

O'ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI
VA KOMMUNIKATSIYALARNI RIVOJLANTIRISH VAZIRLIGI

MUHAMMAD AL – XORAZMIY NOMIDAGI TOSHKENT AXBOROT
TEXNOLOGIYALARI UNIVERSITETI SAMARQAND FILIALI

“Kompyuter injiniringi” fakulteti

“Kompyuter tizimlari” kafedrası

5330500-“Kompyuter injiniringi”(Kompyuter injiniringi, AT servisi) yo'nalishi

MALAKAVIY (BMI OLDI) AMALIYOTI

2-3-HAFTALIK HISOBOTI

**Mavzu: Kafedra faoliyati monitoringini olib boruvchi dasturiy platformasini
yaratish usullari**

***BMI mavzusi: Kafedra faoliyati monitoringini olib boruvchi dasturiy
platformasini ishlab chiqish***

Bitiruvchi: 401- guruh talabasi
Boyqobilova Sitora.

Ilmiy rahbar: Hamiyev A.T.

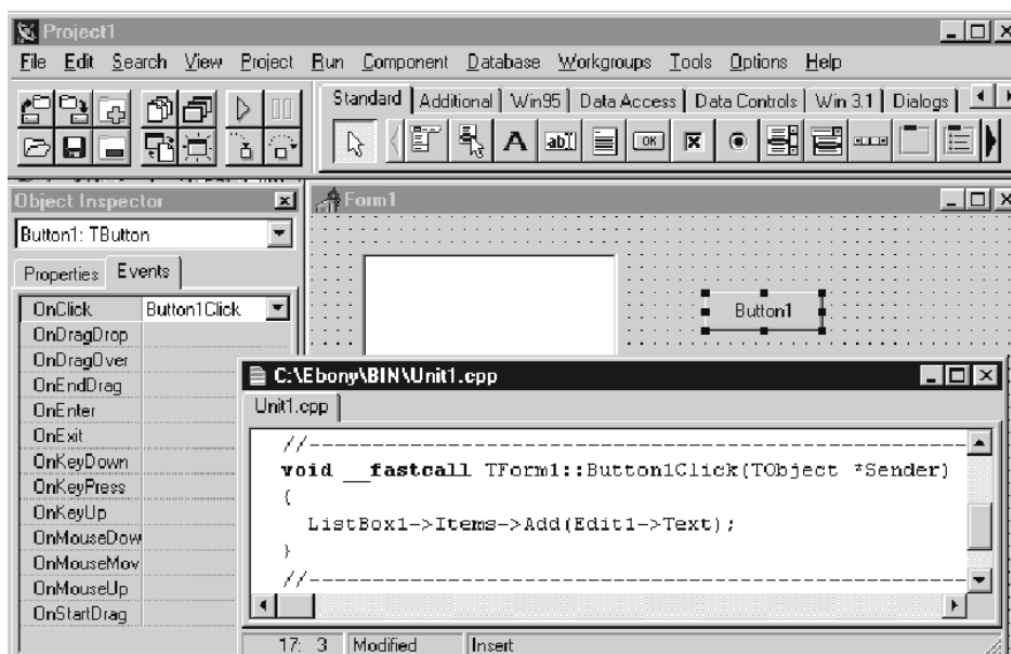
Amaliyot rahbari: Sobirov R.

SAMARQAND 2020

2-bob. Kafedra faoliyati monitoringini olib boruvchi dasturiy platformasini yaratish usullari

2.1. Kafedrani faoliyati monitoringini olib boruvchi dasturiy platformasini C++ Builder da dasturlash

C++ Builder dasturi SDI ilova dasturi hisoblanib, asosan sozlovchi asboblari paneli (chap tomonda) va komponentlar palitrasini (o'ng tomonda) o'z ichiga oladi. Bundan tashqari C++ Builder yuklanayotganda ob'yektlar inspektori (chap tomonda) hamda yangi ilova dasturi shakli (o'ng tomonda) paydo bo'ladi. Yangi ilova dasturi shakli orqasida kodni tahrirlovchi oyna ham mavjud bo'ladi. Shakllar C++ Builder ning asosi hisoblanadi. Foydalanuvchi ilova dasturi interfeysini yaratish uchun oynaga C++ Builder ob'yekti formasi elementlarini, ya'ni komponentlarni qo'yish bilan amalga oshiriladi. C++ Builder komponentlari o'ng tomondagi komponentlar palitrasida joylashgan bo'ladi. C++ Builderning eng yaxshi tomonlaridan biri shuki, unda dasturchi o'zi hohlagan yangi komponent yaratishi va uni komponentlar palitrasiga sozlab olishi mumkin. Bundan tashqari turli loyihalar uchun yangi komponentlar yaratish mumkin.

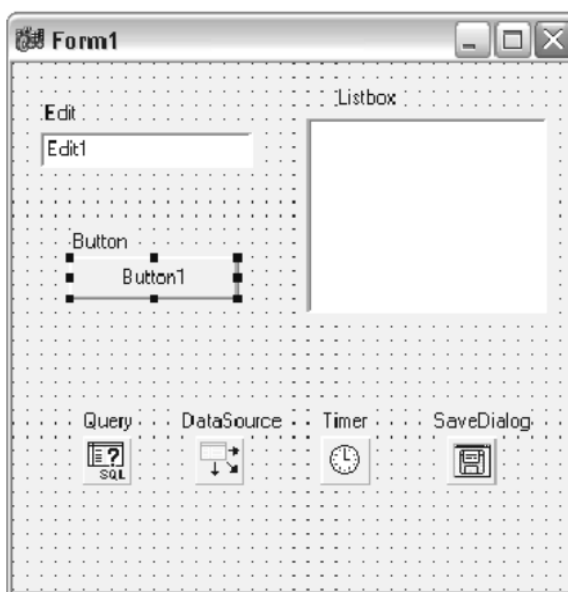


2.1.1-rasm. C++ Builderning umumiy ko'rinishi.

C++ Bulder komponentlari. Komponentlar ko'rinadigan (vizual) va ko'rinmaydigan komponentlarga bo'linadi. Vizual komponentlar loyihalashtirayotgan paytda qanday holatda ko'rinsa, dastur ishga tushganda ham shunday ko'rinadi. Ko'rinmaydigan komponentlar loyihalashtirish paytida formadagi piktogramma ko'rinishida bo'ladi. Ular dastur ishga tushgan paytda

umuman ko'rinmaydi, ammo aniq bir vazifaga egadir (masalan, ma'lumotlar bilan aloqa qilish imkonini beradi yoki Windows satndart dialoglarini chaqiradi).

Formaga component qo'shish uchun sichqoncha yordamida palitradagi kerakli komponentni tanlab loyihalanayotgan formaning kerakli joyiga olib kelib, sichqonchaning chap tugmasini bosish kerak. Komponent formada paydo bo'ladi, keyin uni boshqa joyga siljitish, o'lchovi yoki boshqa xarakteristikalarini o'zgartirish mumkin. C++ Builder komponentlarining uch xil xarakteristikasi bor: tarkibi, hodisa va uslub. Agar palitradan komponentni tanlab uni formaga qo'yilsa, ob'yektlar inspektori komponent bilan ishlanadigan uning tarkibi va hodisasini ko'rsatadi. Ob'yektlar inspektorining yuqori qismida formada mavjud bo'lgan ob'yektlardan birini tanlash imkonini beruvchi ro'yhat mavjud bo'ladi.

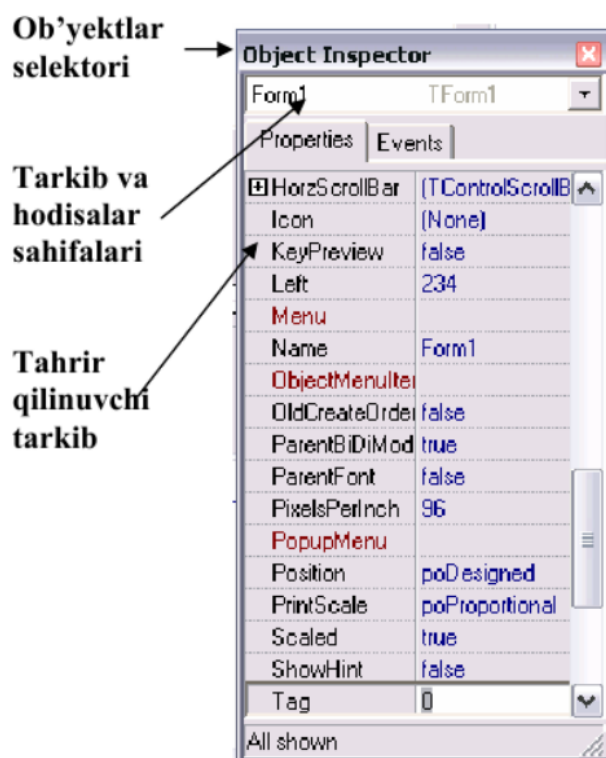


2.1.2-rasm. Ko'rinadigan va ko'rinmaydigan komponentlardan foydalanish bo'yicha misol.

Komponentlar tarkibi. Komponent tarkibi uning tashqi ko'rinishi va harakatini boshqaruvchi atribut hisoblanadi. Komponentlar tarkibi tarkiblar (Properties) sahifasida ko'rinib turadi. Ob'yektlar inspektori nashrlangan (published) komponent tarkibini ko'rsatib turadi. Shunday ekan komponentlar faqat ilova dastur ishga tushganda kirish mumkin bo'ladigan umumiy (public) tarkibga ega bo'ladi. Ob'yektlar inspektori dastur loyihanayotgan paytda komponent tarkibini sozlash uchun foydalaniladi. Komponent tarkibini loyialash davomida aniqlash o'zgartirish mumkin yoki dastur ishga tushayotgan paytda tarkibni visual o'zgartirish uchun kod yozish mumkin.

Loyihalar menejeri. Loyihalar menejeri ilova dasturining fayllari va modullar ro'yxatini ko'rsatadi va u lar orasidan avigatsiya o'rnatadi. View/Project Manager yo'li bilan loyiha menegerini ochish mumkin. Yaratilgan loyihaning avtomatik ravishda berilgan dastlabki nomi Project1.cpp bo'ladi.

Dastlab loyiha bitta forma uchun boshlang'ich fayllarni o'z ichiga oladi. Ammo ko'p loyihalarda bir necha forma va modullar mavjud bo'ladi. Loyihaga modul yoki forma qo'shish uchun modul yoki forma qo'shish uchun sichqoncha o'ng tugmasini bosib, kontekst menyudan New Form punktini tanlash kerak.

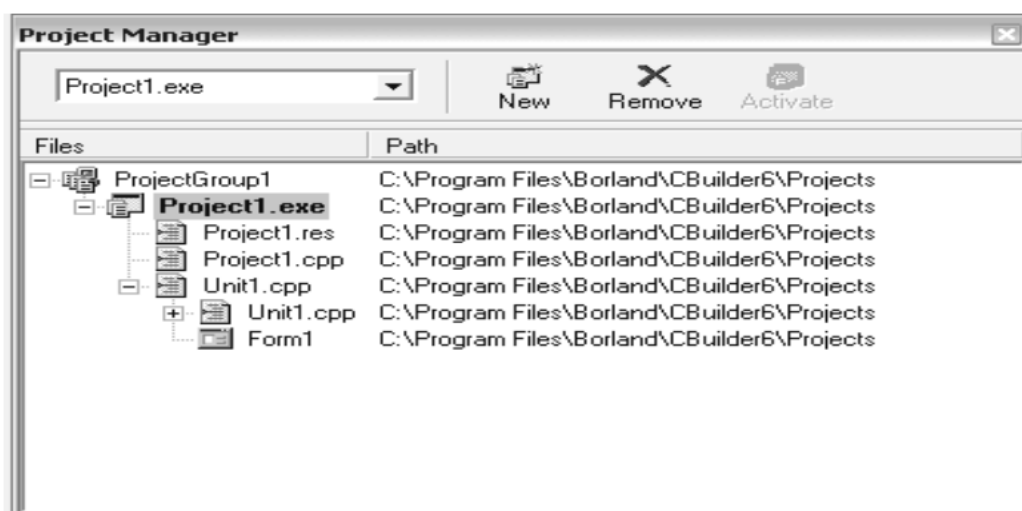


2.1.2-rasm. Obyektlar inspektori.

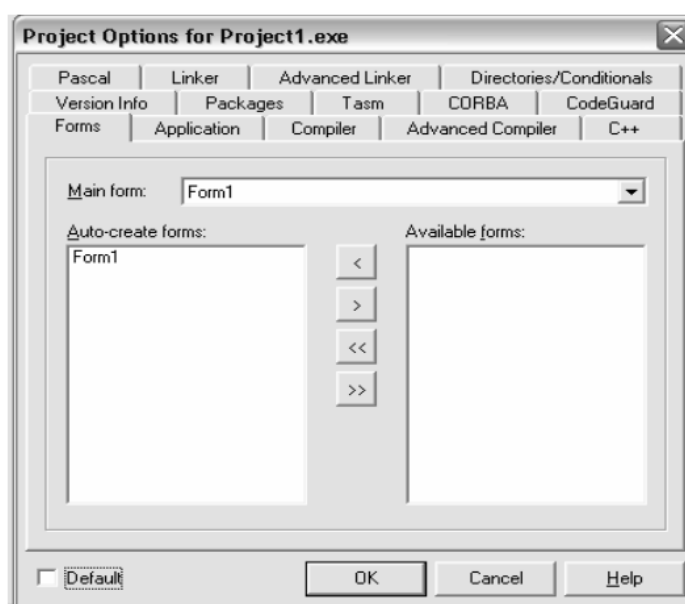
Loyihaga kompyuterda mavjud forma yoki modulni qo'shish ham mumkin. Buning uchun loyiha menejerining kontekst menyusidan Add ni bosib, qo'shish kerak bo'lgan modul yoki formani tanlash kerak. Loyiha ishlayotganda ixtiyoriy paytda forma yoki modulni o'chirib yuborish mumkin. Ammo modul bilan forma bir-biriga bog'liq bo'lgani uchun ulardan birini ikkinchisiz o'chirmaslik kerak. Lekin bu ishni modul formaga bog'liq bo'lmagan paytda amalga oshirish mumkin. Loyihadan modulni o'chirish uchun loyiha menejeridagi Remove tugmasidan foydalaniladi.

Loyiha menejeridan Options tugmasi tanlansa, loyiha opsiyasi, ya'ni muloqot panel ochiladi. Unda dasturning asosiy formasini tanlash, dinamik ravishda qanday formalar yaratilishi va modul kompilyatsiyasi parametrlari qanaqaligini aniqlash mumkin.

C++ Builder ish maydonida muhim elementlardan biri sichqonchanning o'ng tugmasi bosilganda paydo bo'ladigan kontekst menyudir. U tez-tez ishlanadigan buyruqlarga tez kirish imkonini beradi.



2.1.3-rasm. Loyiha menejeri



2.1.4-rasm. Loyiha opsiyasini sozlash

C++ Builder da ilova dastur yaratish. C++ Builder dasturida ishni boshlashning birinchi qadami bu loyiha yaratishdir. Loyiha fayllarining boshlanishida avtomatik ravishda dastlabki matnlari bo'ladi. Kompilyatsiya qilingan va ishlashga tayyorlanganida u ilova dasturning bir qismi bo'lib qoladi. Yangi loyiha yaratish uchun File/New Application menyu punktini tanlash kerak.

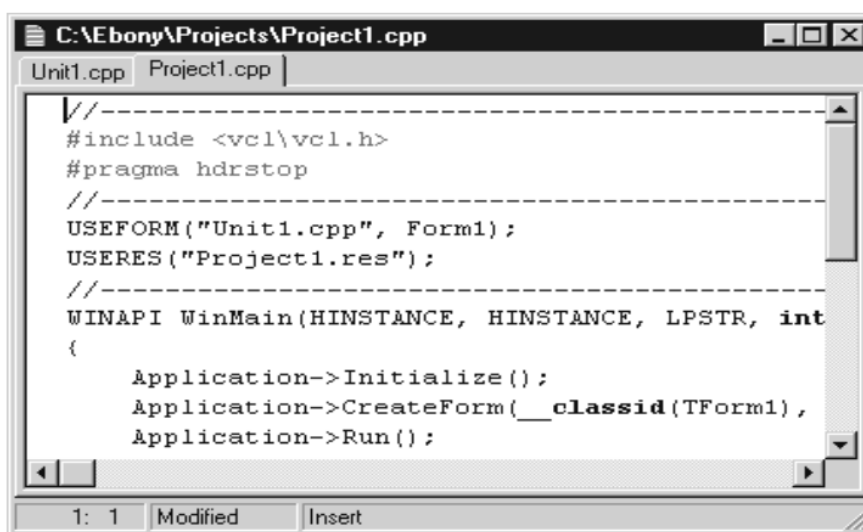
C++ Builder dastlabki Project1.cpp nomli loyiha fayli va Project1.mak nomli **make** fayl yaratadi. Loyihaga yangi forma qo'shish kabi o'zgartirish kiritilganda C++ Builder loyiha faylini yangilaydi.

Odatda loyiha yoki dastur bir necha formaga ega bo'ladi. Loyihaga forma qo'shish quyidagi qo'shimcha fayllarni yaratadi:

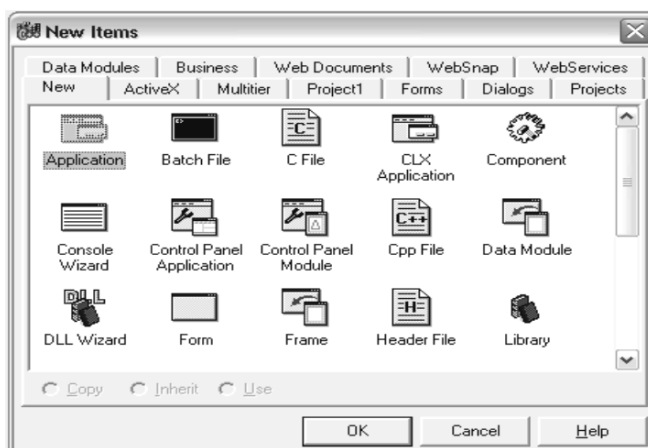
- *.DFM kengaytmali forma fayli formani konstruksiyalash uchun oyna resurslari to'g'risidagi axborotni o'z ichiga oladi.
- *.CPP kengaytmali modul fayli C++ dagi kodni o'z ichiga oladi.

*.H kengaytmali sarlavha fayli forma klassi tavsifini o'z ichiga oladi. Yangi forma qo'shganingizda, loyiha fayli avtomatik ranishda yangilanadi.

Loyihaga yangi forma qo'shish uchun, File/New Form menyu punktini tanlang. Loyihaga qo'shiladigan bo'sh forma paydo bo'ladi. Bundan tashqari File/New menyu punktidan foydalanib, Forms sahifasini belgilash kerak hamda ob'yektlar ro'yhatidan mos shablonni tanlash kerak. Joriy loyihani faqat kompilyatsiya qilishgina kerak bo'lsa, Compile menyusidan Compile punktini tanlash kerak. Loyihani kompilyatsiya qilish va joriy loyiha uchun bajariladigan fayl yaratish uchun, Run menyusidan Run punktini tanlash kerak.



2.1.5-rasm. Loyiha fayli

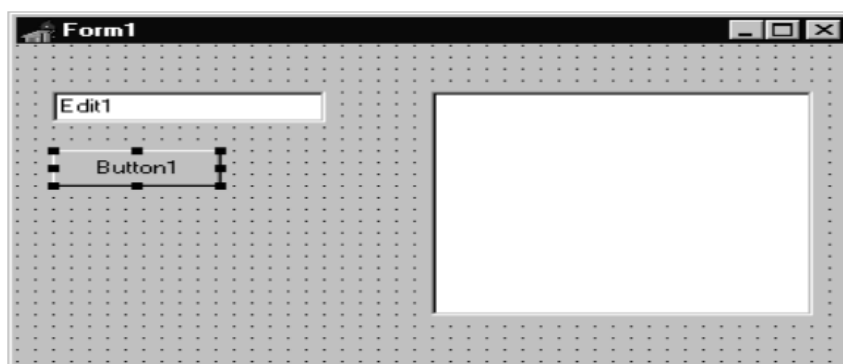


2.1.6-rasm. Formalar shabloni.

Agar dastur ishga tushganida xato paydo bo'lsa, C++ Builder dastur ishlashni to'xtatadi va xato manbai bo'lgan qatorni kursor bilan belgilab qo'yadi. Kerakli to'g'rilash ishlarini olib borishdan avval dasturni qayta ishga tushirib yuborish kerak. Buning uchun Run menyusidan Run punktini tanlash va dasturni yopish kerak. Keyin esa loyihaga o'zgartirish kiritish kerak bo'ladi. Shunda Windows resurslarini yo'qotish ehtimoli kamayadi.

Oddiy ilova dasturini yaratish. Endi tahrirlovchi maydonga matnni kiritib, uni tugma bosilishi bilan ro'yhatga qo'shadigan oddiy bir ilova dasturini yaratishga harakat qilamiz. Demak, yangi loyiha yaratish uchun File/new Application menyu punktini tanlaymiz, uning asosiy formasini misol1.cpp nom bilan va loyihani esa misol1.mak nom bilan saqlab qo'yamiz. Formaga esa Button, Edit va ListBox komponentlarini palitradan olib qo'yamiz. Shundan keyin formadagi Edit komponentini tanlab, Text tarkibidagi joriy matnni o'chirib yuboramiz. Keyin Button1 uchun Caption tarkibini sozlaymiz. —Qo'shish tugmasiga OnClick hodisasini qo'shish uchun formadagi o'sha tugmani belgilab, ob'yektlar inspektoridagi hodisalar sahifasini ochish kerak va ro'yhat o'ng tomonidagi OnClick hodisasini ikki marta bosish kerak. So'ng OnClick ga mos kiritish satrida funksiya nomi paydo bo'ladi. C++ Builder hodisa prototipini paydo qiladi va uni kod tahrirlagichda ko'satadi. Shundan keyin, funksiya {} tanasiga kodni kiritish kerak bo'ladi.

Dasturni kompilyasiya qilish uchun Run menyusidan Run punktini tanlaymiz. Endi taxrir maydoniga biror matnni kiritib, qo'shish tugmasini bosish kerak va kiritilgan matn ro'yhatga qo'shilganligiga guvoh bo'lamiz.

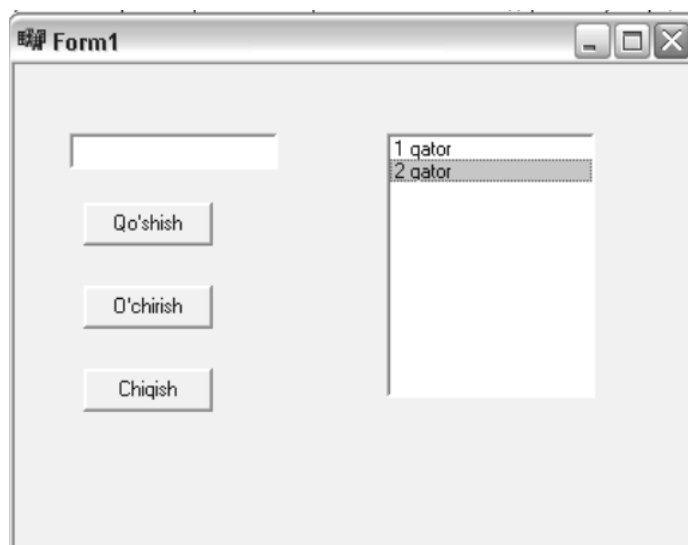


2.1.7-rasm. Formaga komponentlarni joylashtirish



2.1.8-rasm. Tayyor ilova dastur ko'rinishi

Endi esa, dasturimizga “Chiqish” va “O’chirish” tugmalarini qo’shib uni shakllantiramiz. Buning uchun ikkita tugma qo’shib, ularning Caption tarkibini o’zgartiramiz va o’sha tugmalarga bog‘liq bo’lgan hodisalarga jarayon yaratamiz.



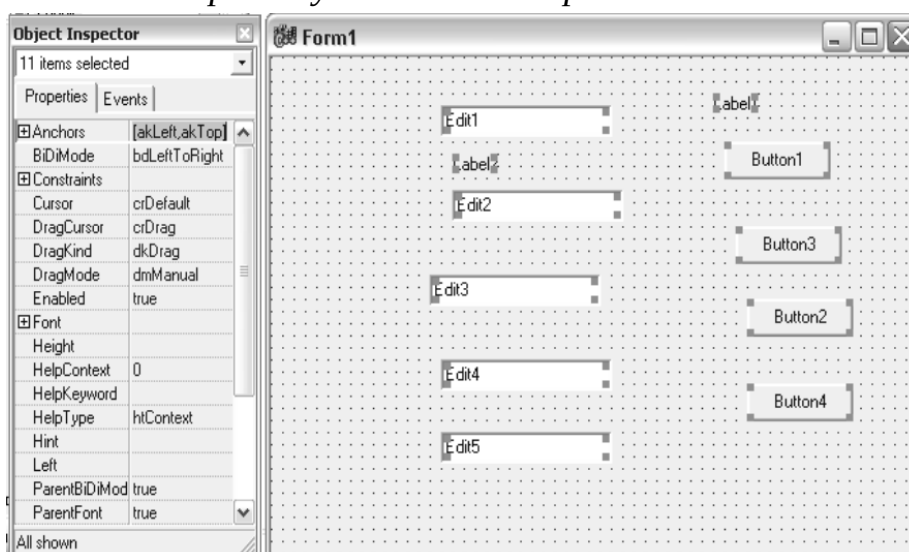
2.1.9-rasm. Shakllantirilgan ilova dastur

C++ Builder komponentlarini o’rganish. C++ Builder dasturida effektiv foydalanuvchi interfeysini ishlab chiqish katta ahamiyat kasb etadi. Komponentlarni manipulyatsiya qilishning ko’p operatsiyalari Edit menyusida joylashgan. Bu menyuning ko’plab opsiylariga formadagi tarkibi o’zgartirilishi kerak bo’lgan komponentlar tanlangandagina murojaat qilish mumkin.

Bitta komponentni tanlash quyidagi usullar bilan amalga oshiriladi:

- Shift tugmasini bosib turgan holda, har bir komponentni sichqoncha tugmasi bilan bosish;
- Sichqoncha chap tugmasini bosib, kerakli komponentlarni aylantirib belgilash;

Guruhli operatsiyalar uchun komponentlarni tanlash

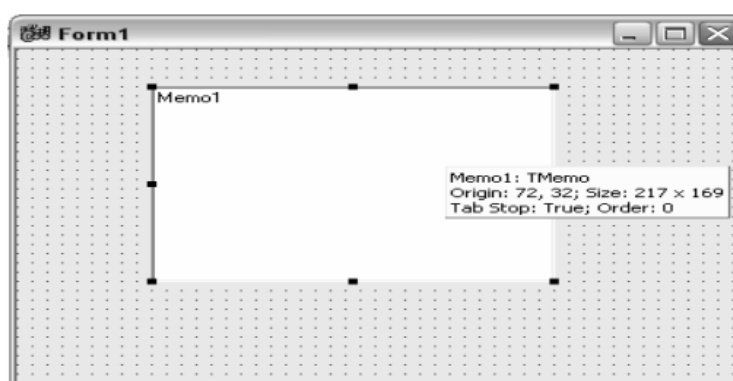


2.1.10-raasm. Guruhli operatsiyalar uchun bir necha komponentlarni belgilash

Yuqoridagi rasm Font va Left tarkiblari o'zgartirish natijasini ko'rsatib turmoqda. Barcha belgilangan komponentlar bir xil o'zgartirilgan tarkiblarga ega bo'ldi.

Komponentlar o'lchovlarini o'zgartirish. Komponentni qo'shish uchun komponentlar palitrasini tanlab olish kerak. Keyin esa sichqoncha kursori forma ichiga olib kelib, sichqoncha chap tugmasini bosish kerak. Natijada komponent chegaralarini aks ettiruvchi to'g'ri to'rtburchak forma ichida paydo bo'ladi. To'g'ri to'rtburchak kerakli o'lchovga ega bo'lganidan keyin sichqoncha tugmasini qo'yib yuborish kerak.

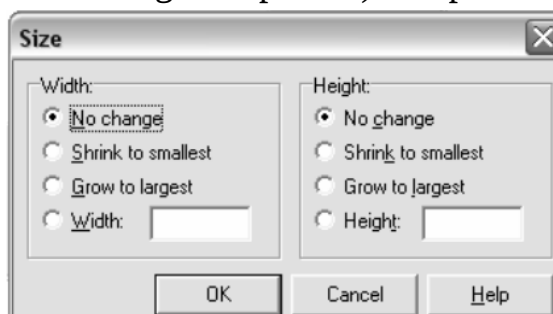
Sichqoncha ko'rsatkichini komponent atrofidagi kichkina qora to'rtburchak nuqtalarga olib kelinsa, sichqoncha kursori ko'rinishi o'zgaradi. Sichqonchani bosgan holda ushbu kursorni komponent o'zgarishi bilan birga jildirib uning o'lchovini o'zgartirish mumkin.



2.1.11-rasm. Komponentni formaga qo'shishda uning o'lchovini o'zgartirish

Bir necha komponentlarning o'lchovini o'zgartirish uchun ularning hammasi belgilanadi. Keyin esa Edit/Size menyu punktini tanlash kerak. Size oynasi paydo bo'ladi hamda o'lchov opsiyasi tanlanadi. Komponent o'lchovini piksellarda aniq o'rnatish uchun Width va Height maydonlariga kerakli raqamlarni kiritish kerak. Keyin esa OK tugmasini bosish kerak.

Bir xil turdagi komponentlarni formaga qo'shish uchun Shift tugmasini bosgan holda komponentni palitradan tanlash kerak. Bu holda komponent atrofida to'rtburchak paydo bo'ladi. Shundan keyin sichqoncha tugmasining formaga har bir bosilishi unda komponentlar paydo bo'lishiga olib keladi. Komponentni ko'paytirish jarayonini yakunlagach sichqoncha bilan komponentlar palitrasidagi birinchi (ko'rsatkich ko'rinishidagi komponent) komponentni bosish kerak.



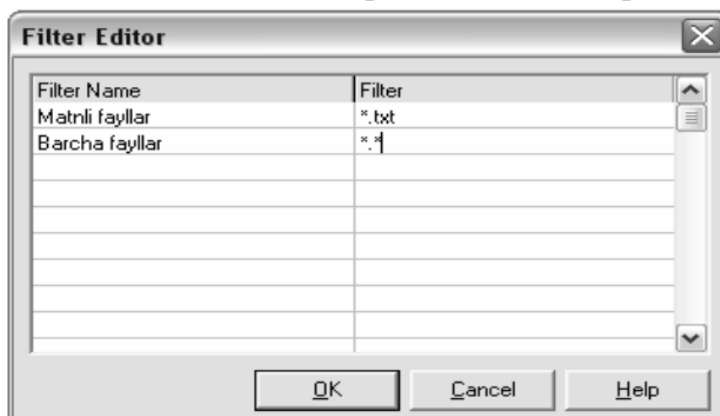
2.1.12-rasm. Edit/Size dan foydalangan holda komponent tarkibini sozlash.

Dastur formalarini loyihalash. Olingan bilimlarni yangi fayl yaratadigan, uni ochadigan, tahrir qiladigan va saqlaydigan hamda matnlar bilan ishlaydigan matn muharriri yaratishga harakat qilamiz. Buning uchun yangi formaga asoslangan yangi loyiha yaratamiz va uni Tahrir.mak deb nomlaymiz. Loyihaga esa Tahrir.cpp deb nom beramiz.

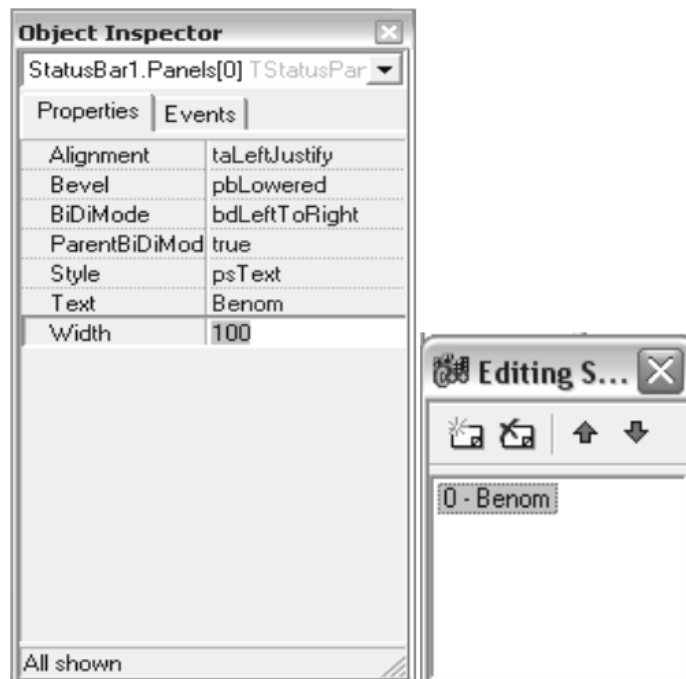
Bo'sh formaga muharrirning kelgusida asboblari paneli bo'ladigan TPanel komponentini qo'yamiz. Olingan Panel1 komponentining Align tarkibi qiymatini alTop qilib belgilatmiz va Caption tarkibi qiymatini o'chirib yuboramiz. Keyin esa formaga TMemo komponentini joylashtiramiz. Uning Align tarkibi qiymatini alClient, ScrollBars tarkibi qiymatini ssVertical qilib belgilab, Lines tarkibi qiymatini bo'sh qoldiramiz. Kelgusi matn muharririmiz fayllarni ochish va saqlash kerakligini unutmashligimiz kerak. Shuning uchun comdlg32.dll kutubxonasida mavjud Windows standart dialogidan foydalanamiz. Buni amalga oshirish uchun komponentlar palitrasidagi Dialogs sahifasidan TOpenDialog va TSaveDialog komponentlarini joylashtiramiz. Yaratilgan OpenFileDialog komponentining Filter tarkibini o'zgartiramiz: Filter Editor muloqot paneliga ikki satr kiritamiz va OK tugmasini bosamiz.

Endi Filter tarkibi to'g'risida joylashgan qiymatlar ro'yhatidagi satr almashish buferidagi qiymatni olamiz. SaveDialog1 komponentini tanlab, almashish buferidagi qiymatni Filter tarkibi to'g'risidagi qatorga qo'yamiz. Bu yo'l bilan biz ikkinchi dialog uchun Filter tarkibi qiymatini o'rnatamiz. Ixtiyoriy ravishda boshqa parametrlarni ham sozlash mumkin.

Endi esa formaga StatusBar komponentini joylashtiramiz. Bu tarkib tahrirlagichi ekranga muloqot oyna bo'lib chiqadi. Tahrirlanadigan fayl nomi chiqadigan panelni sozlaymiz. Buning uchun New tugmasini bosamiz va yaratilgan panelning Width parametrini 100 ga teng qilib o'zgartiramiz. Keyin sichqoncha bilan Panel1 komponentini tanlaymiz va unga TSpeedButton tipdagi komponentdan 9 ta joylaymiz. Buni osonroq amalga oshirish uchun esa, Shift tugmasini bosib, Additional sahifasidan Speed Button komponentini tanlaymiz.

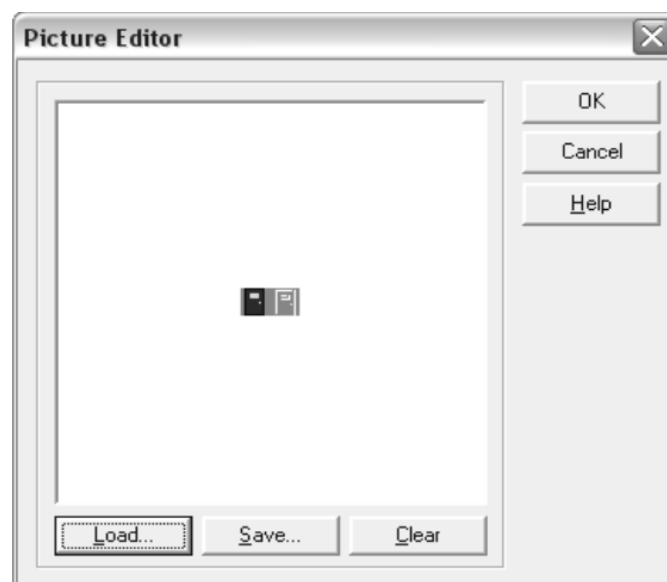


2.1.13-rasm. OpenFileDialog1 komponenti Filter tarkibini sozlash



2.1.14-rasm.StatusBar1 komponenti Panels tarkibini sozlash

Tugmalarimizni rasmlar bilan jihozlaymiz. Buning uchun bu tugmalarning Glyph takibini o'zgartiramiz. Bu bilan biz C++ Builder tarkibiga kiruvchi ko'plab rasmlar to'plamidan foydalanishimiz mumkin. Bizning misolimiz uchun ushbu katalogdan Doorshut.bmp, Filenew.bmp, Fileopen.bmp, Filesave.bmp, Cut.bmp, Copy.bmp, Paste.bmp, Help.bmp fayllari tanlangan.



2.1.15-rasm.SpeedButton1 ... SpeedButton9 komponentlari Glyph tarkibini sozlash.

Endi quyidagi rasmga qarab, formadagi tugmalarni tartibli joylashtiramiz. Bu tugmalar ShowHint tarkibi qiymatini True ga aylantiramiz, Hint tarkibi qiymatiga esa "Chiqish", "Yaratish", "Ochish", "Saqlash", "...nom bilan saqlash", "Qirqib olish", "Nusha olish", "Qo'yish", "Dastur haqida" so'zlarini yozamiz. Bu

sichqoncha ko'rsatkichini tugmalarga olib kelganda, izohli sariq yorliqlar paydo bo'lishiga olib keladi.

Menyu yaratish. Endi muharririmiz uchun menyu yaratamiz. Buning uchun standart sahifasidan TMainMenu komponentini olib asosiy formaga qo'yamiz. Sichqoncha o'ng tugmasini bosib kontekst menyudan Menu Designer punktini tanlaymiz. Klaviaturadagi ko'rsatkich (“←”, “↑”, “→”, “↓”) tugmalarini bosib menyuning boshqa boshqa punktlarni yaratamiz. Ushbu menyular quyidagilardir: “Fayl” (“Yaratish”, “Ochish”, “Saqlash”, “...nom bilan saqlash”, “-”, “Chiqish”) punktlari bilan), “Tahrir qilish” (“Qirqib olish”, “Nusxa olish”, “Qo'yish” punktlari bilan) va “?”-“Dastur haqida” punkti bilan.

Endi ob'yektlar inspektoridan hodisalar sahifasini tanlab avval yaratilgan SpeedButton1ClickSpeedButton9Click funksiyalarini menyu punktlaridan nomlari mos kelganlarini bir-biriga bog'laymiz. Bizda foydalanilmagan “Asboblari paneli” menyu punkti qoldi. Ushbu menyu punktining Checked tarkib qiymatini True qilib qoyamiz. “Asboblari paneli” menyu punkti uchun quyidagi OnClick hodisa jarayonini yaratamiz:

Va nihoyat, ilova dastur asosiy formasi elementlari uchun kontekst menyu yaratamiz. Buning uchun formaga ikki TPopupMenu komponentidan qo'yamiz. Birinchisini “Qirqib olish”, “Nusxa olish”, “Qo'yish” punktlari bilan, ikkinchisini esa “Berkitish” punkti bilan yaratamiz. Ushbu menyu punktlar menyular uchun mavjud funksiyalardan mosini tanlab OnClick hodisa jarayonlarini o'rnatamiz.



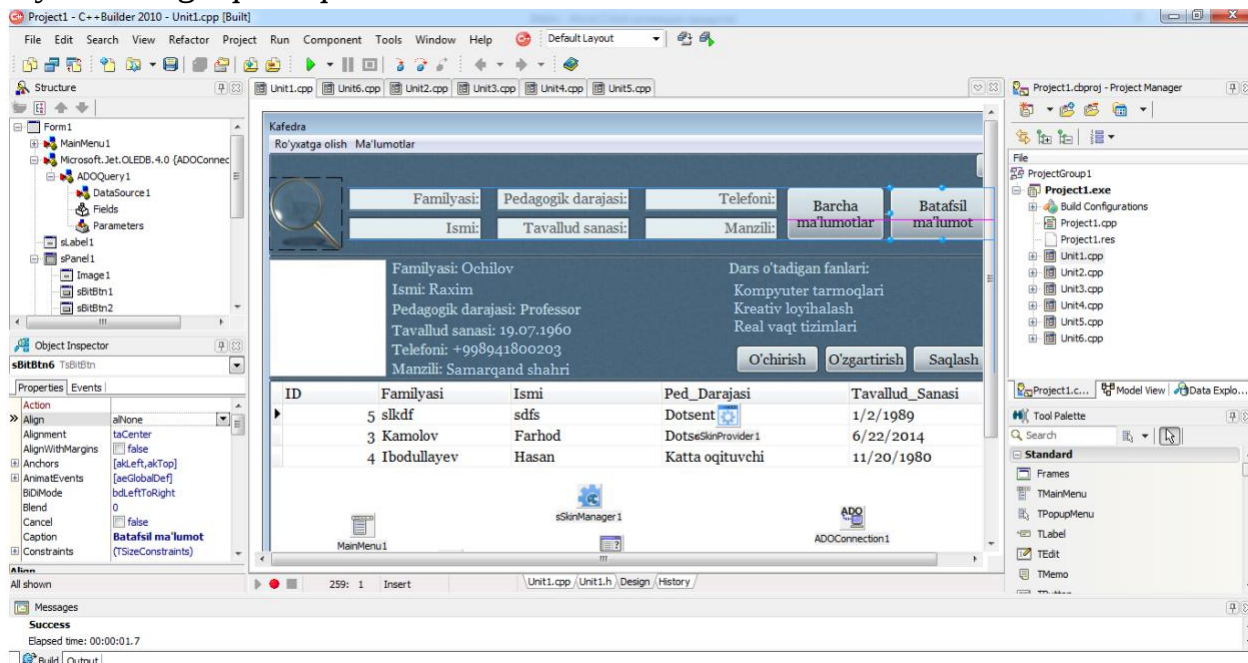
2.1.17-rasm. Menu Designer yordamida menyu yaratish.

2.2. Kafedrani faoliyati monitoringini olib boruvchi dasturiy platformasining yaratilish jarayoni

C++ Builder da dasturlar qanday yaratilishi va uning komponentalari

xususiidagi ma'lumotlarga ega bo'dik. Endi dasturimizning yaratilish jarayoni ko'rib chiqamiz.

Dastlab asosiy oynamizni tashqi ko'rinishini chizib olamiz. Bunda kerakli komponentalarni form1 ga joylaymiz. Asosiy oyna foydalanuvchini zeriktirmaydigan bo'lishi uchun men dizayner do'stlarim bilan maslahatlashib ish ko'rdim. Ular menga kerakli maslahatlarni berganlaridan keyin "skin" dan foydalanishga qaror qildim.



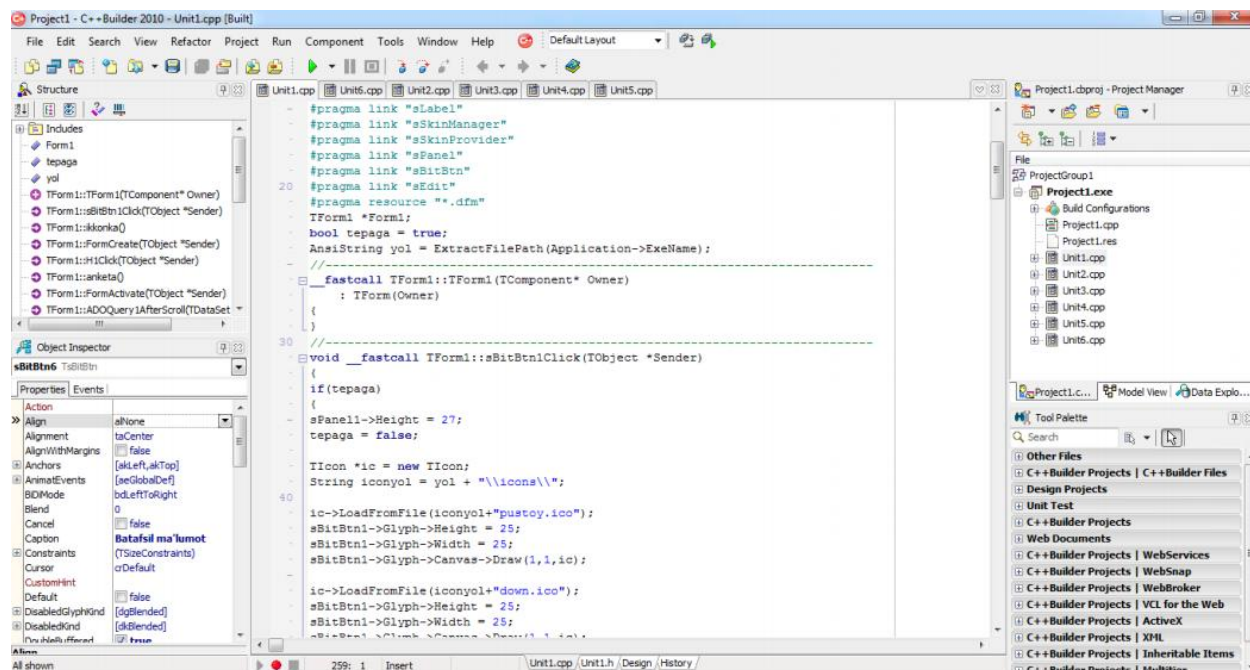
2.2.1-rasm.Asosiy oynamizning ko'rinishi.

C++ Builder da ko'rinadigan va ko'rinmaydigan komponentalar mavjud. Masalan, edit, label, button va boshqalar ko'rinadigan komponentalar. ADOConnection, skinProvider, skinManager va boshqalar ko'rinmaydi. Kafedraning asosiy oyna ko'rinishida dastur ishchi ko'rinishida yuqorida sanab o'tilgan komponentalar mavjud.

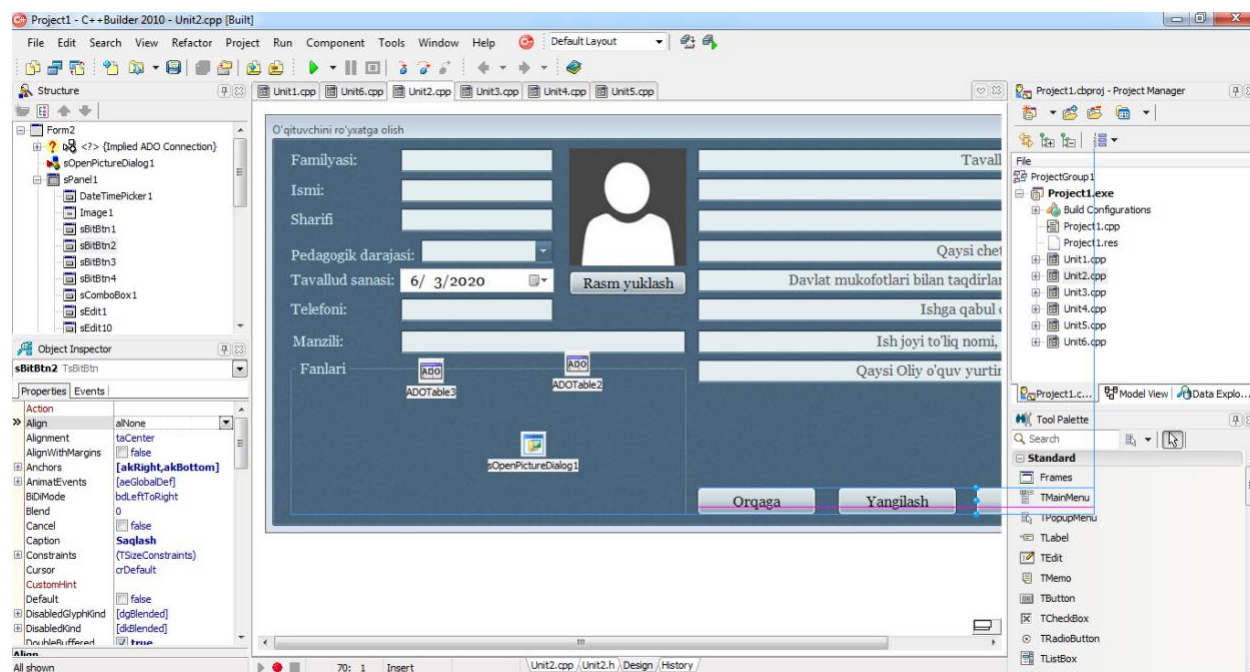
- ADOConnection – ma'lumotlar bazasi bilan ulanish uchun xizmat qiladi.
- ADOQuery – so'rovlar asosida ma'lumotlarni chaqirib beradi.

Kerakli komponentalarni form1 ga joylab olganimizdan keyin ular nima ish qilishi kerakligini unit.cpp qismiga yozamiz.

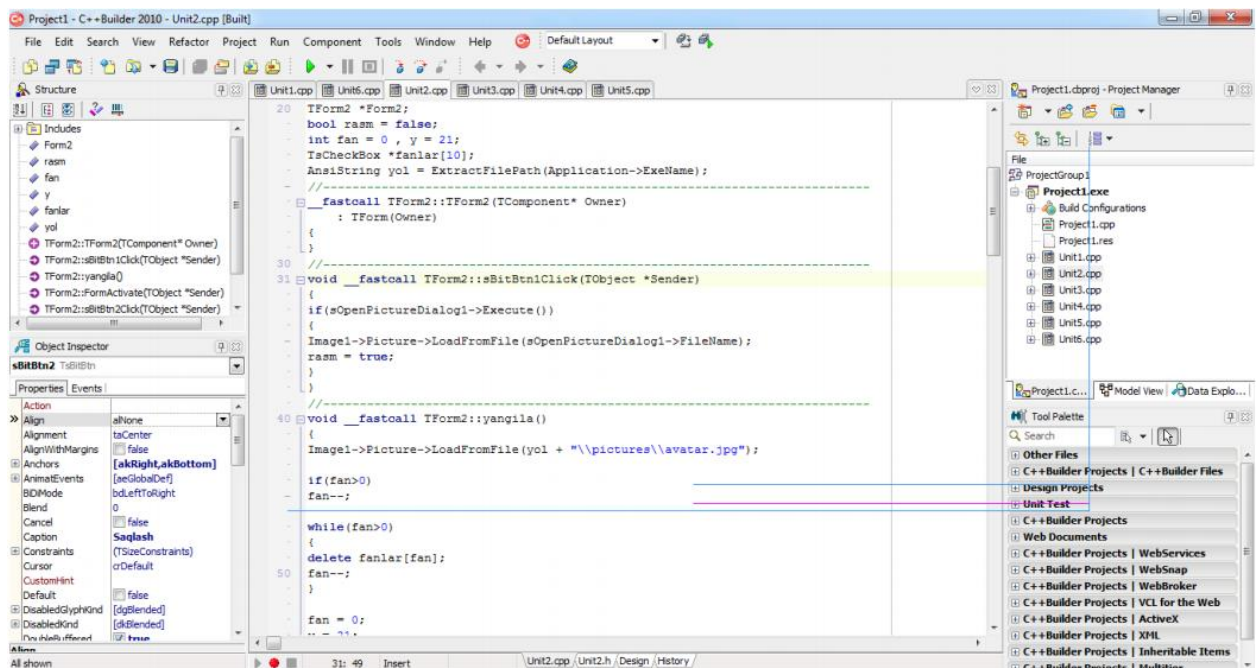
Keyingi oyna "O'qituvchilarni ro'yhatga olish" deb nomlanadi. Unda biz yabgi kelgan o'qituvchilar haqidagi ma'lumotlarni qulay interfeys oynasi yordamida bazga qo'shib qo'yamiz.



2.2.2-rasm. Asosiy oynamizning kod ko'rinishi.

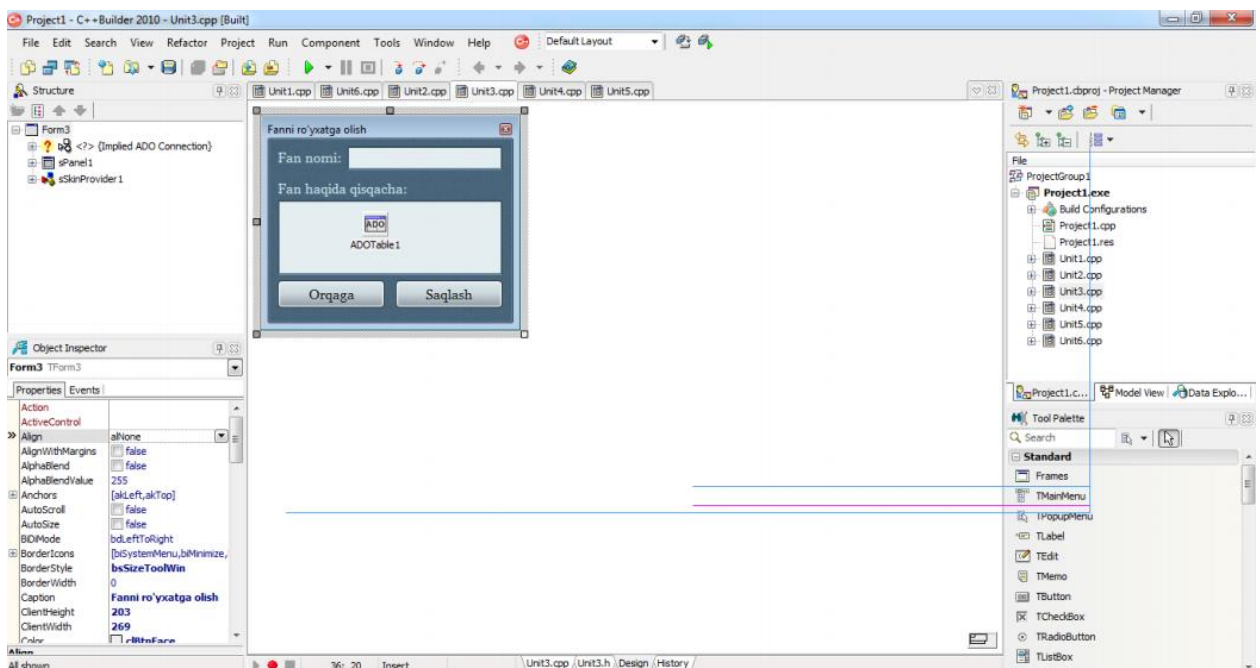


2.2.3-rasm. Ro'yhatga olish oynasi



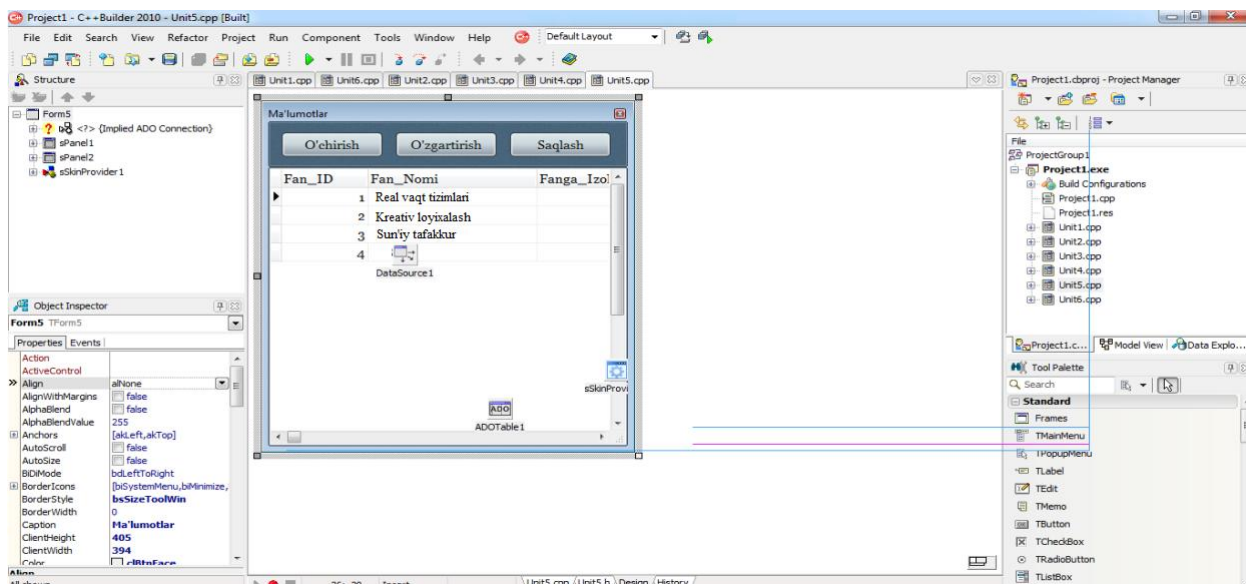
2.4-rasm. Ro'yhatga olish oynasining kod ko'rinishi.

Keyingi oyna "Fanni ro'yhatga olish" deb nomlanadi. Bu nima uchun qilingan degan savol tug'iladi. Kafedraga yangi o'qituvchi kelishidan tashqari yangi fanlar qo'shilishi yoki joriy fanning nomi o'zgarishi mumkin. Manu shunday holatlarda kerakli hisoblanadi.



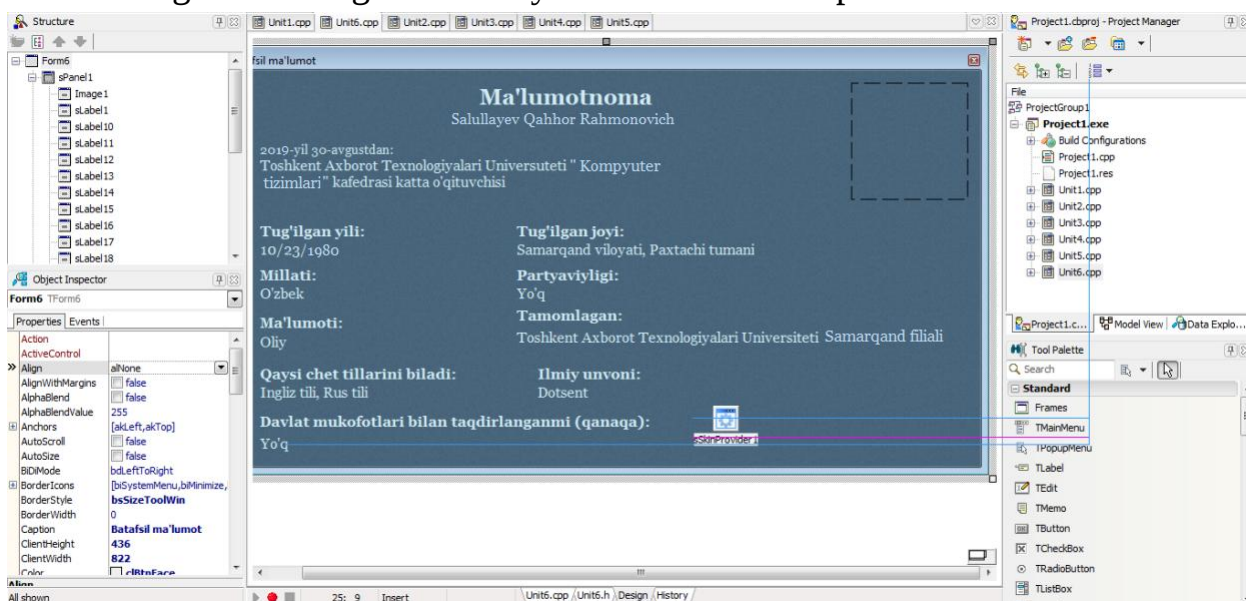
2.2.5-rasm. Fanni ro'yhatga olish oynasi.

Endi keyingi muloqot oynamizning yaratilish jarayoniga o'tsak. Ushbu oyna "Ma'lumotlar" deb nomlanadi. U asosan fanlar bilan ishlash uchun mo'ljallangan. Bunda biz o'tilmayotgan fanlar bo'lsa, ularni o'chirishimiz, o'zgartirishimiz hamda o'zgartirishlarni saqlashimiz mumkin. Har bir o'zgartirishlar bazaga avtomatik saqlanadi.



2.2.7-rasm. Ma'lumotlar oynasi.

Ma'lumotlarni foydalanuvchi uchun qulay holatda taqdim etishga "Batafsil ma'lumotlar" degan muloqot oynasini yaratdim. Undagi ma'lumotlar chop etish uchun tayyor holatda ko'rsatilgan. Oyna ustiga ikki marta bosish orqali ma'lumotlarni chop etishga berib yuborish mumkin. Bunda dialog oynasi ochiladi va "Yes" tugmasini bosganizdan keyin ma'lumotlar chop etiladi.



2.2.9-rasm. Ma'lumotlar oynasi ko'rinishi

2.3. Kafedrani faoliyati monitoringini olib boruvchi dasturiy platformasining interfeyslari va foydalanish ko'rsatmalari

Men yaratgan dastur kafedra ishini avtomatlashtirish uchun xizmat qiladi. Unda olti xil ma'lumot tiplari bo'yicha qidiruv tizimini amalga oshirganman. Unga ikkita menyu joylashtirdim:

- Ro'yhatga olish

- Ma'lumotlar

Har bir menyuga mos menyu ostlari mavjud. Ro'yhatga olish menyusi menyu ostlari:

- O'qituvchini

- Pedagogik darajani

- Fanlarni

“Kafedra” muloqot oynasidan foydalanib bazadagi ma'lumotlarga o'zgartirish va qo'shimchalar kiritishimiz mumkin.

The screenshot shows a web application window titled 'Kafedra'. It has two tabs: 'Ro'yhatga olish' (selected) and 'Ma'lumotlar'. The main area contains a form with fields for 'Familyasi', 'Pedagogik darajasi', 'Telefoni', 'Ismi', 'Tavallud sanasi', and 'Manzili'. There are also buttons for 'Barcha ma'lumotlar' and 'Batafsil ma'lumot'. Below the form, there is a section for 'Dars o'tadigan fanlari' with a list of subjects: 'MOBIL TIZIMLAR' and 'KREATIV LOYIXALASH'. At the bottom, there is a table with columns: ID, Familyasi, Ismi, Ped_Darajasi, and Tavallud_Sanasi. The table contains three rows of data.

ID	Familyasi	Ismi	Ped_Darajasi	Tavallud_Sanasi
7	Pozilov	Mirsodiq	Dotsent	08.05.1955
10	Tashpulatov	Muhiddin	Katta oqituvchi	25.12.1979
11	Sadaddinova	Sanobar	Dotsent	10.03.1974

2.3.1-rasm. Kafedra oynasi umumiy ko'rinishi

“Batafsil ma'lumotlar” da barcha ma'lumotlar foydalanuvchini zeriktiradigan jadval ko'rinishida emas, balki qulay interfeysda. Oyna ustiga sichqoncha tugmasini ikki marta bosish orqali ma'lumotlarni chop etishga berish mumkin.

“O'qituvchini ro'yhatga olish” oynasida biz o'qituvchilarni to'liq ma'lumotlari bilan ro'yhatdan o'tkazamiz. O'qituvchilarni ro'yhatdan o'tkazish bo'yicha turli talablar qo'yilgan (Familyasi, ismi, sharifi va boshqalar). Bundan tashqari ro'yhatdan o'tayotgan shaxsning suratini ham yuklash mumkin. Qaysi fandan dars berishini belgilash uchun *Checkbox*dan foydalandim. Chunki bitta o'qituvchi bittadan ortiq fandan dars berishi mumkin.

2.3.4-rasm. O'qituvchini ro'yhatga olish

“Pedagogik darajani ro'yhatga olish” da pedagogik daraja kiritiladi va unga izoh beriladi.

2.3.5-rasm. Pedagogik darajani ro'yhatga olish.

2.4. Kompyuter xonalarida, ish joyiga qo'yiladigan ergonomik talablar

Kompyuterni xonada to'g'ri joylashtirish va to'g'ri loyixalanib, o'rnatilgan yoritgichlar foydalanuvchini ko'rishini yaxshi taminlaydi, asab tizimiga qo'shimcha zo'riqish bermaydi, operatorni normallashi faoliyatini taminlaydi, ish jarayonidagi xatolarni keskin kamaytiradi. Kompyuterlarni alohida xonalarga 5-6 displaydan ortiq bo'lmagan xolda joylashtirish tavsiya qilinadi. Bu eng avvalo mikroiklimni yo'l qo'yilgan qiymatlari parametrlarini taminlashga imkon beradi. Sanitar normalarga muvofiq bitta foydalanuvchi uchun 5m² maydon, hajmi

20m³ dan kam bo'lmisligi kerak. Gigienik nuqtai nuqtai nazardan kompyuterni shunday joylashtirish kerakki, ekrandan ko'zni ko'targanda, xonadagi eng uzoqda joylashgan narsa ham ko'rsin. Operatorning ish joyini kirish eshigiga yuzi qaragan holda joylashtirish eng samarali hisoblanadi. Eng uzoq masofaga nigohni o'tkazish imkoni kompyuterda ishlagandagi ko'rish tizimining og'irligini kamaytirishni eng samarali usuli hisoblanadi. Ish joylarini kompyuterdan devorgacha bo'lgan masofa 1m dan kam bo'lmagan holda xonaning burchaklariga yoki devorga qaratib joylashtirish, derazadan tushgan yorug'lik ko'z uchun ortiqcha zo'riqish bo'lmisligiga yordam beradi. Shuning uchun ham kompyuterni derazaga qaratib joylashtirmaslik darkor. Agar bir xonada bir necha komp'yuterlar joylashgan bo'lsa, elektromagnit nurlarning tasirini kamaytirish uchun bir manitor ekranidan ikkinchisining orqa devorigacha masofa 2 m dan kam bo'lmisligi kerak. Sanitar qoidalariga muvofiq shaxsiy kompyuterlar joylashgan xonada aralash yoritilganlik, ya'ni tabiiy va sun'iy bo'lishi kerak. Tabiiy yoritilganlik iloji boricha shimolga va shimoliy sharqqa yo'naltirilgan bo'lishi, imkoni bo'lmasa, jadal quyosh nuri janubiy va g'arbiy derazalardan yaltillashlar yuzaga keltirmisligi va ishlashga xalaqit qilmasligi uchun derazalarni pardalar, jalyuzlar yoki tashqi to'sqichlar bilan taminlash kerak. Ish joyi derazaga nisbatan yonlanmasiga joylashgan bo'lib, tabiiy yorug'lik chap tarafdin tushishi maqsadga muvofiq. Kompyuterlar shunday joylashishi kerakki, yoniq ekran boshqa operatorning ko'rish maydoniga tushmasligi, ekranda tabiiy va suniy yoritilganlikning aksidan yaltillashlar bo'lmisligi kerak.

Ko'rish sharoitini baxolash uchun yaltillaganlik tushunchasi kiritiladi. Yaltillaganlik-ko'rish funksiyasini buzilishiga olib keluvchi yaltillagan yuzlarning kuchaygan yorug'ligi bo'lib, obektni ko'rishni yomonlashtiradi. Yaltillamaslikning birligi-kg/m² . 30 ming kg/m² ga teng yorug'lik ko'zni ko'r qiladi. Yaltillaganlik haddan tashqari asabiylashuvini yuzaga keltiradi. Shuning uchun sanitar qoidalar yorug'lik keltiradi. Shuning uchun sanitar qoidalar yorug'lik manbaidan to'g'ridan-to'g'ri paydo bo'ladigan yaltillaganlikni chegaralaydi. Deraza, yoritgichlardan tushgan yorug'lik ko'rish maydonida 200 kg/m² dan oshmasligi kerak. Ekran, stol, klaviatura kabi ish yuzasidan qaytgan nurlardan hosil bo'ladigan yaltillaganliklarni ham chegaralash kerak. Bu yoritgichlarni to'g'ri tanlab, ish o'rinlarini tabiiy va suniy yoritgichlarga nisbatan to'g'ri joylashtirish hisobiga amalga oshadi, bunda yaltillashlarning yorug'ligi displey ekranida 40 kg/m² dan oshmasligi kerak.

Mutaxassislarining tavsiyasiga ko'ra devorlar, mebellar och sut rangda, shipdan nur qaytarish koeffitsienti 0.7- 0.8, devordan va poldan 0.6 va 0.3 bo'lishi kerak. Bunga shipni oq ranga, devorlarni och sariq va qizg'ish ranga bo'yash natijasida erishish mumkin. Umumiy yoritish uchun lyuminessent lampalar

ishlatilishi natijasida ulardan yorug'lik oqimi kuchlanishining o'zgarishiga qattiq bog'liq bo'lganligi sababli yoritilganlikning tebranishi yuzaga keladi, bu o'z yo'lida ko'zni har safar adaptatsiya qilishiga, toliqishiga olib keladi. Shuning uchun mahalliy va umumiy yoritgichlar sifatida yuqori chastotali, yonishini nazorat qiluvchi uskunali gazorazryadli lampalar ishlatilishi kerak. To'g'ri tanlangan, yani eng kamida Shvesiya o'lchovlar va sinovlar Milliy komiteti tomonidan qabul qilingan MRK(II) talablariga javob beradigan va kerakli sertifikat bo'lgan kompyuterlarda ishlaganda foydalanuvchi sog'lig'ini saqlash maqsadida quyidagi qiyin bo'lmagan qoidalarga rioya qilish kerak:

- ish joyi qulay bo'lishi va tayanch-harakat apparatini hamda qon almashishini normal ishlashini taminlash kerak;

- kun davomida videoterminalda umumiy ishlash davomiyligi 4 soatdan oshmasligi, videoterminalda uzluksiz ishlash 1.5-2 soatdan ko'p bo'lmasligi, har bir soat ishdan so'ng kamida 10-15 minut tanaffuz qilish, shu paytda o'rindan turib, ko'z, bel, qo'l va oyoq uchun maxsus mashqlar qilish kerak.

- normal ko'rish qobiliyatida ko'z ekrandan qo'l cho'zganchalik (yani 60-70 smdan kam bo'lmagan) masofada bo'lishi va yiliga kamida bir marotaba ko'z vrachiga tekshirtirib turish kerak;

- bir soat mobaynida 10 mingdan ortiq klavishni bosish kerak emas;

- manitor ekranida yaltillashlar paydo bo'lishiga yo'q qo'yilmasligi kerak;

- homilador ayollarning kompyuterda ishlashiga ruxsat berilmaydi.

Elektr toki, elektromagnit nurlanishlarning organizmga ta'siri va ularda himoyalash. Hozirgi zamon elektr qurilmalari, shu jumladan eng zamonaviy kompyuterlar ham sanoat chastotasidagi 50Gs li elektr toki bilan ishlaydi. Bunday toki 0.6-1.5 mA miqdoridagi tok o'tsa muskullar tartibsiz qisqarib, inson o'z azolari qismlarini boshqarish qobiliyatidan mahrum bo'ladi, ya'ni elektr simini ushlab turgan bo'lsa, panjalarini yecha olmaydi. Bunday tok chegara miqdoridagi ushlab qoluvchi tok deyiladi. Agar tok miqdori 25-50 mA ga yetsa, unda tok tasir ko'rsatadi, buning natijasida nafas olish qiyinlashadi. Ta'sir qiluvchi tok miqdori 100mA dan ortiq bo'lsa, u yurak muskullariga ta'sir ko'rsatadi va yurakning ishlash tartibi buziladi, natijada qon aylanishi tizimi butunlay ishdan chiqadi va bu holat o'limiga olib keladi. Eng zararli tok chastotasi 20 Gs dan kam toklarning tasir darajasi kamayadi. Katta chastotadagi elektr toklarida tok urish bo'lmaydi, lekin kuyishi mumkin. Elektr kuchlanish qancha yuqoiri bo'lsa, u shuncha xavfli hisoblanadi.

Elektr qurilmalar, elektron hisoblash mashinalari va kompyuter bilan ishlashda, ularning tok o'tkazuvchi qismlarining izolatsiya qobig'i yemirilishi oqibatida, ularning korpusida elektr kuchlanganligi paydo bo'lishi mumkin. Shuning uchun ular qaysi joyda va qanday binoda ishlatilashidan qa'iy nazar,

ularning korpusida elektr kuchlanganligi paydo bo'lishi mumkin. Shuning uchun ular qaysi joyda va qanday binoda ishlatilishidan qat'iy nazar, ularning metall korpuslarini yerga ulab muhofazalanadi. Yerga ulab muhofaza qilish qurilmasining umumiy qarshiligi, yilning hamma fasllari uchun, 1000 V kuchlanishiga bo'lgan elektr qurilmalarda 4 Om dan kata bo'lmasligi kerak.

Elektr qurilmalarining tok o'tmaydigan metall qismlarini oldindan nolga ulab muhofaza qilish deb yuritiladi. Agar elektr qurilmasi korpusida inson hayoti uchun xavfli kuchlanish hosil bo'lib qolsa, uni tezda o'chirish uchun muhofazalofchi avtomatik o'chirish qurilmalari o'rnatiladi. Ular elektr asbobni 0.2 c dan oshmagan vaqt davomida o'chirish imkoniyatini berish kerak.

Elektromagnit maydonning elektr kuchlanganligining eng katta miqdori:

- Elektromagnit maydon kuchlanganligining elektr tashkil etuvchisining videomanitor yuzasidan 50 sm masofadagi qiymati-10V/m;

- Elektromagnit maydon kuchlanganligining magnit tashkil etuvchisining videomonitor yuzasidan 50sm masofadagi qiymat-0.3 V/m;

- Elektr maydon kuchlanganligi talabalar uchun 15 kV/m va kata yoshdagilar uchun 20 kV/m va kata yoshdagilar uchun 20 Vt/m² dan oshmasligi kerak. Elektr qurilmalari, ShK va kompyuterlardan foydalanishda turli xildagi yonishlar xavfi doim mavjuddir. Zamonaviy kompyuterlarda elektron sxemalarning elementlarini joylashish zichligi juda yuqoridir, ulash simlari, kommunakatsion kabellar bir-biriga juda yaqin joylashgan. Ulardan tok oqqanda, kata miqdorda issiqlik ajraladi, bazi bo'limlarda harorat 80-100 S gacha ko'tarilishi mumkin. Bu ularning izolyatsiya qobig'ining erishiga, o'tkazgich qismlarining ochilib qolishiga, oqibatda qisqa tutashuv bo'lib, uchqun chiqishi va yonib ketishiga olib kelishi mumkin.

Ortiqcha issiqlikni yo'qotish uchun havoni kondensiyalash va ventilyatsiya tizimi xizmat qiladi. Lekin bu tizimlar mashinazallari va boshqa xonalar uchun qo'shimcha yong'in xavfini yuzaga keltiradi, chunki bir tarafdin, yong'in sodir bo'lganda, ularni xonalarga tezda tarqalishiga yordam beradi. Elektr qurilmalariga tok alohida yong'in xavfi bo'lgan kabel simlari orqali uzatiladi, yonuvchi izolyatsiya materialining mavjudligi, elektr uchquni va yaqinlashish qiyinligi kabel liniyalaridan yong'in yong'in chiqishi va rivojlanishi ehtimoli yanada kattaligidin dalolat beradi. Shuning uchun kabel simlari yonmaydigan materiallardan tayyorlangan, olib-qo'yiluvchi texnologik pol ostidan o'tishi kerak. Hisoblash markazlaridagi xonalarda yong'in jo'mraklari yo'lkalarga zina maydonchalariga, kirish joylariga o'rnatiladi. Olov o'chirgichlari 40-50 sm² ga bittadan o'rnatiladi. Yong'in signalizatorlari va avtomatik yong'in o'chirish qurilmalari o'rnatiladi.

XULOSA

Bitiruv malakaviy ishini bajarish bo'yicha quyidagi asosiy natijalarga erishildi:

1) Berilgan masalani hal qilishda matematik, informatik va pedagogikaga oid bilimlar yanada mustahkamlandi;

2) Yuqorida berilgan ishlar bajarilgandan so'ng, masalani dasturiy qismini yaratish uchun birinchi qadamlar tashlandi. Bunda dastur tuzish uchun kerakli algoritmlarni tanlash amalga oshirildi. Algoritmlarni tuzishga va ularni eng muqobilini tanlashga doir bilim va ko'nikmalar yana bir bor yodga tushirildi.

3) Eng maqbul algoritm tanlab olingandan so'ng, dasturni va uning ma'lumotlar bazasini yaratish uchun dasturlash tillarini tanlab olish amalga oshirildi;

4) Dasturni yaratish davomida dasturlash tilining asosiy va qoshimcha imkoniyatlaridan to'laqonli ravishda foydalanish; dasturlash tilida mavjud bo'lgan ob'yektga yo'naltirilgan dasturlash texnologiyalaridan mukammal ravishda foydalanishga erishish.

Men yuqoridagi keltirgan bandlar asosida mavjud masalani yechishni optimal variantini yechishga harakat qildim. Bunda o'zim egallagan bilimlarga, ustozlarim va yaqin o'rtoqlarimning bilim va tajribalarini o'rtoqlashgan holda kafedrafaoliyati monitoringini olib boruvchi dasturiy platformni ishlab chiqdim. Berilgan masalani yechishda obyektga yo'naltirilgan dasturlash texnologiyalaridan kengroq foydalanishga harakat qildim.

Foydalanilgan adabiyotlar

1. O'zbekiston Respublikasining "Ta'lim to'g'risida"gi qonuni // O'zbekiston Respublikasi Oliy Majlisining Axborotnomasi, 1997 y., 9-son, 225-modda.
2. "O'zbekiston Respublikasini yanada rivojlantirish bo'yicha Harakatlar strategiyasi to'g'risida"gi PQ-4947-sonli Farmoni, 07.02.2017 y.
3. "Oliy ma'lumotli mutaxassislar tayyorlash sifatini oshirishda iqtisodiyot sohalari va tarmoqlarining ishtirokini yanada kengaytirish chora-tadbirlari to'g'risida"gi PQ-3151 -sonli qarori. 27.07.2017 y.
4. Yormatov G'.Yo. Isamuxamedov Yo.U. Mehnatni muxofaza qilish. Darslik. O'zbekistan nashriyoti. Toshkent 2002.
5. H. Rahimov, T. Dehqonov: "Zamonaviy dasturlash tillari" darslik. Toshkent – 2009.
6. Полный справочник по C++ 4-е издание. Герберт Шилдт —Вильямс, 2006.
7. C++ среда разработки C++ Builder.2013 г.
8. Программирование в C++ Builder 6 — А.Я. Архангельский 2012 г.
9. C++ tilida programmalash asoslari, Madraximov Sh.F. Gaynazarov S.M. Toshkent 2009 yil.
10. <http://www.google.com>
11. <http://www.listsoft.ru>
12. <http://www.ou.tsu.ru>
13. <http://www.referat.ru>
14. <http://www.atdt.uz>

ILOVA

Dastur kodi:

```
#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
#include "Unit2.h"
#include "Unit3.h"
#include "Unit4.h"
#include "Unit5.h"
#include "Unit6.h"
//-----
#pragma package(smart_init)
#pragma link "sLabel"
#pragma link "sSkinManager"
#pragma link "sSkinProvider"
#pragma link "sPanel"
#pragma link "sBitBtn"
#pragma link "sEdit"
#pragma resource "*.dfm"
TForm1 *Form1;
bool tepaga = true;
AnsiString yol = ExtractFilePath(Application->ExeName);
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{}
//-----
void __fastcall TForm1::sBitBtn1Click(TObject *Sender)
{
if(tepaga)
{
sPanel1 ->Height = 27;
tepaga = false;
TIcon *ic = new TIcon;
String iconyol = yol + "\\icons\\";
ic->LoadFromFile(iconyol+"pustoy.ico");
sBitBtn1 ->Glyph->Height = 25;
sBitBtn1 ->Glyph->Width = 25;
sBitBtn1 ->Glyph->Canvas->Draw(1,1,ic);
```

```

ic->LoadFromFile(iconyol+"down.ico");
sBitBtn1 ->Glyph->Height = 25;
sBitBtn1 ->Glyph->Width = 25;
sBitBtn1 ->Glyph->Canvas->Draw(1,1,ic);
}
else
{
sPanel1 ->Height = 112;
tepagea = true;
TIcon *ic = new TIcon;
String iconyol = yol + "\\icons\\";
ic->LoadFromFile(iconyol+"pustoy.ico");
sBitBtn1 ->Glyph->Height = 25;
sBitBtn1 ->Glyph->Width = 25;
sBitBtn1 ->Glyph->Canvas->Draw(1,1,ic);
ic->LoadFromFile(iconyol+"up.ico");
sBitBtn1 ->Glyph->Height = 25;
sBitBtn1 ->Glyph->Width = 25;
sBitBtn1 ->Glyph->Canvas->Draw(1,1,ic);
}
}
//-----
void __fastcall TForm1::ikkonka()
{
TIcon *ic = new TIcon;
String iconyol = yol + "\\icons\\";
ic->LoadFromFile(iconyol+"up.ico");
sBitBtn1 ->Glyph->Height = 25;
sBitBtn1 ->Glyph->Width = 25;
sBitBtn1 ->Glyph->Canvas->Draw(1,1,ic);
}
//-----
void __fastcall TForm1::FormCreate(TObject *Sender)
{
ikkonka();
}
//-----
void __fastcall TForm1::H1Click(TObject *Sender)
{

```

```

Form2->ShowModal();
}
//-----
void __fastcall TForm1::anketa()
{
sLabel9->WordWrap = false;
sLabel2->Caption = "Familyasi: " + ADOQuery1 ->Fields->Fields[1]->AsString;
sLabel3->Caption = "Ismi: " + ADOQuery1 ->Fields->Fields[2]->AsString;
sLabel4->Caption = "Pedagogik darajasi: " + ADOQuery1 ->Fields->Fields[3]-
>AsString;
sLabel5->Caption = "Tavallud sanasi: " + ADOQuery1 ->Fields->Fields[4]-
>AsString;
sLabel6->Caption = "Telefoni: " + ADOQuery1 ->Fields->Fields[6]->AsString;
sLabel7->Caption = "Manzili: " + ADOQuery1 ->Fields->Fields[7]->AsString;
sLabel9->Caption = ADOQuery1 ->Fields->Fields[5]->AsString;
sLabel9->WordWrap = true;
sLabel9->Height = ((sLabel9->Width/220) + 1)*19;
sLabel9->Width = 220;
}
//-----
void __fastcall TForm1::FormActivate(TObject *Sender)
{
anketa();
}
//-----
void __fastcall TForm1::ADOQuery1AfterScroll(TDataSet *DataSet)
{
anketa();
}
//-----
void __fastcall TForm1::qidir()
{
String zapa = "Select * From Oqituvchilar WHERE Familyasi like '" + sEdit1 -
>Text + "%' AND Ismi like '" + sEdit2->Text +
"%" AND Ped_Darajasi like '" + sEdit3->Text + "%' AND Tavallud_sanasi like '" +
sEdit4->Text + "%' AND Telefoni like '" +
sEdit6->Text + "%' AND Manzili like '" + sEdit5->Text + "%' ";
ADOQuery1 ->Close();
ADOQuery1 ->SQL->Clear();
}

```

```

ADOQuery1 ->SQL->Add(zapa);
ADOQuery1 ->Open();
}
//-----
void __fastcall TForm1::sEdit1Change(TObject *Sender)
{
qidir();
}
//-----
void __fastcall TForm1::sEdit2Change(TObject *Sender)
{
qidir();
}
//-----
void __fastcall TForm1::sEdit3Change(TObject *Sender)
{
qidir();
}
//-----
void __fastcall TForm1::sEdit4Change(TObject *Sender)
{
qidir();
}
//-----
void __fastcall TForm1::sEdit6Change(TObject *Sender)
{
qidir();
}
//-----
void __fastcall TForm1::sEdit5Change(TObject *Sender)
{
qidir();
}
//-----
void __fastcall TForm1::sBitBtn2Click(TObject *Sender)
{
sEdit1 ->Text = "";
sEdit2->Text = "";
sEdit3->Text = "";
}

```

```

sEdit4->Text = "";
sEdit5->Text = "";
sEdit6->Text = "";
ADOQuery1 ->Close();
ADOQuery1 ->SQL->Clear();
ADOQuery1 ->SQL->Add("Select * From Oqituvchilar");
ADOQuery1 ->Open();
}
//-----
void __fastcall TForm1::sBitBtn3Click(TObject *Sender)
{
if( MessageDlg(ADOQuery1 ->Fields->Fields[0]->AsString + "-ID ostidagi
o'qituvchini bazadan o'chirishni hohlaysizmi
?",mtConfirmation,TMsgDlgButtons()<<mbYes<<mbNo,0)==mrYes)
{ADOQuery1 ->Delete();
yangilash();
}
}
//-----
void __fastcall TForm1::sBitBtn4Click(TObject *Sender)
{
ADOQuery1 ->Edit();
DBGrid1 ->ReadOnly = false;
}
//-----
void __fastcall TForm1::sBitBtn5Click(TObject *Sender)
{
ADOQuery1 ->Post();
DBGrid1 ->ReadOnly = true;
yangilash();
}
//-----
void __fastcall TForm1::P2Click(TObject *Sender)
{
Form3->ShowModal();
}
//-----
void __fastcall TForm1::P1Click(TObject *Sender)
{
Form4->ShowModal();
}

```

```

}
//-----
void __fastcall TForm1::F1Click(TObject *Sender)
{
Form5->ADOTable1->Close();
Form5->ADOTable1->TableName = "Fanlar";
Form5->ADOTable1->Open();
Form5->ADOTable1->Active = True;
Form5->ShowModal();
}
//-----
void __fastcall TForm1::P3Click(TObject *Sender)
{
Form5->ADOTable1->Close();
Form5->ADOTable1->TableName = "Pedagogik_darajalar";
Form5->ADOTable1->Open();
Form5->ADOTable1->Active = True;
Form5->ShowModal();
}
//-----
void __fastcall TForm1::yangilash()
{
AnsiString yol1 = ExtractFilePath(Application->ExeName) + "\\Kafedra.mdb";
ADOConnection1 ->Connected = False;
ADOConnection1 ->ConnectionString="Provider=Microsoft.Jet.OLEDB.4.0;User
ID=Admin;Data Source=\\\"+yol1+\\\";Mode=Share Deny None;Jet
OLEDB:System database=\\\";Jet OLEDB:Registry Path=\\\";Jet
OLEDB:Database Password=\\\";Jet OLEDB:Engine Type=5;Jet
OLEDB:Database Locking Mode=1;Jet OLEDB:Global Partial Bulk Ops=2;Jet
OLEDB:Global Bulk Transactions=1;Jet OLEDB:New Database
Password=\\\";Jet OLEDB>Create System Database=False;Jet OLEDB:Encrypt
Database=False;Jet OLEDB:Don't Copy Locale on Compact=False;Jet
OLEDB:Compact Without Replica Repair=False;Jet OLEDB:SFP=False;";
ADOConnection1 ->Connected = True;
ADOQuery1 ->Active = True;
Form2->ADOTable1->Active = True;
Form2->ADOTable2->Active = True;
Form2->ADOTable3->Active = True;
Form3->ADOTable1->Active = True;

```

```

Form4->ADOTable1->Active = True;
Form5->ADOTable1->Active = True;
}
//-----
void __fastcall TForm1::sBitBtn6Click(TObject *Sender)
{
Form6->Image1 ->Picture = DBImage1 ->Picture;
Form6->sLabel2->Caption = ADOQuery1 ->Fields->Fields[1]->AsString + " " +
ADOQuery1 ->Fields->Fields[2]->AsString + " " + ADOQuery1 ->Fields-
>Fields[17]->AsString;
Form6->sLabel3->Caption = ADOQuery1 ->Fields->Fields[15]->AsString;
Form6->sLabel4->Caption = ADOQuery1 ->Fields->Fields[16]->AsString;
Form6->sLabel14->Caption = ADOQuery1 ->Fields->Fields[4]->AsString;
Form6->sLabel15->Caption = ADOQuery1 ->Fields->Fields[10]->AsString;
Form6->sLabel20->Caption = ADOQuery1 ->Fields->Fields[12]->AsString;
Form6->sLabel22->Caption = ADOQuery1 ->Fields->Fields[13]->AsString;
Form6->sLabel16->Caption = ADOQuery1 ->Fields->Fields[11]->AsString;
Form6->sLabel19->Caption = ADOQuery1 ->Fields->Fields[14]->AsString;
Form6->sLabel21->Caption = ADOQuery1 ->Fields->Fields[3]->AsString;
Form6->ShowModal();
}
#include <vcl.h>
#pragma hdrstop
#include "Unit6.h"
//-----
#pragma package(smart_init)
#pragma link "sLabel"
#pragma link "sPanel"
#pragma link "sSkinProvider"
#pragma resource "*.dfm"
TForm6 *Form6;
//-----
__fastcall TForm6::TForm6(TComponent* Owner)
: TForm(Owner)
{}
//-----
void __fastcall TForm6::sPanel1Click(TObject *Sender)
{
if( MessageDlg("Ma'lumotni chop etish uchun saqlashni

```

```
hohlaysizmi?",mtConfirmation,TMsgDlgButtons())<<mbYes<<mbNo,0)==mrYes)
Form6->Print();
else
Close();
}
```