**O’ZBEKISTON AXBOROT TEXNOLOGIYALARI VA TELEKOMUNIKATSIYALARINI RIVOJLANTRISH VAZIRLIGI**

**TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI FARG’ONA FILIALI**

**Kompyuter injineringi fakulteti**

**Dasturiy injineringi kafedrasi**

**C++ da dasturlash fanidan**

KURS ISHI

Bajardi: 611-14 guruh talabasi

Turg’unboyev B

Raxbar: Musayev X

Qabul qildi: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Nukus 2022

**Mavzu:Talabalarning labaratoriya ishini qabul qilish tizmini avtomatlashtirish.**

**Reja:**

**Kirish**

1. **Nazariy qism**
   1. Ma’lumotlar bazasi haqida tushuncha
   2. C++ Builder dasturida ma’lumotlar ombori bilan ishlaydigan komponentalar ro‘yxati.
2. **Loyixa qism**
   1. C++ Builderda Button, Label companentalari.
   2. Dasturda bajarilgan ishlar

**Xulosa**

**Foydalanilgan adabiyotlar**

**Ilova**

## 

Insoniyat tarixining ko‘p asrlik tajribasi ezgu g‘oyalardan va sog‘lom mafkuradan mahrum biron-bir jamiyatning uzoqqa bora olmasligini ko‘rsatdi. Shu bois, mustaqillik tufayli mamlakatimiz o‘z oldiga ozod va obod Vatan, erkin va farovon hayot barpo etish, rivojlangan mamlakatlar qatoridan o‘rin olish, demokratik jamiyat qurish kabi ezgu maqsadlarni qo‘ydi.

Bu esa kelajagimizni yaqqol tasavvur etish, jamiyatimizning ijtimoiy-ma’naviy poydevorini mustahkamlash ehtiyojini tug‘diradi. Demak, galdagi eng asosiy vazifa: yosh avlodni Vatan ravnaqi, yurt tinchligi, xalq farovonligi kabi   
olijanob tuyg‘ular ruhida tarbiyalash, yuksak fazilatlarga ega, ezgu g‘oyalar bilan qurollangan komil insonlarni voyaga etkazish, jahon andozalariga mos, kuchli bilimli, raqobatbardosh kadrlar tayyorlashdir.

«Jahon sivilizatsiyasiga daxldor bo‘lgan eng zamonaviy ilmlarni egallamay turib, mamalakat taraqqiyotini ta’minlash qiyin», - degan edilar prezidentimiz Sh. Mirziyoyev. O‘zbekistonning iqtisodiy va ijtimoiy sohalarda yuqori natijalarga erishishi, jahon iqtisodiy tizimida to‘laqonli sheriklik o‘rnini egallay borishi, inson faoliyatining barcha jabhalarida zamonaviy axborot texnologiyalaridan yuqori darajada foydalanishning ko‘lamlari qanday bo‘lishiga hamda bu texnologiyalar ijtimoiy mehnat samaradorligini oshishida qanday rol o‘ynashiga bog‘liq.

Yuqorida aytilgan mezon va talablarga rioya qilgan xolda Respublikamizda, zamonaviy hisoblash texnikasi vositalaridan samarali foydalanishni uddalay oladigan, zamonaviy kompyuterlardan amaliy ish faoliyatida keng foydalana oladigan etuk kadrlar tayyorlash dolzarb vazifalardan hisoblanadi. Shuning uchun, kadrlar tayyorlash milliy dasturining ikkinchi bosqichida yuqori malaka, raqobatbardosh kadrlar tayyorlash uchun sifatli, jahon andozalariga mos darsliklar, o‘quv qo‘llanmalari va ma’ruza matnlarini tayyorlab, chop ettirish masalasiga juda katta e’tibor berilgan.

Mustaqillikka erishganimizdan so’ng yurtimizning barcha sohalari qatorida axborot texnologiyalari sohasi ham tubdan o’zgarib, rivojlanishga yuz tutdi. Hozirgi kunda yurtimizda biz yoshlarga keng imkoniyatlar eshigi ochilmoqda.

Hozirgi texnika va texnologiyalarning jadallik bilan rivojlanib borayotgan zamonida inson faktoriga bo’lgan talab kundan kunga kamayib bormoqda. Buning asosiy sababi — zamonaviy texnologiyalarning inson og’irini yengillatayotganligida.

Hozirgi kunda hayotimizning barcha jabhalariga texnik va texnologik qurilmalar, avtomatlashgan tizimlar, kompyuter texnologiyalari kirib kelib ulgirdi. Kimdir bu texnika va texnologiyalardan unumli vasamarali oydalanmoqda, unga yanada qo’shimcha yangilik va qo’shimchalar kiritmoqda, aksincha yana kimlardir ularni boricha o’z holatida ishlatmoqda.

XXI asr — “Axborot asri”. Ushbu asrda texnika va texnologiyalar shu darajada rivojlanib kettiki, hatto inson aqli bovar qilmas dajaga yetib ulgirdi.

Hozirgi kunda juda ham ko’plab yordamchi dasturlar ishlab chiqaruvchi kompaniyalar mavjud. Bularning ichida foydalanuvchiga yordam beradigan va bevosita o’zining dastur yaratishiga imkon beruvchi dasturlarni ham ishlab chiqaruvchi kompaniyalar mavjud.

# I. Nazariy qism

# 1.1 Malumotlar bazasi haqida tushuncha

Ma’lumotlar bazasi - bu EHM ning tashqi xotirasida saqlanadigan, har qanday jismoniy, ijtimoiy, statistik, tashkiliy va boshqa obyektlar, jarayonlar, holatlarning o‘zaro bog‘liq va tartiblashtirilgan majmuidir.

Ma’lumotlar bazasi (MB) har xil foydalanuvchilarning axborot yetishmovchiligini ta’minlash uchun mo‘ljallangan. Amaliyotda ko‘pchilik ma’lumotlar bazasi chegaralangan predmet sohasi uchun loyihalashtiriladi. Bitta EHMda bir qancha ma’lumotlar bazasi yaratiladi.Vaqti bilan turdosh vazifalarni bajarishga mo‘ljallangan ba’zi bir ma’lumotlar bazasi birlashishi ham mumkin.

Har qanday axborot tizimi bizni qamragan real dunyoning ba’zi tomonlarini yoki predmet sohasini yoki muammoni yoritishi mumkin.

Muammoli muhit vaqti o‘zgarishi bilan o‘zgarishi, ya’ni ob’yektlar xususiyatining o‘zgarishida, eski ob’yektlarning yo‘qolishi va yangilarining paydo bo‘lishida ifodalanadi. Bu o‘zgarishlar voqealar natijasida bo‘lib o‘tadi. Voqealarning ketma-ketligi jarayonni tashkil etadi.Har qanday axborot tizimi ob’yektlarning o‘zlari bilan emas, mavjudiy borliq singari, balki ularning belgili yoritqichi-idenfikatori bilan ishlaydi.

Belgi - idenfikatorlarning bosh vazifasi - ob’yektni guruhdagi turdosh ob’yektlardan farqlash.Ob’yektning idenfikatori, umumiy aytganda, ob’yektning xususiyati haqida yoki, ya’ni shunga o‘xshash, uning u yoki bu sinfga tegishligi haqida hech qanday axborot bermasligi mumkin.

Ma’lumotlar. Belgili shaklda ifodalangan ob’yekt yoki ob’yektlarning muomalasi haqidagi axborotlar ma’lumotni tashkil qiladi.

Bu ma’lumotlar inson yoki qandaydir texnik qurilma tomonidan qabul qilinishi va tegishli tarzda interpretatsiya qilinishi mumkin. Ma’lumotlarning tavsifli xususiyati shu hisoblanadiki, ya’ni ularni bir belgili tizimdan boshqasiga qayta kodlash axborotni yuqotmagan holda o‘tkazish mumkin.Belgili ifodalanishning bunaqangi xususiyatining ahamiyati - mavjud predmetli holatning qabul qiluvchiga yo‘naltirilgan, belgilarning har xil tizimlarida ifodalanish imkonini beradi.

Ma’lumotlar bazasini qurishda insonga yo‘naltirilgan mantiqiy ifodalanish haqida va uzoq muddatli xotira qurilmasiga yo‘naltirilgan jismoniy ifodalanish haqida gapirish an’anaga aylangan.

Ma’lumotlar ba’zasi - ma’lum masalaga yoki biror faoliyatga taalluqli o’zaro bog’langan va aloxida ko’rinishda tashkil etilgan ma’lumotlar.

MB ma’lumotlarni xsusiyatlari: – tuliklik; – aktuallik; – kulaylik; – Tugri tashkil etish.

Ma’lumotlar banki - bir nechta MB, texnik vosita va dastur ta’minotidan iborat avtomatlashtirilgan tizim.

Malumotlar bazasini (MB) yaratish “ACCESS” dasturi bilan ishlash.

MB tashkil qilish, ularga qo’shimcha ma’lumotlarni kiritish va mavjud MBdan foydalanish uchun maxsus MBlar bilan ishlaydigan programmalar zarur bo’ladi. Bunday programmalar majmui ma’lumotlar bazasini boshqarish sistemalari (MBBT) deb yuritiladi. Aniqroq qilib aytganda, MBBT–bu ko’plab foydalanuvchilar tomonidan MBni yaratish, unga qo’shimcha ma’lumotlarni kiritish va MBni birgalikda ishlatish uchun zarur bo’lgan programmalar majmuidir. MBBTning asosiy tarkibiy qismi–ma’lumotlar bo’lsa, boshqa tarkibiy qismi–foydalanuvchilardir. Bulardan tashqari Hardware- texnik va Software-dasturiy ta’minoti ham MBBTning samarali ishlashini ta’minlovchi tarkibiy qismlar hisoblanadi. Hardware tashqi qo’shimcha qurilmadan iborat bo’lsa, programma qismi esa MB bilan foydalanuvchi o’rtasidagi muloqotni tashkil qilishni amalga oshiradi. MBning tuzilishi o’rganilayotgan ob’ektning ma’lumotlari ko’rinishi, ma’nosi, tuzilishi va hajmiga bog’liq bo’ladi.

Odatda, foydalanuvchilar quyidagi kategoriyalarga bo’linadilar:

* foydalanuvchi-programma tuzuvchi;
* sistemali programma tuzuvchi;
* ma’lumotlar bazasi administratori.

Bunda programma tuzgan foydalanuvchi MBBT uchun yozgan programmasiga javob beradi, sistemali programma tuzuvchi esa butun sistemaning ishlashi uchun javobgar hisoblanadi. U holda MB administratori sistemaning saqlanish holatiga va ishonchliligiga javob beradi.

MBBT quyidagicha tavsiflanadi:

Ispolnimost-Bajarilishlik, foydalanuvchi so’roviga hozirjavoblik bilan muloqotga kirishish;

Minimalnayapovtoryaemost- Minimal takrorlanishlik, MBdagi ma’lumot iloji boricha kam takrorlanishi lozim, aks holda ma’lumotlarni izlash susayadi;

Yaxlitlik –axborotni MBda saqlash iloji boricha ma’lumotlar orasidagi bog’liqlikni asragan holda bo’lgani, ayni muddao;

Bezopasnost–Xavfsizlik, MB ruxsat berilmagan kirishdan ishonchli himoya qilingan bo’lishi lozim. Faqat foydalanuvchi va tegishli tashkilotgina ma’lumotlarga kira olish va foydalanish huquqiga egalik qilishi mumkin;

Migratsiya–ba’zi bir ma’lumotlar foydalanuvchilar tomonidan tez ishlatilib turiladi, boshqalari esa faqat talab asosida ishlatiladi. SHuning uchun ma’lumotlar tashqi xotiralarda joylashtiriladi va uni shunday tashkil qilish kerakki, eng ko’p ishlatiladigan ma’lumotlarga murojaat qilish qulay bo’lsin.

Ma’lumotlar bazasini boshqarish sistemasida har bir MB modeli quyidagi xususiyatlari bo’yicha tavsiflanadi:

* Ma’lumotlar tuzilmalarining turi;
* Ma’lumotlar ustida bajariladigan amallar;
* Butunlikning cheklanganligi.

Bu xususiyatlarni e’tiborga olgan holda ma’lumotlar bazasi modellari quyidagi turlarga bo’linadi:

* Daraxtsimon (ierarxik) modellar;
* Tarmoqli (to’rli) modellar;
* Relyatsion modellar.

Yana shu narsani ta’kidlash lozimki, ma’lumotlar bazasi modellarining faqatgina yuqorida qayd qilingan modeli mavjud deyish noto’g’ri. CHunki bulardan tashqari yana ma’lumotlar bazasining binar munosabatlar modeli, ER – modellari, semantik model kabi boshqa turlari ham mavjud. Lekin amalda asosan dastlabki ta’kidlangan 3 turli modellar ko’proq qo’llanilib kelinmoqda.SHuning uchun ham biz ushbu modellarga qisqacha to’xtalib o’tamiz.

Daraxtsimon (ierarxik) modelda ob’ektlar yozuvlar ko’rinishida ifodalanadi.

Ierarxik modelda ikki yarusdagi elementlar bog’langan bo’lsa, unday ma’lumotlar tarmoqli (to’rli) modelda ifodalangan deyiladi.Tarmoqli modellarda ham ob’ektlar daraxtsimon modellardagi kabi yozuvlar ko’rinishida tasvirlanadi.Ob’ektlarning o’zaro aloqalari yozuvlar o’rtasidagi aloqalar sifatida tavsiflanadi.

Relyatsion modellarda esa ob’ektlar va ularning o’zaro aloqalari ikki o’lchovli jadval ko’rinishida tasvirlanadi. Ma’lumotlarning bunday ko’rinishda tasvirlanishi ob’ektlarning o’zaro aloqalarini yaqqol tasvirlanishiga asos bo’ldi.

MBBT alohida olingan quyidagi modullardan tashkil topadi:

MBni boshqarish bloki-disklardagi ma’lumotlar bilan foydalanuvchi programmasi va sistemaning so’rovi orasidagi interfeysni aniqlaydi;

Fayl menedjeri-ma’lumotlar tuzilmasi bilan disklar o’rtasidagi bog’lanishni boshqaradi;

Guery protsessor-ingliz tilida yozilgan guery gaplarini MBni boshqarish bloki tushunadigan tilga o’tkazadi;

Prekompilyator DML(Data Manipulation Language)- ma’lumotlar bilan manipulyatsiya qiladigan til bo’lib, u quyidagi operatsiyalarga javob beradi:

MBdan ma’lumotlarni ajratib olish;

MBga ma’lumotlarni kiritish;

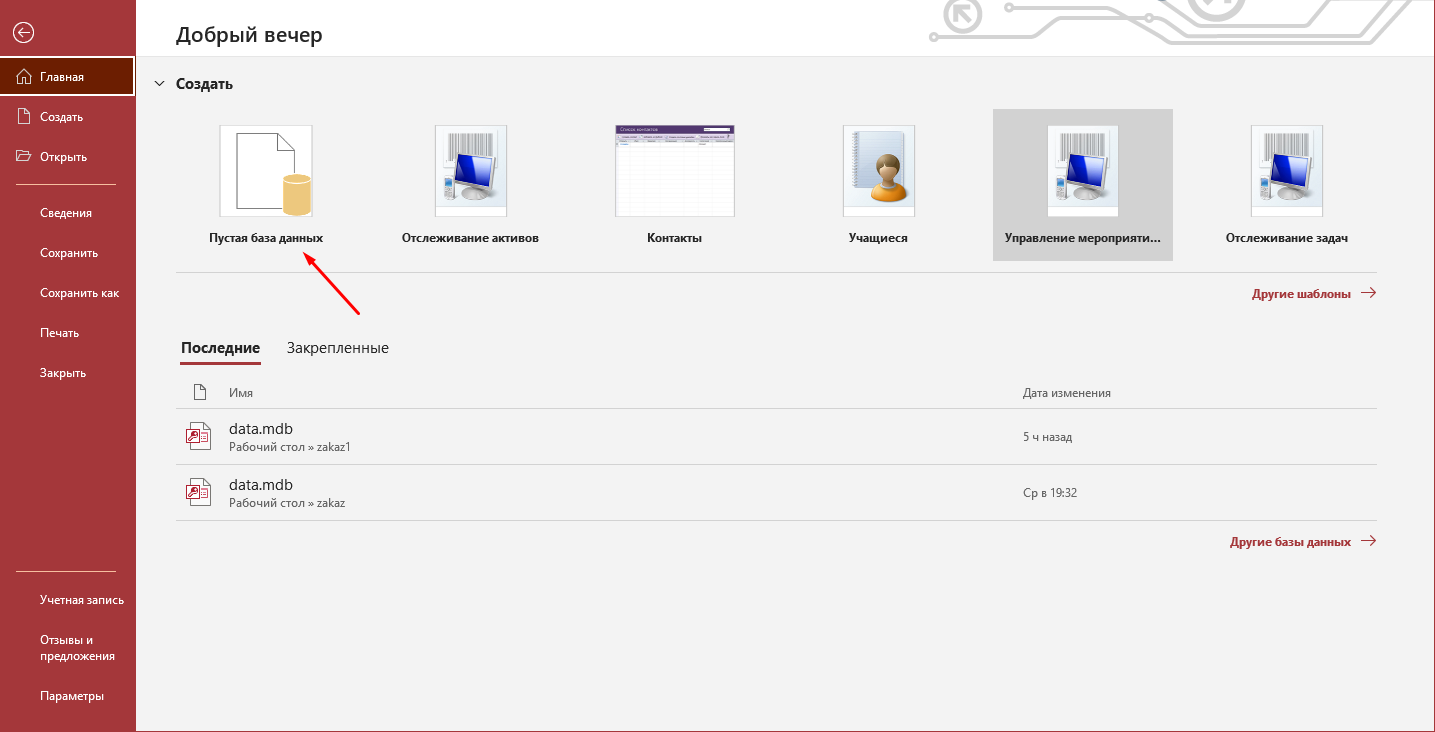
MBdan ma’lumotlarni olib tashlash;

MBni modifikatsiya (o’zgartirishlar) qilish;

Kompilyator DDL(Data Definition Language)-MB tilini, uning tuzilmasini va tashqi xotiralardagi axborot turini aniqlaydi. MBning tuzilmasi ko’pincha jadval shaklida bo’ladi.

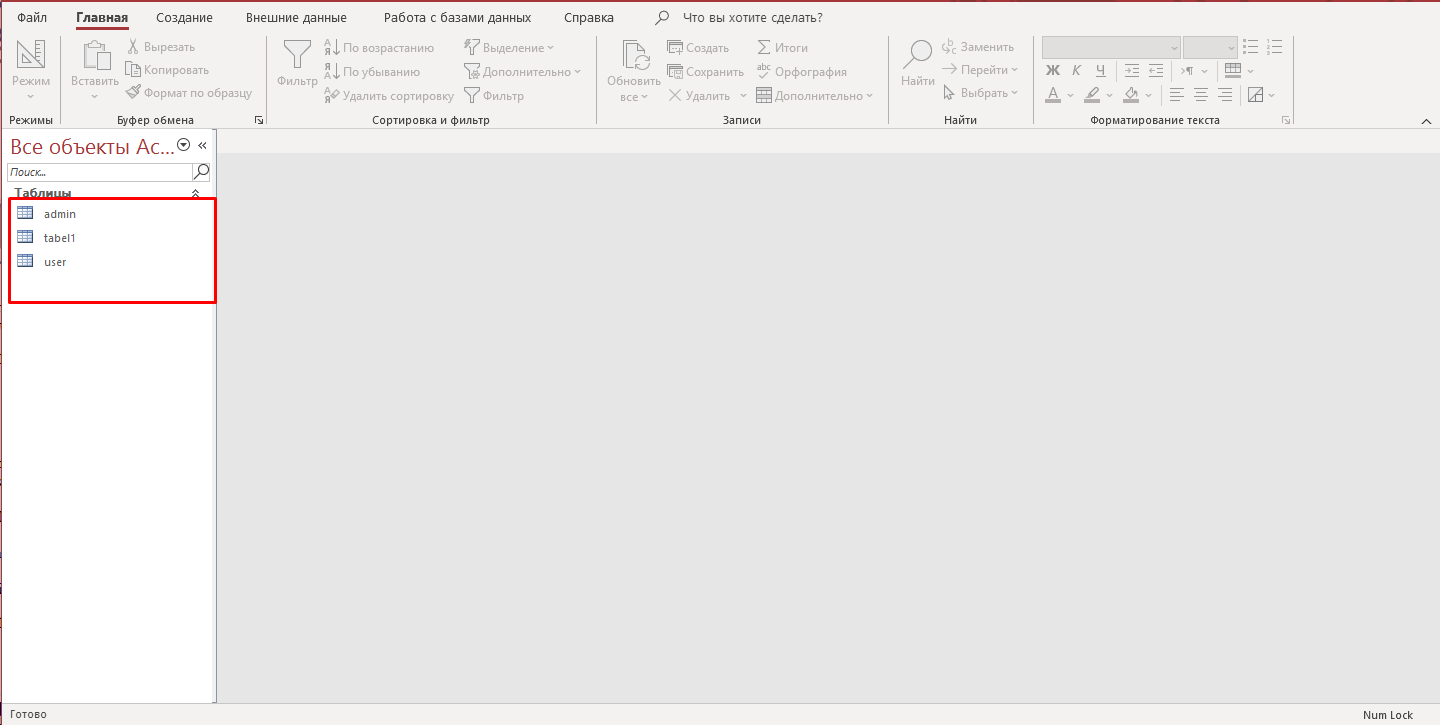
MBga qo’yiladigan talablar :

* Minimum xatolik va dulikat. MBBT ga ma’lumot kiritilayotganda dublikat bo’lmasligiga tekshiriladi;
* Aktualizatsiya imkoniyati. MB saklanayotgan ma’lumotlar eskirishi mumkin, bunda ma’lumotlar strukturasi yangi ma’lumotlarni kiritishga va eskisini uchirishga imkon berishi kerak.
* Ma’lumotlarni butunligini ta’minlash. MBBT ma’lumotlarni buzilishlardan ximoya kilishni ta’minlashi va buzilgan ma’lumotlarni tiklash imkoniyatini yaratish kerak;
* Yuqori qidirish tezligi. Xotira qurilmalarida ma’lumotlarni saklash usullari ma’lumotlar banki bilan ishlashni dialog rejimini ta’minlashi kerak;
* Xavfsizlik va maxfiylik. Foydalanuvchi fakat uziga berkitilgan zaruriyma’lumotlari bilan ishlashi kerak.
* Murakkab surovlar. Foydalanuvchilarni xar xil kurinishdagi surovlarini kayta ishlashni ta’minlashi kerak



* 1. **Rasm MS Access dasturini ishga tushirish.**

MS Access dasturini ishga tushurishingiz uchun 1.1 Rasmda strelka bilan ko’rsatilgan “Пустая база данних” ustiga sichqoncha bilan 2 marta chertamiz.



* 1. **Rasm. Jadvallar ro’yxati.**

## 1.2C++ Builder dasturida ma’lumotlar ombori bilan ishlaydigan komponentalar ro‘yxati.

C++Builder bosh xususiyati avvalam bor uning dasturni vizual ishlash jarayonida nafaqat tayyor komponentalardan foydalanish, balki yangi komponentalarni yaratish qobiliyatida ham namoyon bo‘ladi. Yangi komponentalar, dastlabki komponentalar kabi, sodda bo‘lishi mumkin, bunda ularning funksional imkoniyatlari sal-pal kengaytirilgan yoki o‘zining mutlaqo o‘ziga xos ko‘rinishi, xulq-atvori va kodining mazmuni bilan farqlanadigan bo‘ladi. Komponentalarning yaratilishi OMD ning vorislik mexanizmiga tayanadi, cheklanishlarga deyarli ega bo‘lmaydi hamda quyidagi bosqichlardan o‘tadi:

1. Mavjud komponenta turiga vorislik;
2. Yangi xususiyatlar, metodlar va voqealarni aniqlash;
3. Yaratilgan komponentani qayd etish.

Qidirish oson bo‘lishi uchun politra funktsional jihatdan o‘xshash komponentalarni birlashtiradigan qo‘shimcha ilovalar bilan bo‘lingan. Tanlab olingan komponentaning kontekst menyusini unga sichqonchaning o‘ng tugmasini bosib ochish mumkin.

C++Builder dasturlash tilini ma’lumotlar omborini bilan bog‘lash

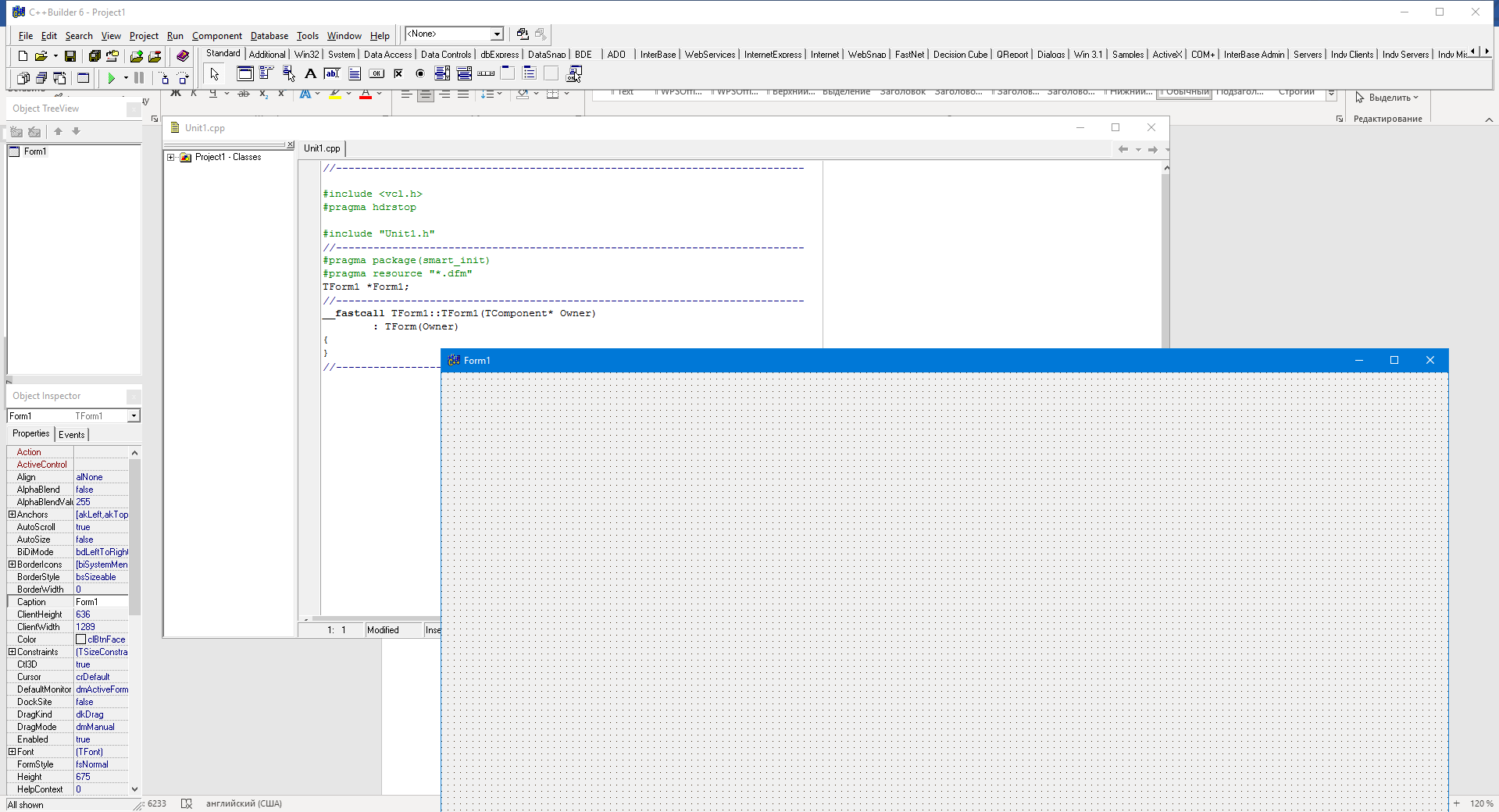
C++Builder dasturlash tilini ma’lumotlar ombori bilan bog‘lash turilari ko‘p bo‘lib ulardan biz ADO komponentlar to‘plamiga tegishli bo‘lgan ADOConnection, ADOTable va DataAccess komponentlar to‘plamiga tegish-li bo‘lgan DataSource komponentlaridan foydalanamiz.

|  |  |  |
| --- | --- | --- |
| Komponentining ko‘rinishi | Komponentning nomi | Vazifasi |
| Описание: AdoCon | ADOConnection | bu komponent ma’lumotlar bazasini C++ dasturlash tili bilan bir –biriga bog‘laydi. |
| Описание: AdoTable | ADOTable | bu komponent ma’lumotlar baza-sidagi jadvalni C++ dasturlash tili bilan bir – biriga bog‘laydi. |
| Описание: Datasour | DataSource | bu komponent ma’lumotlar bazasidagi jadvaldagi ma’lumotni inson ko‘ra oladigan ko‘rinishga keltiradi va aksincha. |

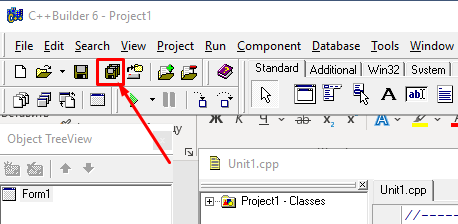
**II.Loyixa qism**.

## C++ Builder bilan ishlash. Form da Button, Edit bilan ishlash.

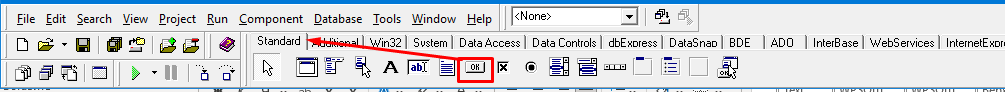
C++ Builder ni ishga tushirganimizda quyidagicha oynalar hosil bo`ladi.

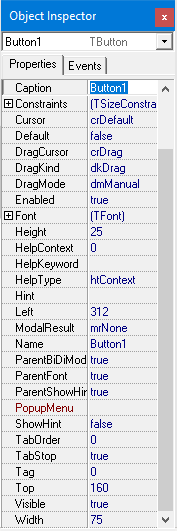


**C++ Builder 6 ning ishchi oynalari.**

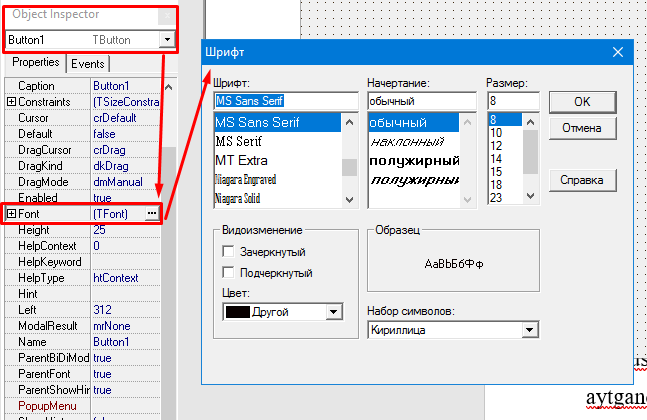


**SaveAll tugmasini bosgan holda alohida papkaga saqlab olamiz.**



Tugma C++ tilida **Button** deb ataladi. **Standart** bo’limi orqali ilovaga Button tugmasi tashlanadi. Button kompanentasining **Caption** xususiyatiga dastur davomida ushbu tugmaning ishlatiladigan nomi yoziladi. Visible xususiyati bizga bu tugmaning ilova ekranida ko’rinish yoki ko’rinmasligini belgilab beradi. Visible xususiyati mantiqiy turdagi qiymatni qabul qiladi. Ya’ni true va false. Agar True qiymat qabul qilsa ko’rinadigan holatda bo’ladi aks holda ko’rinmaydigan holatda bo’ladi.

Yana bir shunday xususiyat borki, u ish jarayonida tugmaning faol yoki nofaol ekanligini belgilab beradi. Bu xususiyatning nomi Enabled deb nomlanadi. Bu xususiyat ham mantiqiy turdagi qiymatlarni qabul qiladi. Agar Enabled xususiyati True holatda tursa Button faol holatda aks holda nofaol holda bo’ladi. Endi esa tugmaning ekrandagi korinishiga nom berish kerak. Buni Buttonning Text xususiyati orqali amalga oshiramiz. Bu yozuv ilovamiz ishlash jarayonida bizga noqulay ko’rinishda ya’ni yozuv katta yoki kichik yoki shrift turi noqulay ko’rinishda bo’lsa buni Buttonning Font xususiyati orqali to’g’irlash mumkin. **Font** ni tanlaganimizda bizga quyidagicha oyna hosil bo’ladi.



Bu yerda Buttonning Text xususiyatida joylashgan matnning shrift o’lchamini va shrift turini tanlashimiz mumkin. Tabiiyki, ba’zi hollarda Buttonning shrift o’lcham va turini tanlaganimizdan keyin yozuvning ranggini o’zgartirishimiz zarur bo’lib qoladi. Buni **Color** xususiyati orqali sozlashimiz mumkin.

Endi ilovamizga Label kompanentasini tashlaymiz va Labelning bazi xususiyatlari bilan tanishamiz. Label odatda qiyat kiritish uchun ishlatiladi. Lekin ba’zi istsno holatlarda Label dan faqat qiymatni o’qish uchun ham ishlatamiz. Label ning birinchi aytilishi kerak bo`lgan xususiyati uning **Caption** xususiyatidir. Bu xususiyatda unga kiritilgan yoki unga qabul qilingan qiymat saqlanadi. Qisqacha aytgandaundagi ma’lumot Caption xususiyatida joylashadi. Bu kompanentaning ham Font xususiyati orqali shriftni sozlashimiz mumkin. Bunda ham xuddi Buttondagi kabi Visible va Enabled xususiyatlari mavjud va xuddi shunday vazifani bajaradi. Uning **Name** xususiyatida Labelning dastur davomida qo’llanadigan nomi saqlanadi. Shuni e’tiborga olishimiz kerakki, Name va Captiont xususiyatlarining farqi bor.

Endi oldimizga shunday savol qo’yaylik. Ilovamizga joylashtirilgan tugmamizni bosganimizda birinchi Labelga yozilgan malumot ikkinchi Labelga o’zlashtirilsin?

Button tugmasini ikki marta chertganimizda dasturning kod yoziladigan ko’rinishiga o’tamiz va bu Button1Click(TObject \*Sender) funksiyasi yaratiladi. Yuqorida qo’yilgan savolimizni yechish uchun quyidagi ko’rinishdagi kodni kiritamiz.

Button1Click(TObject \*Sender)

{

Label1->Caption = Label2->Caption;

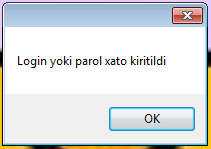
}

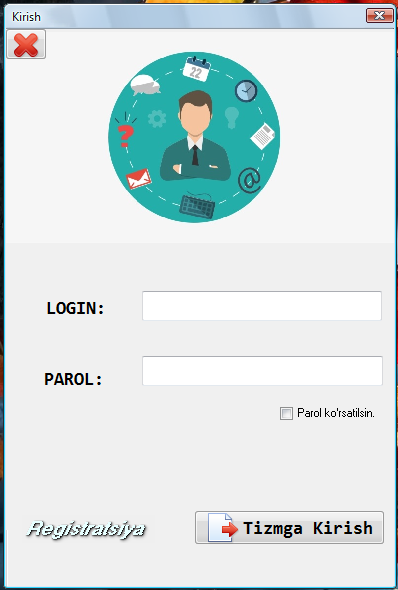
Label ning Caption xususiyati o’zida satr ko’rinishdagi ma’lumotlarni saqlaydi. Agar son bilan ishlashmoqchi bo’sak StrToInt(); dan foydalansak maqsadga muvofiq bo’ladi. Label ga kiritilayotgan ma’lumotni yoningizdagi shaxsga maxfiy shaklda kiritish zarur bo’ladigan hollar juda ko’p uchrab turadi.

## Talabalarni baholash tizimining yaratilishi

Bu loyiha universitet talabalarini o’zlashtirishini nazorat qilish va talabalarni baholash uchun yordam beradi. Biz dasturni ochganimizda bizga maxfiylikni saqlash uchun bizdan login va parol so’raladi. Biz login va parolni kiritib KIRISH tugmasini bosgan holda ishlash jarayonimizni davom etishimiz mumkin bo’ladi.

Agar parol xato kiritlsa bizga ekran markazidan quyidagi oynacha chiqib keladi.





Parolni ham maxfiy saqlash maqsadida Edit1 ning PasportChar xususiyatidan foydalanildi. Hamda 1 ta Checkbox dan foydalanildi. Ya’ni Checkboxga bayroqchani o’rnatadigan bo’lsak yashirin holda kiritilayotgan parol oshkor holda bo’lib qoladi. Checkboxning ustiga 2 marta chertib quyidagicha kodni yozamiz.

void \_\_fastcall TForm3::sCheckBox1Click(TObject \*Sender)

{

if(!(sCheckBox1->Checked == true))

sEdit2->PasswordChar = '\*';

else

sEdit2->PasswordChar = '\0';

}

***Login va parolni tekshirsh uchun bazadan malumotni tekshirib oldik.***

void \_\_fastcall TForm3::sBitBtn1Click(TObject \*Sender)

{

ADOQuery1->Close();

ADOQuery1->SQL->Clear();

ADOQuery1->SQL->Add("Select \* from admin WHERE login = \'"+sEdit1->Text+"\'");

ADOQuery1->Open();

if(ADOQuery1->FieldByName("login")->AsString==NULL)

{

ShowMessage("Login yoki Parol noto'g'ri!!!");

sEdit1->Clear();

sEdit2->Clear();

sEdit1->SetFocus();

return;

}

else{

ADOQuery1->Close();

ADOQuery1->SQL->Clear();

ADOQuery1->SQL->Add("Select \* from admin WHERE parol = \'"+sEdit2->Text+"\'");

ADOQuery1->Open();

if(ADOQuery1->FieldByName("parol")->AsString==NULL)

{

ShowMessage("Login yoki Parol noto'g'ri!!!");

sEdit1->Clear();

sEdit2->Clear();

sEdit1->SetFocus();

return;

}

else

{

sEdit1->Clear();

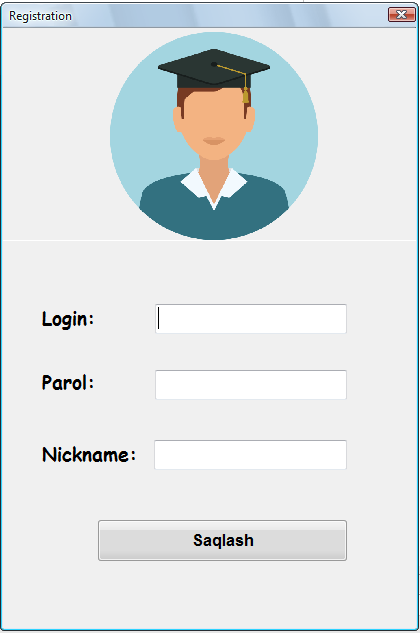
sEdit2->Clear();

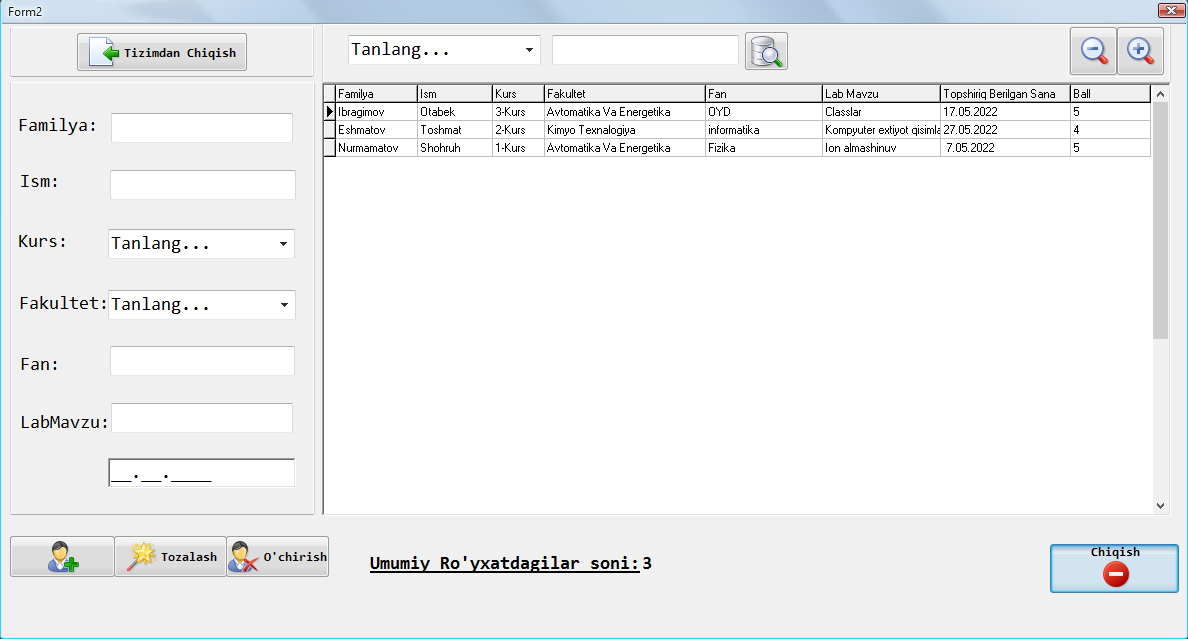
Form2->Show();

Form3->Visible=False;

}

}

“Registratsiya” yozuvini bossak biz registratsiya bo’lishimiz mumkin.



Ilovaning berildi. Buning uchun quyidagi kod yoziladi.

sEdit1->Clear();

sEdit2->Clear();

Form2->Show();

Form3->Visible=False;

Ilovaning **Tizimda chiqish** tugmasini bosish orqali dasturdan chiqiladi. Buning uchun quyidagicha kodni yozishimiz kerak.

void \_\_fastcall TForm2::sBitBtn5Click(TObject \*Sender)

{

Form2->Visible=False;

Form1->Visible=False;

Form3->Visible=True;

}

**Chiqish** tugmasi orqali dasturdan butunlay chiqib ketiladi.

void \_\_fastcall TForm2::sBitBtn1Click(TObject \*Sender)

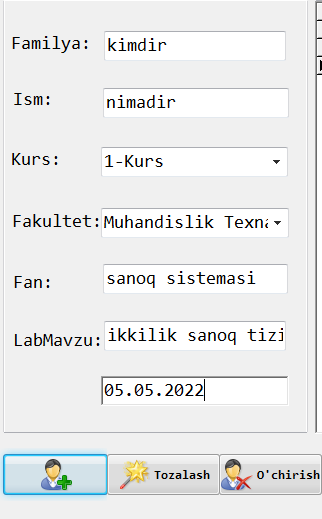
{

Form1->Close();

Form2->Close();

}

**Bu oynada talaba ma’lumotlarini kritib baxolashimiz ham mumkin.**



Bitta saqlash va bitta O’chirish tugmasi joylashgan. Hamda har bir saqlash tugmasiga quyidagicha kod yoziladi.

String ss, aa;

aa = sComboBox2->Items->Strings[sComboBox2->ItemIndex];

ss = sComboBox3->Items->Strings[sComboBox3->ItemIndex];

if (sComboBox2->ItemIndex>-1 && sComboBox2->ItemIndex<=3

&& sComboBox3->ItemIndex>-1 && sComboBox3->ItemIndex<=3)

{

if(sEdit2->Text.Length()>0 && sEdit3->Text.Length()>0

&& sEdit4->Text.Length()>0 && sEdit5->Text.Length()>0){

ADOQuery1->Close();

ADOQuery1->SQL->Clear();

ADOQuery1->SQL->Text="INSERT INTO tabel1 ([Familya], [Ism], [Kurs], [Fakultet], [Fan], [Lab Mavzu], [Topshiriq Berilgan Sana], [Ball]) VALUES (:p1, :p2, :p3, :p4, :p5, :p6, :p7, :p8) ";

ADOQuery1->Parameters->ParamByName("p1")->Value = sEdit2->Text;

ADOQuery1->Parameters->ParamByName("p2")->Value = sEdit3->Text;

ADOQuery1->Parameters->ParamByName("p3")->Value = sComboBox3->Items->Strings[sComboBox3->ItemIndex];

ADOQuery1->Parameters->ParamByName("p4")->Value = sComboBox2->Items->Strings[sComboBox2->ItemIndex];

ADOQuery1->Parameters->ParamByName("p5")->Value = sEdit4->Text;

ADOQuery1->Parameters->ParamByName("p6")->Value = sEdit5->Text;

ADOQuery1->Parameters->ParamByName("p7")->Value = MaskEdit1->Text;

ADOQuery1->ExecSQL();

ShowMessage("Saqlandi.");

ADOQuery1->SQL->Clear();

ADOQuery1->SQL->Add( "select \* from tabel1" );

ADOQuery1->Open();

int k = DBGrid1->DataSource->DataSet->RecordCount;

sLabel9->Caption=IntToStr(k);

}

else{

ShowMessage("To'liq hamma qatorlarni to'ldiring!");

}

}

else

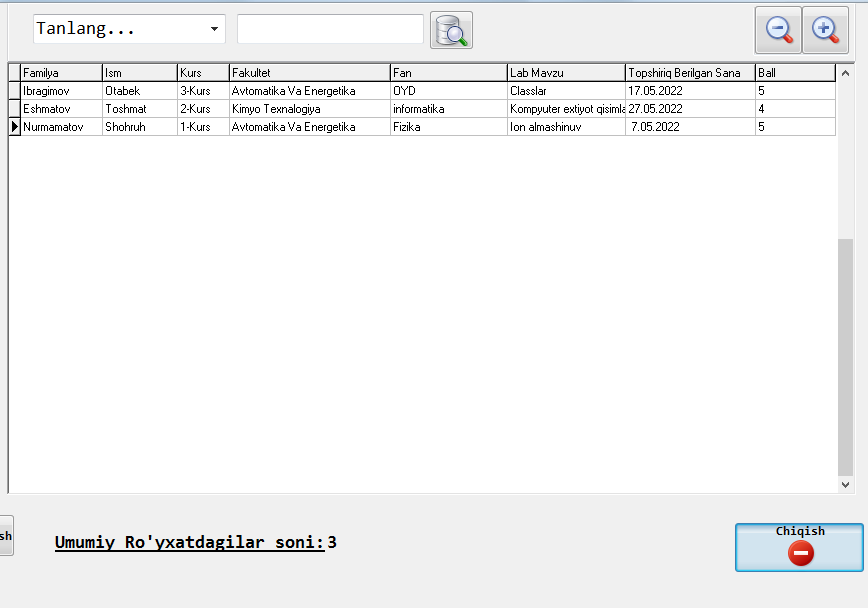
{

ShowMessage("Tanlanmadi!");

}

}

Endi esa Talabalar haqidagi malumotlarni dasturni ishlash jarayoni bilan tanishib chiqaylik. Agar ushbu ushbu jadvalda bizga yangi oyna ochiladi. Bu oynada biz talabalar haqidagi ma’lumotlar bilan tanishib chiqishimiz mumkin.



Yuqoridagi rasmdagi O’chirish tugmasining vazifasi kiritilgan bazadagi biror talaba haqidagi to’liq malumotlarni o’chirib tashlashdan iborat. Yani o’chirmoqchi bo’lgan talabaning jadvaldan belgisak va **o’chirish** tugmasini bosamiz.

int k = DBGrid1->DataSource->DataSet->RecordCount;

sLabel9->Caption=IntToStr(k);

if (k>0){

String del=DBGrid1->DataSource->DataSet->FieldByName("ID")->AsString;

if (Application->MessageBoxA("Foydalanuvchini o`chirmoqchimisiz.","Tasdiqlang!",MB\_YESNO|MB\_ICONINFORMATION)==ID\_YES)

{

ADOQuery1->Close();

ADOQuery1->SQL->Clear();

ADOQuery1->SQL->Add("DELETE FROM tabel1 WHERE ID="+del);

ADOQuery1->ExecSQL();

//ADOQuery1->Active=false;

//ADOQuery1->Active=true;

}

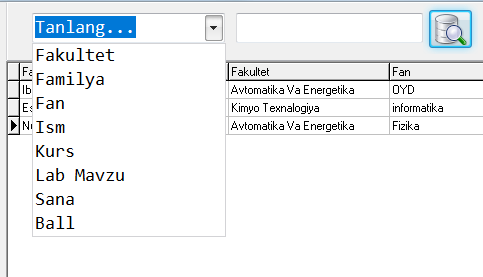
}

else{ShowMessage("O'chirolmaysiz!!!");}

k = DBGrid1->DataSource->DataSet->RecordCount;

sLabel9->Caption=IntToStr(k);

Qidirish tugmasining vazifasi malumotlar bazasidan biror talabani izlab topishdan iborat.



Familya tanlsak va yonidagi maydonga qidirmoqchi bolgan familyani qidirishimiz kerak bo’lgan familiyani kiritib Familiya qidirishimiz mumkin. Ism nomli maydonga qidirishimiz kerak bo’lgan ismni kiritib shunday ismli talabalarni qidirib topib olishimiz mumkin. Fakultet nomli maydonga qidirishimiz kerak bo’lgan fakultetni kiritib 1 fakultetda o’qiydigan talabalar tizimini qidirib topamiz. Quyida familiya orqali qidirish tugmasida yashiringan kodni keltirib o’tamiz.

String s;

s=sComboBox1->Items->Strings[sComboBox1->ItemIndex];

if(sComboBox1->ItemIndex>-1 && sEdit1->Text.Length()>0){

switch(sComboBox1->ItemIndex){

case 0:

DBGrid1->DataSource->DataSet->Locate("Fakultet", sEdit1->Text, TLocateOptions() << loCaseInsensitive << loPartialKey);

DBGrid1->SetFocus();

break;

case 1:

DBGrid1->DataSource->DataSet->Locate("Familya", sEdit1->Text, TLocateOptions() << loCaseInsensitive << loPartialKey);

DBGrid1->SetFocus();

break;

case 2:

DBGrid1->DataSource->DataSet->Locate("Fan", sEdit1->Text, TLocateOptions() << loCaseInsensitive << loPartialKey);

DBGrid1->SetFocus();

break;

case 3:

DBGrid1->DataSource->DataSet->Locate("Ism", sEdit1->Text, TLocateOptions() << loCaseInsensitive << loPartialKey);

DBGrid1->SetFocus();

break;

case 4:

DBGrid1->DataSource->DataSet->Locate("Kurs", sEdit1->Text, TLocateOptions() << loCaseInsensitive << loPartialKey);

DBGrid1->SetFocus();

break;

case 5:

DBGrid1->DataSource->DataSet->Locate("Lab Mavzu", sEdit1->Text, TLocateOptions() << loCaseInsensitive << loPartialKey);

DBGrid1->SetFocus();

break;

case 6:

DBGrid1->DataSource->DataSet->Locate("Sana", sEdit1->Text, TLocateOptions() << loCaseInsensitive << loPartialKey);

DBGrid1->SetFocus();

break;

case 7:

DBGrid1->DataSource->DataSet->Locate("Ball", sEdit1->Text, TLocateOptions() << loCaseInsensitive << loPartialKey);

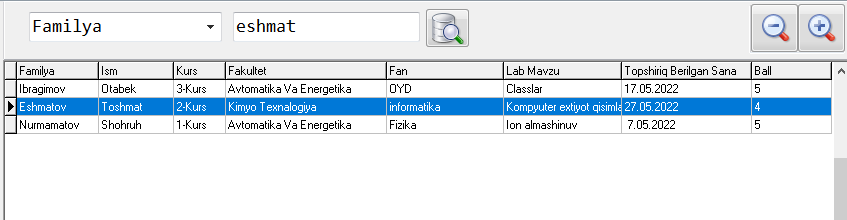
DBGrid1->SetFocus();

break;

}

}

}



**Xulosa:**

Men ushbu kurs ishini bajarish davomida C++ dasturlash tilida dasturlash to’g’risidagi bilimlarimni amaliy jihatdan o’rgandim va nazariy bilimlarimni amaliy jihatdan mustahkamlab oldim. Dastur tuzish davomida C++ Builderda ma’lumotlar bazalari bilan ishlashni o’rgandim. Ma’lumotlar bazalarini dasturga bog’lash, ma’lumotlar bazalaridagi jadvallar bilan ishlash, ya’ni ma’lumotlarni qo’shish, qo’shilgan ma’lumotlarni o’zgartirish, o’chirish kabi amallarni bajarishni o’rganib oldim va amalga oshirdim. C++ Builderda mavjud bo’lgan ko’plab komponentalar haqida ma’lumotga ega bo’ldim. Komponentalarning turlari, ularning vazifalari, dasturda yaratadigan qulayliklari haqida tushunchaga ega bo’ldim va ko’plab komponentalarni o’z dasturimni tuzish davomida ishlatdim. Kurs ishini bajarish davomida dasturdagi ko’plab komponentalar palitrasidan, komponentalardan foydalanishni o’rgandim.O’z dasturimni tayyorlash jarayonida dasturda yuzaga kelishi mumkin bo’lgan hatoliklar, hamda turli xildagi istisno holatlarni oldindan bilib, ularni bartaraf qilish bo’yicha kerakli chora tadbirlarni amalga oshirdim.Dasturda hatolikka olib keladigan ma’lumotlar kiritilganda foydalanuvchini ogoxlantirish xabarlarini chiqaruvchi dastur kodlaridan foydalandim.

O’z dasturimni tayyorlash jarayonida shunga amin bo’ldimki, har bir dasturchi, o’z dasturini tuzar ekan, avvalambor uning foydalanuvchiga tushunarli, boshqarishga oson, hamma qismlari puxta hisob-kitob qilingan, dasturda yuz berishi mumkin bo’lgan xatoliklarni oldi olingan, dasturning yashash sikli hisbga olingan bo’lishi lozim ekan. Shundagina yaratilgan dastur foydalanuvchiga uzoq muddat, samarali hizmat qilishi mumkin.

**Foydalanilgan adabiyotlar**

1. Культин Н. Б. Самоучитель C++ Builder
2. О.Д.Вальпа - Borland C++ Builder. Экспресс-курс – 2006
3. Самоучительпрограммирования C++
4. Самоучитель программирования на языке С++ в системе С++Build
5. Федоренко Ю.П. - Алгоритмы и программы на C++ Builder (Профессиональная серия для программистов) – 2010

Internet manbalari:

[www.cyber-form.ru](http://www.cyber-form.ru)

[www.wikipedia.org](http://www.wikipedia.org)

[www.programmer.com](http://www.programmer.com)

## Ilovalar

**Kirish**

**//---------------------------------------------------------------------------**

**#include <vcl.h>**

**#pragma hdrstop**

**#include "Unit1.h"**

**#include "Unit2.h"**

**#include "Unit3.h"**

**//---------------------------------------------------------------------------**

**#pragma package(smart\_init)**

**#pragma link "sSkinManager"**

**#pragma link "sGauge"**

**#pragma link "sSkinManager"**

**#pragma resource "\*.dfm"**

**TForm1 \*Form1;**

**//---------------------------------------------------------------------------**

**\_\_fastcall TForm1::TForm1(TComponent\* Owner)**

**: TForm(Owner)**

**{**

**}**

**//---------------------------------------------------------------------------**

**void \_\_fastcall TForm1::Timer1Timer(TObject \*Sender)**

**{**

**sGauge1->Progress=sGauge1->Progress+20; //progress barni 1 min da tolishi**

**}**

**//---------------------------------------------------------------------------**

**void \_\_fastcall TForm1::sGauge1Change(TObject \*Sender)**

**{**

**if (sGauge1->Progress==100)**

**{**

**Form1->Visible=False;**

**Form3->Show();**

**}**

**}**

**//---------------------------------------------------------------------------**

**Login va Parol ni tekshirish**

**//---------------------------------------------------------------------------**

**#include <vcl.h>**

**#pragma hdrstop**

**#include "Unit3.h"**

**#include "Unit1.h"**

**#include "Unit2.h"**

**#include "Unit6.h"**

**//---------------------------------------------------------------------------**

**#pragma package(smart\_init)**

**#pragma link "sBitBtn"**

**#pragma link "sCheckBox"**

**#pragma link "sEdit"**

**#pragma link "sLabel"**

**#pragma link "sPanel"**

**#pragma link "sButton"**

**#pragma resource "\*.dfm"**

**TForm3 \*Form3;**

**//---------------------------------------------------------------------------**

**\_\_fastcall TForm3::TForm3(TComponent\* Owner)**

**: TForm(Owner)**

**{**

**}**

**//---------------------------------------------------------------------------**

**void \_\_fastcall TForm3::sCheckBox1Click(TObject \*Sender)**

**{**

**if(!(sCheckBox1->Checked == true))**

**sEdit2->PasswordChar = '\*';**

**else**

**sEdit2->PasswordChar = '\0';**

**}**

**//---------------------------------------------------------------------------**

**void \_\_fastcall TForm3::sBitBtn1Click(TObject \*Sender)**

**{**

**ADOQuery1->Close();**

**ADOQuery1->SQL->Clear();**

**ADOQuery1->SQL->Add("Select \* from admin WHERE login = \'"+sEdit1->Text+"\'");**

**ADOQuery1->Open();**

**if(ADOQuery1->FieldByName("login")->AsString==NULL)**

**{**

**ShowMessage("Login yoki Parol noto'g'ri!!!");**

**sEdit1->Clear();**

**sEdit2->Clear();**

**sEdit1->SetFocus();**

**return;**

**}**

**else{**

**ADOQuery1->Close();**

**ADOQuery1->SQL->Clear();**

**ADOQuery1->SQL->Add("Select \* from admin WHERE parol = \'"+sEdit2->Text+"\'");**

**ADOQuery1->Open();**

**if(ADOQuery1->FieldByName("parol")->AsString==NULL)**

**{**

**ShowMessage("Login yoki Parol noto'g'ri!!!");**

**sEdit1->Clear();**

**sEdit2->Clear();**

**sEdit1->SetFocus();**

**return;**

**}**

**else**

**{**

**sEdit1->Clear();**

**sEdit2->Clear();**

**Form2->Show();**

**Form3->Visible=False;**

**}**

**}**

**}**

**//---------------------------------------------------------------------------**

**void \_\_fastcall TForm3::FormCloseQuery(TObject \*Sender, bool &CanClose)**

**{**

**ShowMessage("Dasturdan Chiqish Tugmasi Orqali Chiqiladi!");**

**CanClose=false;**

**return;**

**}**

**//---------------------------------------------------------------------------**

**void \_\_fastcall TForm3::sLabelFX1Click(TObject \*Sender)**

**{**

**Form2->Visible=False;**

**Form6->ShowModal();**

**}**

**//---------------------------------------------------------------------------**

**void \_\_fastcall TForm3::sBitBtn2Click(TObject \*Sender)**

**{**

**Form1->Close();**

**Form3->Close();**

**}**

**//---------------------------------------------------------------------------**

**Login va parolni o`zgartirish**

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit6.h"

#include "Unit3.h"

#include "Unit1.h"

#include "Unit2.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma link "acPNG"

#pragma link "sPanel"

#pragma link "sBitBtn"

#pragma link "sEdit"

#pragma link "sLabel"

#pragma resource "\*.dfm"

TForm6 \*Form6;

//---------------------------------------------------------------------------

\_\_fastcall TForm6::TForm6(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm6::sBitBtn1Click(TObject \*Sender)

{

if(sEdit1->Text.Length()>0 && sEdit1->Text.Length()>0 && sEdit1->Text.Length()>0)

{

ADOQuery1->Close();

ADOQuery1->SQL->Clear();

ADOQuery1->SQL->Add("SELECT \* FROM admin WHERE nickname = \'"+sEdit3->Text+"\'");

ADOQuery1->Open();

if(ADOQuery1->IsEmpty())

{

ADOQuery1->Close();

ADOQuery1->SQL->Clear();

ADOQuery1->SQL->Add("INSERT INTO admin (login, parol, nickname) VALUES ('"+sEdit1->Text +"', '"+sEdit2->Text+"', '"+sEdit3->Text+"')");

ADOQuery1->ExecSQL();

ShowMessage("Saqlandi.");

sEdit1->Clear();

sEdit2->Clear();

}

else

{

ShowMessage("Bu nom band qilingan.");

sEdit1->Clear();

sEdit2->Clear();

sEdit1->SetFocus();

return;

}

}

else

{

ShowMessage("Barcha qatorlarni to'ldiring.");

}

}

//---------------------------------------------------------------------------

**Asosiy oyna**

//---------------------------------------------------------------------------

#include <vcl.h>

#pragma hdrstop

#include "Unit2.h"

#include "Unit1.h"

#include "Unit3.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma link "sBitBtn"

#pragma link "sComboBox"

#pragma link "sEdit"

#pragma link "sPanel"

#pragma link "sLabel"

#pragma resource "\*.dfm"

TForm2 \*Form2;

//---------------------------------------------------------------------------

\_\_fastcall TForm2::TForm2(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::FormCloseQuery(TObject \*Sender, bool &CanClose)

{

ShowMessage("Dasturdan Chiqish Tugmasi Orqali Chiqiladi!");

CanClose=false;

return;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::sBitBtn2Click(TObject \*Sender)

{

String ss, aa;

aa = sComboBox2->Items->Strings[sComboBox2->ItemIndex];

ss = sComboBox3->Items->Strings[sComboBox3->ItemIndex];

if (sComboBox2->ItemIndex>-1 && sComboBox2->ItemIndex<=3

&& sComboBox3->ItemIndex>-1 && sComboBox3->ItemIndex<=3)

{

if(sEdit2->Text.Length()>0 && sEdit3->Text.Length()>0

&& sEdit4->Text.Length()>0 && sEdit5->Text.Length()>0){

ADOQuery1->Close();

ADOQuery1->SQL->Clear();

ADOQuery1->SQL->Text="INSERT INTO tabel1 ([Familya], [Ism], [Kurs], [Fakultet], [Fan], [Lab Mavzu], [Topshiriq Berilgan Sana], [Ball]) VALUES (:p1, :p2, :p3, :p4, :p5, :p6, :p7, :p8) ";

ADOQuery1->Parameters->ParamByName("p1")->Value = sEdit2->Text;

ADOQuery1->Parameters->ParamByName("p2")->Value = sEdit3->Text;

ADOQuery1->Parameters->ParamByName("p3")->Value = sComboBox3->Items->Strings[sComboBox3->ItemIndex];

ADOQuery1->Parameters->ParamByName("p4")->Value = sComboBox2->Items->Strings[sComboBox2->ItemIndex];

ADOQuery1->Parameters->ParamByName("p5")->Value = sEdit4->Text;

ADOQuery1->Parameters->ParamByName("p6")->Value = sEdit5->Text;

ADOQuery1->Parameters->ParamByName("p7")->Value = MaskEdit1->Text;

ADOQuery1->ExecSQL();

ShowMessage("Saqlandi.");

ADOQuery1->SQL->Clear();

ADOQuery1->SQL->Add( "select \* from tabel1" );

ADOQuery1->Open();

int k = DBGrid1->DataSource->DataSet->RecordCount;

sLabel9->Caption=IntToStr(k);

}

else{

ShowMessage("To'liq hamma qatorlarni to'ldiring!");

}

}

else

{

ShowMessage("Tanlanmadi!");

}

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::sBitBtn4Click(TObject \*Sender)

{

int k = DBGrid1->DataSource->DataSet->RecordCount;

sLabel9->Caption=IntToStr(k);

if (k>0){

String del=DBGrid1->DataSource->DataSet->FieldByName("ID")->AsString;

if (Application->MessageBoxA("Foydalanuvchini o`chirmoqchimisiz.","Tasdiqlang!",MB\_YESNO|MB\_ICONINFORMATION)==ID\_YES)

{

ADOQuery1->Close();

ADOQuery1->SQL->Clear();

ADOQuery1->SQL->Add("DELETE FROM tabel1 WHERE ID="+del);

ADOQuery1->ExecSQL();

//ADOQuery1->Active=false;

//ADOQuery1->Active=true;

}

}

else{ShowMessage("O'chirolmaysiz!!!");}

k = DBGrid1->DataSource->DataSet->RecordCount;

sLabel9->Caption=IntToStr(k);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::sBitBtn3Click(TObject \*Sender)

{

sComboBox2->ItemIndex = -1;

sComboBox3->ItemIndex = -1;

sEdit2->Text = "";

sEdit3->Text = "";

sEdit4->Text = "";

sEdit5->Text = "";

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::sBitBtn5Click(TObject \*Sender)

{

Form2->Visible=False;

Form1->Visible=False;

Form3->Visible=True;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::sBitBtn1Click(TObject \*Sender)

{

Form1->Close();

Form2->Close();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::sBitBtn6Click(TObject \*Sender)

{

DBGrid1->Font->Size=StrToInt(DBGrid1->Font->Size)-2;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::sBitBtn7Click(TObject \*Sender)

{

DBGrid1->Font->Size=StrToInt(DBGrid1->Font->Size)+2;

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::FormCreate(TObject \*Sender)

{

int k = DBGrid1->DataSource->DataSet->RecordCount;

sLabel9->Caption=IntToStr(k);

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm2::sBitBtn8Click(TObject \*Sender)

{

String s;

s=sComboBox1->Items->Strings[sComboBox1->ItemIndex];

if(sComboBox1->ItemIndex>-1 && sEdit1->Text.Length()>0){

switch(sComboBox1->ItemIndex){

case 0:

DBGrid1->DataSource->DataSet->Locate("Fakultet", sEdit1->Text, TLocateOptions() << loCaseInsensitive << loPartialKey);

DBGrid1->SetFocus();

break;

case 1:

DBGrid1->DataSource->DataSet->Locate("Familya", sEdit1->Text, TLocateOptions() << loCaseInsensitive << loPartialKey);

DBGrid1->SetFocus();

break;

case 2:

DBGrid1->DataSource->DataSet->Locate("Fan", sEdit1->Text, TLocateOptions() << loCaseInsensitive << loPartialKey);

DBGrid1->SetFocus();

break;

case 3:

DBGrid1->DataSource->DataSet->Locate("Ism", sEdit1->Text, TLocateOptions() << loCaseInsensitive << loPartialKey);

DBGrid1->SetFocus();

break;

case 4:

DBGrid1->DataSource->DataSet->Locate("Kurs", sEdit1->Text, TLocateOptions() << loCaseInsensitive << loPartialKey);

DBGrid1->SetFocus();

break;

case 5:

DBGrid1->DataSource->DataSet->Locate("Lab Mavzu", sEdit1->Text, TLocateOptions() << loCaseInsensitive << loPartialKey);

DBGrid1->SetFocus();

break;

case 6:

DBGrid1->DataSource->DataSet->Locate("Sana", sEdit1->Text, TLocateOptions() << loCaseInsensitive << loPartialKey);

DBGrid1->SetFocus();

break;

case 7:

DBGrid1->DataSource->DataSet->Locate("Ball", sEdit1->Text, TLocateOptions() << loCaseInsensitive << loPartialKey);

DBGrid1->SetFocus();

break;

}

}

}//---------------------------------------------------------------------------