIcon.Information);
    }
    this.oromgohTableAdapter.Fill(this.databaseYullanmaDataSet.oromgoh);
}

```

Oromgohlarni o`chirish algoritmi:

```

private void button1_Click(object sender, EventArgs e)
{
    int id = int.Parse(comboBox1.SelectedValue.ToString());
    oromgohTableAdapter.DeleteQuery(id);
    this.Close();
    MessageBox.Show("Amal bajarildi.", "Xabar", MessageBoxButtons.OK,

```

```

MessageBoxIcon.Information);
this.orumgohTableAdapter.Fill(this.databaseYullanmaDataSet.orumgoh);
}

```

Ma`lumotlarni MBsiga qo`shish algoritmi:

```

private void button1_Click(object sender, EventArgs e)
{
    if (editor)
    {
        if (comboBox1.SelectedIndex > -1)
        {
            if ((textBox1.Text == "") || (textBox2.Text == "") || (textBox3.Text == "") ||
                (textBox4.Text == "") || (textBox5.Text == ""))
            {
                MessageBox.Show("Kerakli ma`lumotni yozing !!!.", "Xabar",
                    MessageBoxButtons.OK,
                    MessageBoxIcon.Error);
            }
            else
            {
                asosiyTableAdapter.InsertQuery(int.Parse(comboBox1.SelectedValue.ToString()),
                    textBox1.Text, textBox2.Text, textBox3.Text, textBox5.Text,
                    textBox4.Text, int.Parse(comboBox3.Text), int.Parse(comboBox2.Text));
                this.Close();
                MessageBox.Show("Amal bajarildi.", "Xabar", MessageBoxButtons.OK,
                    MessageBoxIcon.Information);
            }
        }
        else
        {
            if (comboBox1.SelectedIndex > -1)
            {
                if ((textBox1.Text == "") || (textBox2.Text == "") || (textBox3.Text == "") ||
                    (textBox4.Text == "") || (textBox5.Text == ""))
                {
                    MessageBox.Show("Kerakli ma`lumotni yozing !!!.", "Xabar",
                        MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
                else
                {
                    asosiyTableAdapter.UpdateQuery(int.Parse(comboBox1.SelectedValue.ToString()),
                        textBox1.Text, textBox2.Text, textBox3.Text, textBox5.Text, textBox4
                            .Text, int.Parse(comboBox3.Text), int.Parse(comboBox2.Text), int.Parse(id));
                }
            }
        }
    }
}

```



```

this.Close();
MessageBox.Show("Amal bajarildi.", "Xabar", MessageBoxButtons.OK,
MessageBoxIcon.Information);
}
}
}
this.oringohTableAdapter.Fill(this.databaseYullanmaDataSet.oringoh);
this.asosiyTableAdapter.Fill(this.databaseYullanmaDataSet.asosiy);
}
}

```

Dasturning MB bilan bog`lanish algoritmi:

```

<connectionStrings>
<add
name="WindowsFormYullanma.Properties.Settings.DatabaseYullanmaCon
nectionString"
connectionString="Data Source=|DataDirectory|\DatabaseYullanma.sdf"
providerName="Microsoft.SqlServerCe.Client.3.5" />
</connectionStrings>

```

Hisobotni tayyorlash algoritmining yozilishi:

```

DataSources>
<DataSource Name="DatabaseYullanmaDataSet">
<ConnectionProperties>
<DataProvider>System.Data.DataSet</DataProvider>
<ConnectionString>/* Local Connection */</ConnectionString>
</ConnectionProperties>
<rd:DataSourceID>53dd0294-c52f-48db-8e49-637f53d2b6b9</rd:DataSourceID>
</DataSource>
</DataSources>
<DataSets>
<DataSet Name="DataSet1">
<Fields>
<Field Name="Expr1">
<DataField>Expr1</DataField>
<rd:TypeName>System.Int32</rd:TypeName>
</Field>
<Field Name="tashkilot">
<DataField>tashkilot</DataField>
<rd:TypeName>System.String</rd:TypeName>
</Field>

```

```

<Field Name="tashtel">
<DataField>tashtel</DataField>
<rd:TypeName>System.String</rd:TypeName>
</Field>
<Field Name="tashmanzil">
<DataField>tashmanzil</DataField>
<rd:TypeName>System.String</rd:TypeName>
</Field>
<Field Name="xodim">
<DataField>xodim</DataField>
<rd:TypeName>System.String</rd:TypeName>
</Field>
<Field Name="xodimtel">
<DataField>xodimtel</DataField>
<rd:TypeName>System.String</rd:TypeName>
</Field>
<Field Name="yil">
<DataField>yil</DataField>
<rd:TypeName>System.Int32</rd:TypeName>
</Field>
<Field Name="muddat">
<DataField>muddat</DataField>
<rd:TypeName>System.Int32</rd:TypeName>
</Field>
<Field Name="nomi">
<DataField>nomi</DataField>
<rd:TypeName>System.String</rd:TypeName>
</Field>
<Field Name="manzili">
<DataField>manzili</DataField>
<rd:TypeName>System.String</rd:TypeName>
</Field>
<Field Name="id">
<DataField>id</DataField>
<rd:TypeName>System.Int32</rd:TypeName>
</Field>
</Fields>
<Query>
<DataSourceName>DatabaseYullanmaDataSet</DataSourceName>

```

```

<CommandText>/* Local Query */</CommandText>
</Query>
<rd:DataSetInfo>
<rd:DataSetName>DatabaseYullanmaDataSet</rd:DataSetName>
<rd:SchemaPath>y:\Visual Studio
2010\Projects\WindowsFormYullanma\WindowsFormYullanma\DatabaseYullanm
aDataSet.xsd</rd:SchemaPath>
<rd:TableName>DataTable1</rd:TableName>
<rd:TableAdapterFillMethod>Fill</rd:TableAdapterFillMethod>
<rd:TableAdapterGetDataMethod>GetData</rd:TableAdapterGetDataMethod>
<rd:TableAdapterName>DataTable1TableAdapter</rd:TableAdapterName>
</rd:DataSetInfo>
</DataSet>
</DataSets>
<Body>
<ReportItems>
<Tablix Name="Tablix2">
<TablixBody>
<TablixColumns>
<TablixColumn>
<Width>0.98425in</Width>

```

MB bilan ishlashda amalga oshirilgan so`rovlarning yozilishi:

```

Name="DatabaseYullanmaConnectionString (Settings)"
ParameterPrefix="@
PropertyReference="ApplicationSettings.WindowsFormYullanma.Properties.Settings.GlobalReference.Default.DatabaseYullanmaConnectionString"
Provider="Microsoft.SqlServerCe.Client.3.5" />
DataAccessorName="usersTableAdapter"
GeneratorDataComponentClassName="usersTableAdapter" Name="users"

```

O`chirich amali:

```

<DeleteCommand>
<DbCommand CommandType="Text" ModifiedByUser="false">
<CommandText>DELETE FROM [users] WHERE ([id] =
@p1))</CommandText>
<Parameters>
<Parameter AllowDBNull="false" AutogeneratedName=""
DataSourceName="" DbType="Int32" Direction="Input"
ParameterName="@p1" Precision="0" ProviderType="Int" Scale="0"

```

```
Size="0" SourceColumn="id" SourceColumnNullMapping="false"
SourceVersion="Original" />
</Parameters>
</DbCommand>
</DeleteCommand>
```

Qo`shish amali:

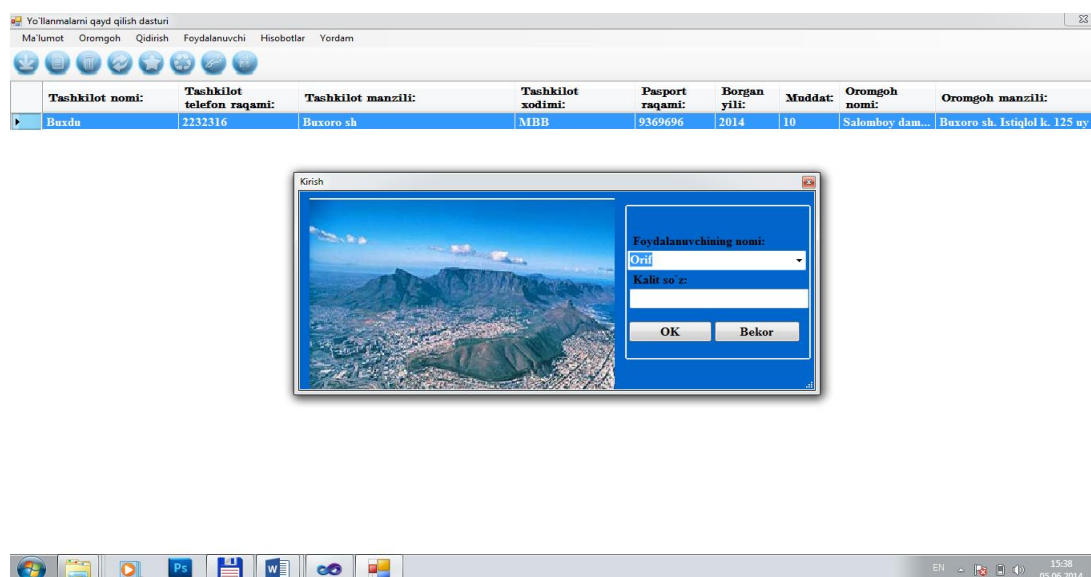
```
<InsertCommand>
<DbCommand CommandType="Text" ModifiedByUser="false">
<CommandText>INSERT INTO [users] ([login], [pass]) VALUES (@p1,
@p2)</CommandText>
<Parameter AllowDBNull="true" AutogeneratedName="" DataSourceName=""
DbType="String" Direction="Input" ParameterName="@p1" Precision="0"
ProviderType="NVarChar" Scale="0" Size="0" SourceColumn="login"
</Parameters>
</DbCommand>
</InsertCommand>
```

Tanlash amali:

```
<SelectCommand>
<DbCommand CommandType="Text" ModifiedByUser="true">
<CommandText>SELECT      users.*
FROM users</CommandText>
<Parameters />
</DbCommand>
```

2.3. Yaratilgan dasturdan foydalanish

Yo'llanma dasturini yuklangandan so'ng quyidagi oyna ochiladi:



2.3.1-chizma. Yo'llanma dasturining asosiy oynasi.

Bu dasturni himoyalash maqsadida foydalanuvchi tomonidan foydalanuvchi nomi va “Kalit soʻz” qoʻyish imkoniyatiga ega. Kalit soʻz kiritilgandan soʻng dasturning asosiy oynasi ochiladi.

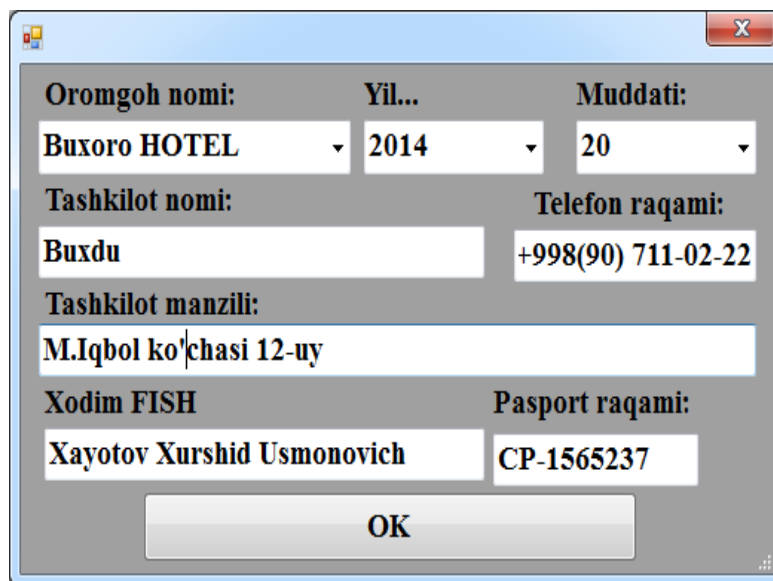


A screenshot of a software window titled "Yangi ma'lumotga kiritish" (Enter new information). The window contains several input fields for user data. The fields are arranged in a grid-like fashion. At the bottom, there is an "OK" button.

Oromgoh nomi:	Yil...	Muddati:
Tashkilot nomi:	Telefon raqami:	
Tashkilot manzili:		
Xodim FISH	Pasport raqami:	
OK		

2.3.2-chizma. Yangi ma'lumotga kiritish.

Ma'lumot yangi kiritishda oromgoh nomi tanladi, qaysi yilda oromgohga borganligi ko'rsatildi, qancha muddat oromgohga yo'llanma berilganligi, Tashkilot nomi ko'rsatiladi, telefon raqam, tashkilot manzili, Xodim FISH va pasport raqami kerak bo'ladi. Bu menyuga Ctrl+ 1 tugmasi orqali ham tez qulay o'tish imkononi beradi.



A screenshot of the same software window as in the previous image, but now with data entered into the fields. The data is as follows:

Oromgoh nomi:	Yil...	Muddati:
Buxoro HOTEL	2014	20
Tashkilot nomi:	Telefon raqami:	
Buxdu	+998(90) 711-02-22	
Tashkilot manzili:		
M.Iqbol ko'chasi 12-uy		
Xodim FISH	Pasport raqami:	
Xayotov Xurshid Usmonovich	CP-1565237	
OK		

2.3.3-chizma. Ma'lumotlar oynasidaga kiritish.

Ma'lumotga yangi ishchi xodimga tegishli bo'lgan ma'lumotlarni kiritishdan iborat bo'lib (Oromgoh nomi, yil, muddat, tashkilot nomi, telefon raqami, tashkilot manzili, xodim FISH, pasport raqami) ma'lumotlar kiritilishi kerak bo'ladi. Yuqorida keltirilgan matnli (shriftli) maydonda to'ldirilmagan maydon bo'lsa u holda "Kerakli ma'lumotni yozing" deb ogohlantiradi va shu oynani to'ldirilgandan so'ng "OK" tugmasini tanlashdan keyin "Amaliyot bajarildi" deb ma'lumotlar bazasiga kiritiladi.

The screenshot shows a window titled "Ma'lumotlarni tahrirlash" with a close button (X) in the top right corner. The form contains the following fields:

- Oromgoh nomi:** Buxoro HOTEL (dropdown menu)
- Yil...** 2014 (dropdown menu)
- Muddati:** 20 (dropdown menu)
- Tashkilot nomi:** Buxdu
- Telefon raqami:** +998(90) 711-02-22
- Tashkilot manzili:** M.Iqbol kochasi 12-uy
- Xodim FISH:** Xayotov Xurshid Usmonovich
- Pasport raqami:** CP-1565237

At the bottom of the form is an "OK" button.

2.3.4-chizma. Ma'lumotlarni tahrirlash

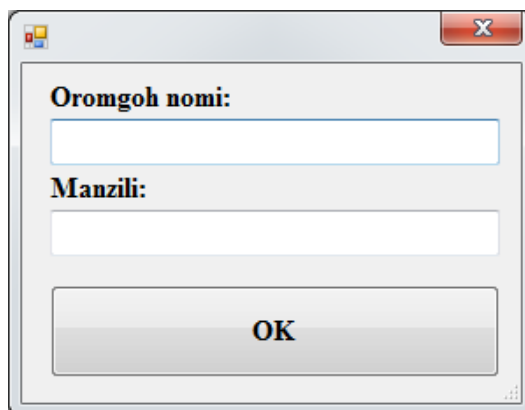
Ma'lumotlarni tahrirlash bu menyuda ishchi xodimning ma'lumotlari (Oromgoh nomi, yil, mud-dat, Tashkilot nomi, telefon raqami, tashkilot manzili, Xodim FISH, pasport raqa-mi) noto'g'ri kiritilsa yoki orfografik xatolarga yo'l qo'yilgan bo'lsa bu menyudan to'g'irlash imkoniyatiga ega bo'ladi. Bu menyuga Ctrl+2 tugmasi orqali ham tez qulay o'tish imkononi beradi.

The screenshot shows a window titled "Ma'lumotlarni ko'rish" with a close button (X) in the top right corner. It displays a table with the following data:

Tashkilot nomi:	Telefon raqami:	Manzili:	Xodim FISH	Pasport raqami:	Borgan yili:	Muddat	Oromgoh nomi:	Manzili:
Buxdu	+998(90) 711-02-22	M. Iqbol kochasi 12-uy	Xayotov Xurshid Usmonovich	CP-1565237	2014	20	Buxoro HOTEL	buxoro viloyati Buxoro shahar 466-uy

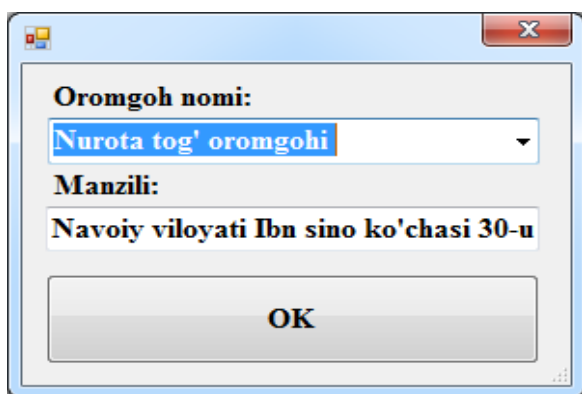
2.3.5-chizma. Ma'lumotlarni ko'rish oynasi.

Ma'lumotlarni ko'rish menyusida nechta ishchi xodimga yo'llanma berilgan bo'lsa ularning ro'yxatini ko'rish imkoniyatiga ega bo'lamiz. Bundan tashqari ro'yxatdagi ishchi xodimlarni kerak bo'lsa kerakli Word, PDF, Excel dasturlari yordamida olib qo'yish ham mumkin. Bu menyuda yana ishchi xodimlar ro'yxatini chop qilish imkoniyati bor. Bu menyuga Ctrl+3 tugmasi orqali ham tez qulay o'tish imkononi beradi.



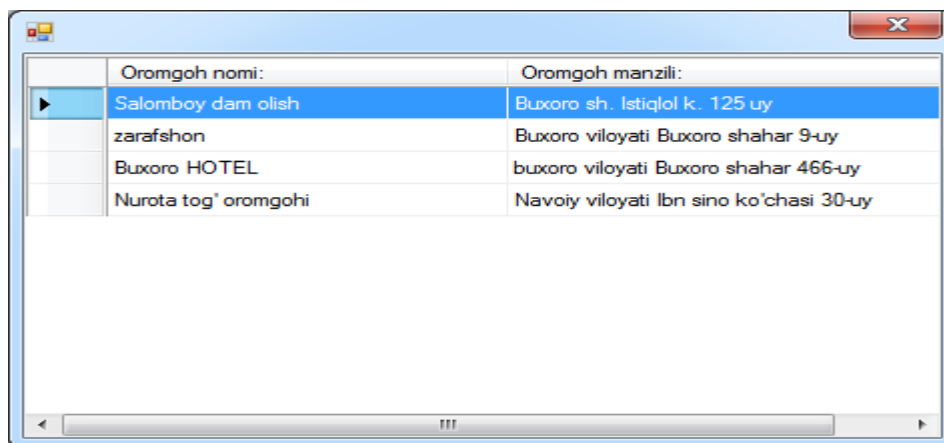
2.3.6–chizma oromgoh kiritish oynasi.

Yangi oromgoh kiritish uchun bizga oromgoh nomi va manzili bo'lsa maqsadga muvofiq bo'ladi. Kiritilgan oromgoh nomi va manzili yangi ma'lumot kiritishdagi oromgoh nomida paydo bo'ladi. Bu menyuga Alt+1 tugmasi orqali ham tez qulay o'tish imkononi beradi.



2.3.7 –chizma.Oromgohni tahrirlash oynasi.

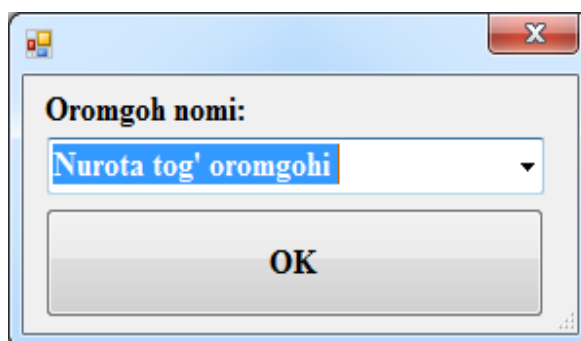
Bu menyuda oromgoh nomi yoki manzili kiritishda noto'g'ri yoki orfografik xatolarga yo'l qo'yilgan bo'lsa bu menyuda xatolarni to'g'irlash imkoniyati mavjud. Bu menyuga Alt+2 tugmasi orqali ham tez qulay o'tish imkononi beradi.



Oromgoh nomi:	Oromgoh manzili:
Salomboy dam olish	Buxoro sh. Istiqlol k. 125 uy
zarafshon	Buxoro viloyati Buxoro shahar 9-uy
Buxoro HOTEL	buxoro viloyati Buxoro shahar 466-uy
Nurota tog' oromgohi	Navoiy viloyati Ibn sino ko'chasi 30-uy

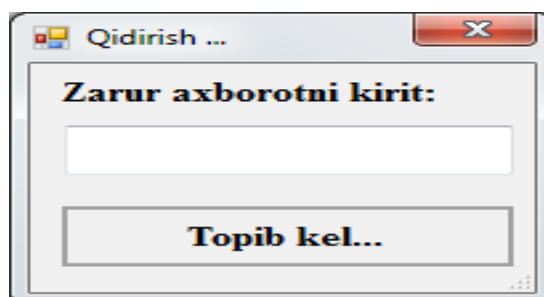
2.3.8 –chizma Oromgohlarni ro'yxatini ko'rish

Oromgoh menyusidagi ko'rish tugmasini bosish orqali yangi oromgoh nomi va manzili kiritilganini ko'rish imkoniyatiga ega bo'lamiz . Bu menyuga Alt+3 tugmasi orqali ham tez qulay o'tish imkononi beradi.



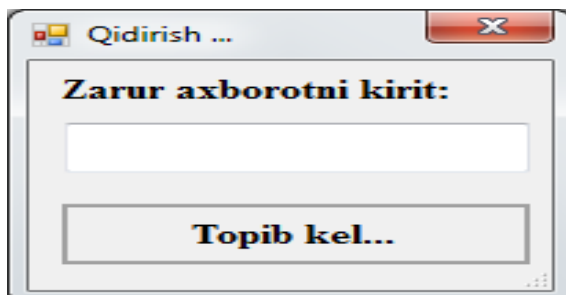
2.3.9-chizma Oromgohni o'chirish oynasi.

Oromgohni o'chirish menyusi bunda oromgoh nomi noto'g'ri kiritilgan bo'lsa oromgoh tanladi va “OK” tugmasini ustida sichqonchanning chap tugmasini bosish orqali bu oromgoh ro'yxatdan o'chiriladi. Bu menyuga Alt+4 tugmasi orqali ham tez qulay o'tish imkononi beradi.



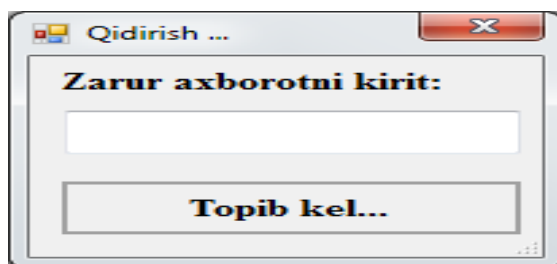
2.3.10- chizma Tashkilotlarni qidirish

Har bir ishni bajarishda vaqtning tejalişini inobatga oladilar. Biz ham bu menyumizda tashkilotlar ro'yxati juda ko'p bo'lgani uchun kerak bo'ladigan tashkilotni Alt+Shift+T tugmasi bilan bir zumda qidirib topishimiz mumkin.



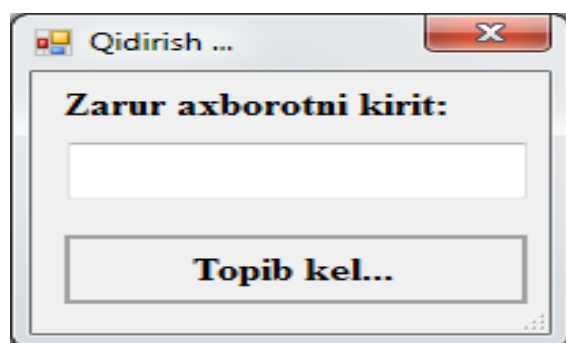
2.3.11-chizma.Oromgohni qidirish oynasi.

Bu menyumizda oromgohlar ro'yxati juda ko'p bo'lgani uchun kerak bo'ladigan oromgohlarni Alt+Shift+O tugmasi bilan bir zumda qidirib topishimiz mumkin.



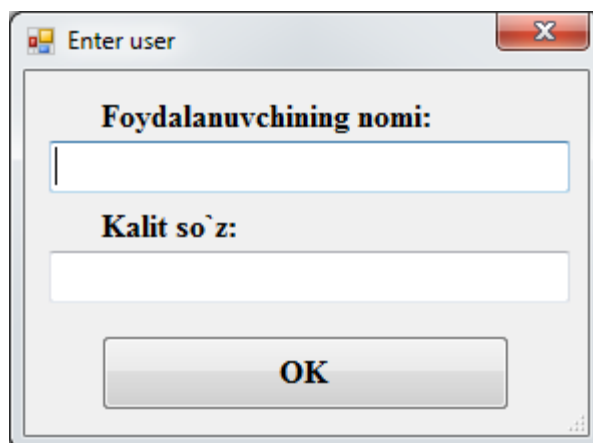
2.3.12-chizma F.I.SH bo'yicha qidirish

Yuqorida ko'rsatilgan bo'limda ishchi xodimlar F.I.SH qidirib topiladi.Bu ishchi xodimlarning qidiruv jarayonini ancha yengillashtiradi va dastur foydalanuvchisi istalgan biror ishchi xodimning oromgohda dam olish jarayonidagi bazaga kiritilgan ma'lumotlarini Alt+Shift+F tugmasi orqali ham osongina topish imkoniyati bor.



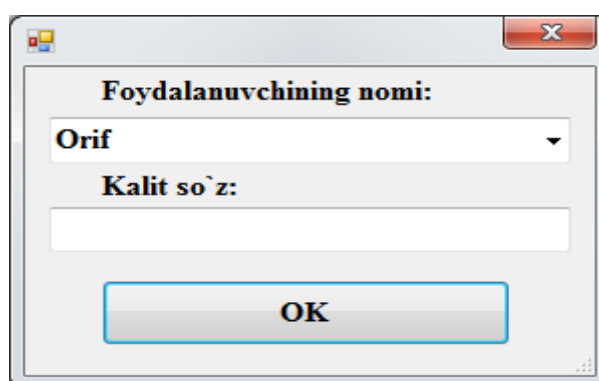
2.3.13-chizma.Muddat bo'yicha qidirish

Bu dasturda ishchi xodimlarning qidiruv jarayonini ancha yengillashtiradi va dastur foydalanuvchisi istalgan oromgohda dam olgan ishchi xodimning muddati bo'yicha qidirib topib beradi. Bu menyuga Alt+Shift+M tugmasi orqali ham tez qulay o'tish imkonini beradi.



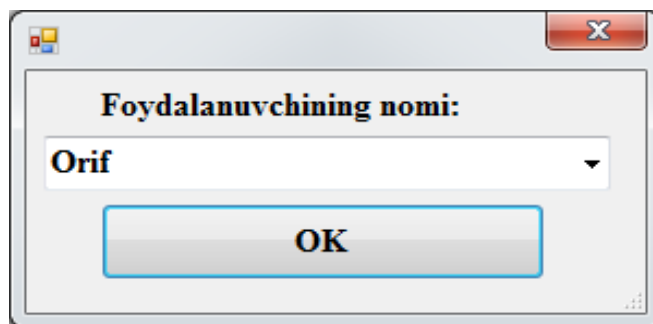
2.3.14-chizma Yangi foydalanuvchini kiritish

Bu menyudan foydalanishda ishchi xodimlarni ro'yxatga oluvchi xodim o'zi dasturni himoyalash maqsadida foydalanuvchi nomi va kalit so'zni kiritish kerak bo'ladi. Bu foydalanuvchi nomi va kalit so'zni kiritishdan maqsad boshqa foydaluvchi foydalanmaslikdan iborat. Bu menyuga Ctrl+Shift+1 tugmasi orqali ham tez qulay o'tish imkonini beradi.



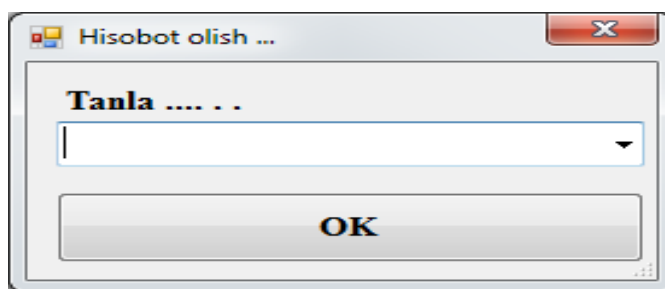
2.3.15-chizma Foydalanuvchi nomini tahrirlash

Bu menyuda foydalanishda ro'yxatga oluvchi xodim o'zining foydalanuvchi nomi va kalit so'zni o'zgartirishi mumkin. Bu menyuga Ctrl+Shift+2 tugmasi orqali ham tez qulay o'tish imkonini beradi.



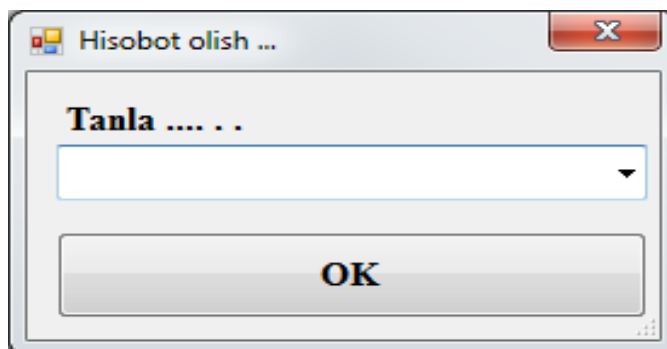
2.3.16-chizma Foydalanuvchi o'chirish

Bu menyuda foydalanuvchi nomi ortiqcha bo'lsa foydalanuvchi menyusidan “o'chirish” tugmasi tanlanib foydalanuvchini ko'rsatamiz va “OK” tugmasini bosish orqali “Amal bajarildi” degandan so'ng ma'lumotlar bazasidan bu foydalanuvchi o'chiriladi. Bu menyuga Ctrl+Shift+4 tugmasi orqali ham tez qulay o'tish imkonini beradi.



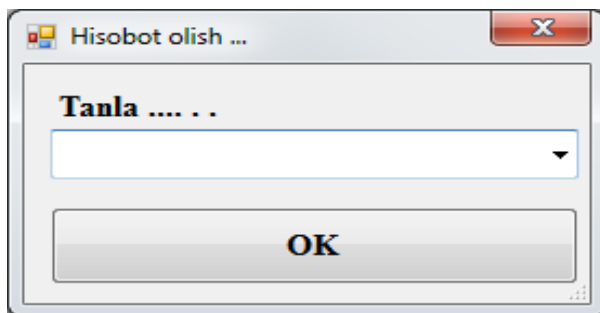
2.3.17-chizma Yil bo'yicha hisobot

Bu menyuda yil bo'yicha hisobot olish mumkin. Juda ko'p ishchi xodim kiritilgan bo'lsa masalan 2014-yilni tanlasak oromgohga shu yili borgan ishchi xodimlarning ro'yxatini chiqarib beradi. Bu menyuga Alt+Y tugmasi orqali ham tez qulay o'tish imkonini beradi.



2.3.18-chizma Muddat bo'yicha hisobot

Bu menyuda muddat bo'yicha hisobot olish mumkin. Ishchi xodimlarning ro'yxatidan masalan 20 ni tanlasak 20-kunlik oromgohga borgan xodimlarning ro'yxatini chiqarib beradi. chiqargan ro'yxatni Word, PDF, Excel ga saqlash imkonini beradi. Bu menyuga Alt+M tugmasi orqali ham tez qulay o'tish imkonini beradi.



2.3.19-chizma Oromgoh bo'yicha hisobot

Bu menyuda oromgoh bo'yicha hisobot olish mumkin. Ishchi xodimlarning ro'yxatidan qaysi oromgohga borganini aniqlamoqchi bo'lsak hisobotlar menyusida "oromgoh bo'yicha" tugmani tanlasak bu oromgohga ishchi xodimlar qachon borgan ro'yxatini chiqarib beradi. Chiqargan ro'yxatni Word, PDF, Excel ga saqlash imkonini beradi. Bu menyuga Alt+O tugmasi orqali ham tez qulay o'tish imkonini beradi.

Bitiruv malakaviy ishining ikkinchi bobi yuqorida ta'kidlab o'tilganidek, "Yo'llanma dasturini yaratish" deb nomlanadi.

Bu bobning o'zi ham uch qismga bo'lingan bo'lib, birinchi qismi "Ma'lumotlar Ba'zasini Boshqarish SQL tizimlari haqida" deb nomlanadi. Bu qismda SQL haqida, SQL tilida ma'lumotlar turlari, jadvallarni o'chirish, Jadvalni yaratilgandan so'ng o'zgartirish, satrlarni o'chirish, jadvalni yaratilgandan so'ng o'zgartirish, foydalanuvchilar, imtiyozlari ko'rsatilgan.

Ikkinchi qismi esa "Yo'llanma dasturning algoritimini yaratish" deb nomlanib unda dasturda foydalanilgan asosiy kutubxonalar ro'yxati, qidiruv oynasi bilan ishlash uchun yozilgan algoritmlar, Add formani chaqirish algoritmi, ma'lumotlarni o'chirish algoritmi, hisobot formasiga murojat qilish algoritmi,

ma'lumotlar qo'shish algoritmi, hisobotni tayyorlash algoritmi va boshqa algoritmlar kiltirilgan.

Bobning uchinchi qismi "Yaratilgan dasturdan foydalanish" deb nomlanadi. Bu bobda yo'llanma dasturining ishlashi haqida bo'lib ishlash jarayonlari ya'ni yangi ma'lumotga kiritish, ma'lumotlarni tahrirlash, ma'lumotlarni ko'rish, oromgoh kiritish oynasi, oromgohni tahrirlash, oromgohlarni ro'yxatini ko'rish, oromgohni o'chirish oynasi, tashkilotlarni qidirish, oromgohni qidirish oynasi, muddat bo'yicha qidirish, yangi foydalanuvchini kiritish, Foydalanuvchini o'chirish, Foydalanuvchi nomini tahrirlash, yil bo'yicha hisobot, muddat bo'yicha hisobot, Oromgoh bo'yicha hisobot oynalari keltirilgan va ishlash yo'nalishilari ko'rsatilgan.

XOTIMA

Ushbu bitiruv malakaviy ishimning mavzusi “Yo’llanmalar nazorati va qayt qilish dasturi ” deb nomlanib, u kirish qismi, ikkita bob, xulosalari, xotima va foydalanilgan adabiyotlardan iborat.

Birinchi bobda asosan “Yo’llanma dasturini yaratishning nazariy asoslari” haqida to’xtalib o’tilgan. Bunda bob ikki qismga bo’lingan bo’lib, birinchi qismda “Microsoft Visual Studio haqida” ma’lumot. Ikkinch qismda “C# dasturlash tili haqida” haqida ma’lumotlar keltirilgan. Birinchi qismda menyular paneli bilan ishlash, yangi loyiha yaratish haqida ma’lumotlar keltirilgan. Ikkinchi qismda C# dasturlash tili haqida, C# da dasturlash tilining strukturalari haqida tavsiflab o’tilgan.

Ikkinchi bob “Yo’llanma dasturini yaratish” deb nomlanib, u ham uch qismga bo’lingan. Birinchi qismi “Ma’lumotlar Ba’zasini Boshqarish SQL tizimlari haqida” deb nomlanadi. Bu qismda SQL haqida, SQL tilida ma’lumotlar turlari, jadvallarni o’chirish, Jadvalni yaratilgandan so’ng o’zgartirish, satrlarni o’chirish, jadvalni yaratilgandan so’ng o’zgartirish, foydalanuvchilar, imtiyozlari ko’rsatilgan.

Ikkinchi qismi esa “Yo’llanma dasturning algoritimini yaratish” deb nomlanib unda dasturda foydalanilgan asosiy kutubxonalar ro’yxati, qidiruv oynasi bilan ishlash uchun yozilgan algoritmlar, Add formani chaqirish algoritmi, ma’lumotlarni o’chirish algoritmi, hisobot formasiga murojat qilish algoritmi, ma’lumotlar qo’shish algoritmi, hisobotni tayyorlash algoritmi va boshqa algoritmlar haqida to’xtalib o’tilgan.

Bobning uchinchi qismida esa dastur menyulari va ularning ishlash prinsiplari haqida ma’lumotlar keltirilgan.

Bitiruv malakaviy ishni bajarishdan asosiy maqsad har bir xodim 3 yilda 1 marta viloyat kasaba uyushmasi tomonidan yo’llanma olishi mumkin. Shunda xatoliklar takror borishlar bo’lmasligi uchun yo’llanmalar nazorat va qayd qilish dasturini yaratish maqsadga muvofiq bo’ladi.

Bitiruv malakaviy ishining hajmi 47 bet, 11 ta adabiyotdan foydalanildi.

Adabiyotlar

1. Karimov I.A. “Yuksak ma’naviyat yengilmas kuch”. –T.:ma’naviyat, 2008-yil. 176 bet.
2. Хомоненко А. Д., Цыганков В. М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений / Под ред. Проф. А.Д. Хомоненко. – 6-е изд. Доп. – СПб.: КОРОНА-Век, 2009. - 736 с.
3. Ицик Бен-Ган Microsoft SQL Server 2008. Основы T-SQL: Пер. с англ. – СПб.: БХВ-Петербург, 2009. – 432 с.: ил.
4. Роберт Виейра - Программирование баз данных MS SQL Server 2005. Базовый курс. : Пер. с англ. – М.: ООО «И.Д. Вильямс», 2007. – 832 с.: ил. – Парал. тит. англ.
5. Ржеуцкая С.Ю. - Базы данных. Язык SQL: учеб. пособие/ С. Ю. Ржеуцкая. – Вологда: ВоГТУ, 2010. – 159 с.
6. Sayfiyev J.F. “SQL tiliga kirish” uslubiy qo’llanma. Buxoro 2005-yil. 76 bet.
7. Молинаро Э. SQL. Сборник рецептов. – Пер. с англ. – СПб: Символ-Плюс, 2009. – 672 с., ил.
8. Фленов М. Е. Библия C#. – 2-е изд. Перераб. И доп.- СПб.: БХВ-Петербург. 2011. – 560 с.: ил. + CD-ROM.
9. Уотсон, Карл и, Нейгел, Кристиан, Педерсен, Якоб Хаммер, Рид, Джон Д., Скиннер, Морган, Уайт, Эрик. Visual C# 2008: базовый курс. : Пер. с англ. - М. : ООО "И.Д. Вильяме", 2009. - 1216 с.: ил. — Парал. тит. англ.
10. Дейтел Х. C# Наиболее полное руководство. Пер. с англ. / Дейтел Х., Дейтел П., Листфилд Дж., Нието Т., Йегер Ш., Златкина М. – СПб.: БХВ-Петербург, 2006. – 1056 с.: ил.
11. www.ziynet.uz