

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

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

«KOMPYUTER TIZIMLARI» KAFEDRASI

«HIMOYAGA»
Kafedra mudiri

«__» _____ 2015 y

**“TATU Farg’ona filiali kadrlar bo’limi” ma’lumotlar
bazasini yaratish**

MAVZUSIDA

KURS ISHI

Bajardi: _____
(imzo)

Jalilov O
(familiyasi)

612-13 KI guruh talabasi

Tekshirdi: _____
(imzo)

(familiyasi)

Farg’ona 2015 yil

Mavzu: “TATU Farg’ona filiali kadrlar bo’limi” ma’lumotlar bazasini yaratish

REJA:

I. Kirish.

1.1 O’zbekistonda axborot texnologiyalarining yuksalishi

1.1 MBBT ni istiqbollari

II. Nazariy qism.

2.1.MBBT haqida umumiy ma’lumot

2.2 C++Builder dasturi haqida tushuncha

2.3.C++Builder dasturining asosiy komponentlari.

2.4. C++ Builder dasturlash tilini ma’lumotlar ombori bilan bog’lash.

III. Amaliy qism.

**3.1. TATU Farg’ona filiali kadrlar bo’limni avtomatlashtirish
.uchun Microsoft Office Access dasturida baza tayyorlash.**

3.2.Accesssda yaratilgan bazani C++ builderga bog’lash.

3.3.Masalani bajarish bosqichlari.

3.4.Dastur imkoniyatlari.

IV. Xulosa.

V. Foydalanilgan adabiyotlar.

VI. Ilova

Kirish

1.1 O'zbekistonda axborot texnologiyalarining yuksalishi.

O'zbekiston 1991 yil 1 sentabrda mustaqillikka erishdi va jahonga yuz tutdi. XXI asr bu axborot texnologiyalari asridir. Mamlakatimiz mustaqillika erishgandan so'ng yoshlarga katta e'tibor berildi. Yoshlarni zamonaviy kompyuterlar bilan ta'minlashga qaratildi.

O'zbekiston boshqa davlatlar bilan munosabatlarini va do'stona aloqalarini o'rnatishga kirishdi. Mustaqillikdan so'ng respublikamizda axborot texnologiyalariga katta e'tibor berildi. Jahoning ko'plab mamlakatlari zamonaviy kompyuterlarga va ularni yoshlarga bevosita o'rgatishga kirishdi. Yangi axborot texnologiyalari rivojlanishiga ko'plab mamlakatlar katta e'tibor berildi va har xil festival, musobaqa, tanlovlar qilindi. Shu jumladan, mamlakatimizda ham bunday ishlarga befarq qaralmayapti. Bunga misol, 2000 yil mart oyida vatanimizda birinchi marta Internet festivali bo'lib o'tdi. Festival 15 turdagi yo'nalishlardan iborat bo'ldi. Shular qatori, iqtisodiy va ijtimoiy sohalar haqida yo'nalishlar mavjud. Bu diyorumizda axborot texnologiyalariga talab va e'tibor kuchli ekanligini ko'rsatadi. O'zbekistonda axborot texnologiyalarining rivojlanishtirish ishlari olib borilmoqda. Eng katta yutuq bu O'zbekistonda milliy kompyuter tarmoqlarining shakllanishidir va u orqali axborot almashinish mumkin. Mamlakatimizda O'rta Osiyoda birinchilardan bo'lib, elektron hisoblash mashinasi ishga tushirilgan. 1956 yilda O'zFA matematika instituti qoshida hisoblash texnikasi bo'limi ochildi. Bu O'rta Osiyo respublikalari ichida birinchi bunday bo'lim ochilishi edi. 1958 yilga kelib esa dekabrning o'ttizinchi kuni O'rta Osiyoda birinchi bo'lib "Ural" elektron hisoblash mashinasi tashkil etildi. Bunday rivojlanish va e'tibor o'z kuchini ko'rsatdi. Toshkent shahri elektron-hisoblash mashinalari sohasidagi markazlarning biriga aylandi. Bunday katta yutuqlarni yanada mustahkamlash maqsadida 1962 yil Toshkent shahrida M20 rusumli, yarim o'tkazgichli M220 rusumli, "Minsk 22", "Ural 2", "BCM6-1", "BCM6-2" rusumli elektron-hisoblash mashinalari ishga tushirildi. Uskunalarining ishga tushirilishi yanada yangi izlanishlar va rivojlantirish ishlari yanada avj olib ketdi. Bunday rivojlantirishlar yangi ehtiyojlarni sezdi va 1966-yilda bo'lim nagizida yangi "Kibernetika" ilmiy ishlab chiqarish birlashmasi ochildi. O'zbekistanimiz mustaqil bo'lgandan so'ng ham axborot texnologiyalariga katta e'tibor berildi va bu bo'lim, birlashmalar yanada rivojlantirildi. 2001 yil "Kibernetika" ilmiy ishlab chiqarish birlashmasida yangi axborot texnologiyalari markazi barpo etildi. Mustaqil O'zbekistonimizda kompyuterlashtirishni rivojlantirish, zamonaviy kompyuterlarni qo'llash va ularni samarali, sifatli ishlatish, kompyuter bozorini kengaytirish, shu va boshqa yo'nalishlar bo'yicha tadbirlar o'tkazildi va yanada yangi yo'nalishlari rejalashtirilmoqda.

”Informatika” fanlar ichida eng yoshi hisoblanadi. Uni yanada rivojlantirish, kompyuter dasturlari bilan tanishish, o’sib kelayotgan yosh avlodni elektron-hisolash mashinalari bilan tanishtirish, ularni o’rgatishni yangi bosqichlarini ishlab chiqish mamlakatimizning dolzarb va asosiy mavzusiga aylangan. Maktab, kollej, litsey, institut, universitetlarni talabga javob beradigan kompyuterlar bilan ta’minlash, ularga eng zamonaviy kompyuterlarni va ular bilan ishlash o’rgatish, O’zbekistonni har jabhada jahonga ko’rsatish asosiy vazifadir. Yoshlarni bo’sh vaqtini mazmunli o’tkazish maqsadida “Kamolot” ko’p va ho’p ishlar qilmoqda. Bizning shaharda “Kamolot”ning bo’limi mavjud. “Kamolot” yoshlarni bo’sh vaqtini qiziqarli o’tkazish maqsadida, bo’linga “Ingliz tili” kursi va “Informatika” kursi tashkil etdi. U yerga “Kamolot” tomonidan zamonaviy kompyuterlar berildi. Bu O’zbekistonda axborot texnologiyalarining qanday rivojlanayotganligi bir kichik va oddiy misoldir. Jahonning ko’p davlatlari O’zbekistonimizga boshqacha ko’z bilan qarashmoqda. Yosh avlodga ko’rsatilayotgan va berilayotgan imkoniyatlar boshqa davlatlarni xavasini keltirmoqda. Har bir davlat rivojlanish yo’lida ma’lum qiyinchiliklarga uchraydi. O’zbekiston ham bundan mustasno emas. Yangi iqtisodiy munosabatlarga o’tish davrida o’zining texnologik imkoniyatlarini rivojlantirish faqat zamonaviy axborot texnologiyalarini qo’llash orqali amalga oshirilishini talat etiladi.

1.2 MBBT ni istiqbollari

Axborotlarni saqlash, qayta ishlash va uzatish kabi vazifalarni bajarishda ma’lumotlar va bilimlar bazasi texnologiyalarining xizmati beqiyos ekaniga ishonch hosil qilmoqda. Respublikamizdagi viloyatlar, shaharlar, tumanlarga qarashli korxonalar, tashkilotlar va muassasalar zamonaviy kompyuter texnikalari bilan jihozlanib, ular maxsus qurilmalar yordamida axborotlarni uzatish va qabul qilish imkoniyatiga ega bo’lmoqda.

Insonning iqtisodiy, ekologik, siyosiy va boshqa sohalarda fikrlash doirasining kengayishi axborotli muhitning sifat va miqdor jihatdan o’zgarishi, yangi xususiyatga ega bo’lgan axborotli muhitning kelib chiqishiga sabab bo’lmoqda. «Ma’lumotlar bazasi» fanini o’rganishdan asosiy maqsad har bir

talabada kompyuter texnologiyalarga bo'lgan qiziqishni uyg'otish, amaliy informatika va avtomatlashtirilgan axborot tizimlaridan foydalanib turli iqtisodiy masalalar echish usullari sohasida bilimlar berishdir.

Boshqaruv jarayonidagi axborotlarni saqlash, saralash, uzatish, qabul qilish, qayta ishlash va foydalanish kabi amallarni organish va turli sohalarga tadbiq qilish mazkur fanning predmeti hisoblanadi.

Ma'lumotlar bazasini boshqarish tizimi bu umumiy tushuncha bo'lib, uning tarkibiga ma'lumotlar bazasi ham kiradi. Misol uchun, mashina chiqaradigan zavod bu MBBT hisoblanadi, mashinalar esa ma'lumotlar bazasidir. MBBT'da ma'lumotlar bazasidan tashqari boshqa narsalar ham bo'ladi, misol uchun ma'lumotlar bazasini kompyuter tezkor xotirasidan joy olishi, kompyuter protsessorlarini ma'lumotlar bazasi bilan bog'lanishi, hullas ma'lumotlar bazasi Xususiyl holda misol ko'ramiz, Oracle MBBT'da ma'lumotlar bazasi hosil qilsangiz, ma'lumotlar bazasidan tashqari PGA va SGA degan tushunchalar bo'ladi, bular MBBT'ga tegishli bo'lib, ma'lumotlar bazasi ishlashi uchun kompyuter xotirasidan joy ajratadi, undan tashqari klient tomondan ma'lumotlar

Nazariy qism

2.1.MBBT haqida umumiy ma'lumot

Ma'lumotlar bazasi – biror sohaga oid o'zaro bog'langan ma'lumotlar yig'indisining disk tashuvchidagi tashkiliy jamlanmasidir. Boshqacha qilib aytganda ma'lumotlar bazasi - bu kompyuter xotirasiga yozilgan ma'lum bir strukturali, o'zaro bog'langan va tartiblangan ma'lumotlar majmu bo'lib, u biror bir ob'yektning xususiyatini, holatini yoki ob'yektlar o'rtasidagi munosabatni ma'lum ma'noda tavsiflaydi.

Ma'lumotlar bazasini boshqarish tizimi (MBBT) -bu dasturiy va apparat vositalarining murakkab majmui bo'lib, ular yordamida foydalanuvchi ma'lumotlar bazasini yaratish va shu bazadagi ma'lumotlar ustida ish yuritish mumkin. MBBT o'z maxsus dasturlash tillariga ham ega bo'lib, bu tillarga buyruqli dasturlash tillari deyiladi. MBBTga Oracle, Clipper, Paradox, FoxPro, Access va boshqalarni misol keltirish mumkin.

Informatsion tizimlarni yaratish bo'yicha jadal harakatlar va ma'lumotlar hajmining tez sur'atlar bilan oshib borishi 60 yillar boshida maxsus "Ma'lumotlar bazasini boshqarish tizimi" (MBBT) dab ataluvchi dasturiy kompleksning yaratilishiga olib keldi. 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 qildi. 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. Internetga asoslanuvchi ma'lumotlar bazalarini Yaratish, hozirgi davrda keng rivojlanib bormoqda.

SQL

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 qacda joylashgani, qanday indekslar va hatto amallarning eng effektiv ketma ketligini qo'llash kerakligini aniqlaydi; bu detallarni ma'lumotlar bazasiga so'rovlarda ko'rsatish kerak emas.

MySQL Ma'lumotlar Bazasi

Ma'lumotlar bazasi o'ta tez rivojlangan hamda MySQL va mSQL o'zini ko'rsatgan soha, Internet uchun dasturlar yaratishdir. Internet uchun murakkab va ishonchli dasturlarga ehtiyoj oshgan sari ma'lumotlar bazasiga ehtiyoj ham oshib bormoqda. Server ma'lumotlar bazasi Internetda ko'p funktsiyalarni qo'llashi mumkin. Har qanday veb- sahifa ma'lumotlar bazasi tomonidan boshqarilishi mumkin.

Informatsion tizimlarni yaratish bo'yicha jadal harakatlar ma'lumotlar hajmining tez suratlar bilan oshib borishi sharoitida 60 yillar boshida maxsus "Ma'lumotlar bazasini boshqarish tizimi" (MBBT) dab ataluvchi dasturiy kompleksning yaratilishiga olib keldi.

MBBT asosiy xususiyatlari - bu protseduralar tarkibi bo'lib, ular faqat ma'lumotlarni kiritish va saqlashda ishlatilmasdan, ularning strukturasi ham tasvirlaydi. Ma'lumotlarni o'zida saqlab va MBBT ostida boshqariladigan fayl, oldin ma'lumotlar banki deb atalib, keyinchalik esa "Ma'lumotlar bazasi" dab yuritila bosladi.

Ma'lumotlarni boshqarish tizimi, quyidagi xossalarga ega:

- ✓ fayllar to'plami mantiqiy kelishuvni quvvatlaydi;
- ✓ ma'lumotlar ustida ish yuritish tili bilan ta'minlaydi;
- ✓ har xil to'xtalishlardan keyin ma'lumotlarni qayta tiklaydi;
- ✓ MBBT bir necha foydaloanuvchilarning parallel ishlashini ta'minlaydi.

MBBT funktsiyalari tarkibiga yanada aniqroq qilib quyidagilar qabul qilingan:

- ✓ **Tashqi xotirada bevosita ma'lumotlarni boshqarish.**

Bu funktsiya MBga bevosita kiruvchi ma'lumotlarni saqlash uchun kerakli strukturani ta'minlab tashqi xotiraga qo'shadi. MBBT ishlatishda mavjud fayl tizimi imkoniyatlari aktiv ravishda ishlatiladi. Rivojlantirilgan MBBTda foydalanuvchi istalgan holda MBBT fayl tizimini ishlatayapdimi bu haqda bilishi shart emas, va agar ishlatlsa, u holda fayllar tashkil qilingan bo'ladi. Xususiy holda MBga berkitilgan ob'ektlarni MBBT quvvatlaydi.

✓ **Tezkor xotirani bufer bilan boshqarish**

MBBT odatda ancha katta hajmdagi MB bilan ish yuritadi. Bu hajm odatda tezkor xotiraning mumkin bo'lgan hajmidan yetarli darajada katta bo'ladi. Ma'lumki, agar ma'lumotlarning biror elementiga murojaat qilish kerak bo'lsa tashqi xotira bilan aloqa o'rnatiladi, lekin barcha tizim tashqi xotira qurilmasi tezligida ishlaydi. Bu tezlikni oshirishning amaliy yagona usulilaridan biri bu operativ xotiraga ma'lumotlarni buferizatsiya qilishdir.

✓ **Tranzaktsiya bilan boshqarish**

Tranzaktsiya – bu qaralayotgan MBBT MB ustida ketma-ket operatsiyalarni bajararishidir, ya'ni ma'lumotlar bilan monipulyatsiya qilib ketma-ket operatsiyalar yordamida MBBTga ta'sir etishdir. Tranzaktsiya ma'lumotlar bazasini bir butun holatdan ikkinchi bir butun holatga o'tkazadi, yoki agar ma'lum sababga ko'ra tranzaktsiyaning biror holati bajarilmaydigan bo'lsa yoki tizimda biror xatolik yuz bersa, ma'lumotlar bazasi boshlang'ich holatiga qaytadi. MBning mantiqiy butunligini quvvatlash uchun tranzaktsiya tushunchasi kerak.

✓ **Jurnalizatsiya**

MBBT ga bo'lgan asosiy talablardan biri bu tashqi xotirada ma'lumotlarning ishonchli saqlanishidir. Ma'lumotlarning ishonchli saqlanishi deganda har qanday apparatli yoki dasturli to'xtab qolishdan (sboydan) keyin MBBT MBning oxirgi holatini qayta tiklashi tushuniladi. Odatda apparatli to'xtab qolish holati ikki xil bo'ladi: engil to'xtab qolish, ya'ni bunda kompyuter ishlashi kutilmaganda to'xtashi (masalan, elektr toki manbaining o'chishi), ikkinchisi qattiq to'xtab

qolish, bu tashqi xotirada ma'lumotlarning yo'qolib ketishi bilan xarakterlanadi. Dasturli to'xtab qolishlarga quyidagilarni keltirishi mumkin: MBBTning to'satdan buzilishi bilan ishni tugatishi (dastur xatosi bo'yicha yoki qaysidir apparatning to'xtab qolishi natijasida) yoki foydalanuvchi dasturining avariya bilan tugallanishi bo'lib natijada ayrim tranzaktsiyalar tugallanmasdan qoladi. Har qanday holda ham MB qayta tiklash uchun qo'shimcha ma'lumotlarni joylashtirish kerak. Boshqacha qilib aytganda MB da ma'lumotlarning butunligini saqlash uchun saqlanadigan ma'lumotlarning to'liqligi talab qilinadi. Ma'lumotlarning ishlatilayotgan qismi qayta tiklanishi uchun alohida ishonchli saqlanishi lozim. Bunda to'liq ma'lumotlarni quvatlash uchun keng tarqalgan usullardan biri MB ning o'zgartirish jurnalini olib borish usuli ishlatiladi.

Jurnal – MBning asosiy qismi bo'lib hisoblanadi va u barcha rivojdagi MBBT da “jurnal zapisi utverjdeniy” (pratakol Write Ahead Log - WAL) deb nomlanadi.

Qattiq to'xtab qolishdan keyin MBni qayta tiklash uchun jurnal va MBning arxiv nusxasi ishlatiladi. Arxiv nusxa - bu MBning to'liq nusxasi bo'lib, jurnalni to'ldirish momentidan boshlanadi.

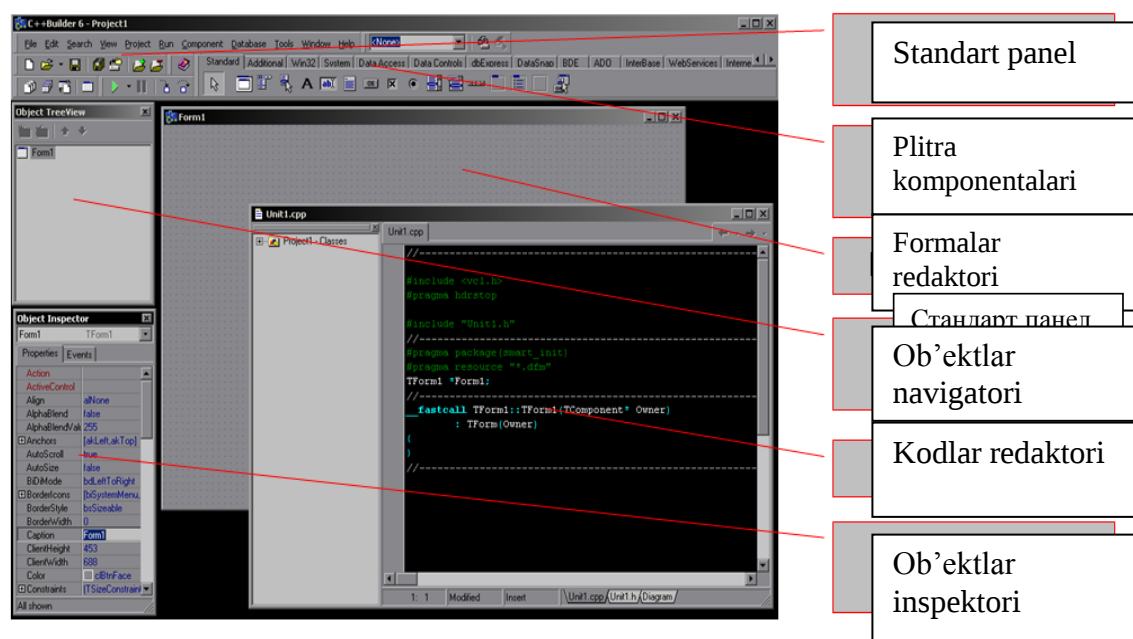
Relyatsion ma'lumotlar bazasi – ikki o'lchamli jadvallarning mohiyat-munosabat mehanizmi orqali bog'lanishidir. Relyacion ma'lumotlar bazasining logic aloqasi o'zaro logic bog'langan jadvallar majmuini tashkil etadi.

Ma'lumotlarning relyacion modeli normallashtirish orqali yaratiladi. Normallashtirish jarayoni qayta takrorlanuvchi ma'lumotlarni yangi jadvalga kuchirishdan iborat hodisadir. Relyacion jadval structurasi maydon bilan belgilanadi. Maydonga ma'lumot turi va ahajmi kabi hossalari kiritiladi. Maydon tarkibi jadval ustunlarida keltiriladi.

Har bir qator ma'lumotning bitta ekzemplari bo'lib, yozuv deb nomlanadi. Har bir yozuvni identifikatsiyasi uchun unikal birlamchi kalit qullaniladi. Kalit bir yoki bir nechta maydondan iborat bo'lishi mumkin. Ikki jadvalni bog'lash uchun tashqi kalitdan foydalaniladi. Bunda birga-bir (1:1), birga-kup(1:N) va kupga-kup(M:N) munosabatlar urnatilishi mumkin.

2.2. C++Builder dasturi haqida tushuncha.

Ishlab chiqishning integratsiyalashgan muhiti Komponentalar palitrasini birlashtiradi. SHakllar Muharriri, Kod Muharriri, Ob'ektlar Noziri, Ob'ektlar Xazinasi - bular hammasi kod va zahiralalar ustidan to'liq nazoratni ta'minlovchi dasturiy ilovalarni tez ishlab chiqish instrumentlari (2.1-rasm).



2.1-rasm. Ishlab chiqish muhitining tuzilishi

■ **Komponentlar Palitrasi** ilovalarni qurishda taklif qilinadigan 100 dan ortiq takroran qo'llanadigan komponentlardan iborat.

■ **SHakllar Muharriri** dasturning foydalanuvchi bilan interfeysini yaratish uchun mo'ljallangan.

■ **Kod Muharriri** dastur matnini, xususan, voqealarga ishlov berish funktsiyalarini yozish uchun mo'ljallangan.

■ **Ob'ektlar Noziri** qotib qolgan chigal dasturlash zaruratibiz ob'ektlar xususiyatlarini vizual o'rnatish imkonini beradi hamda shunday voqealarni o'z ichiga oladiki, bu voqealarni ularning paydo bo'lishiga nisbatan ob'ektlar reaksiyasi kodlari bilan bog'lash mumkin bo'ladi.

■ **Ob'ektlar Xazinasi** ma'lumotlarning shakl va modullari kabi ob'ektlarga ega bo'lib, ular ishlab chiqishda muvaqqat sarflarni kamaytirish maqsadida ko'plab ilovalar bilan bo'linadi.

C++ Builder ilovalarni qurishning vizual metodikasini Komponentlar Palitrasidan kerakli boshqarish elementlarini tanlab olish vositasida joriy etadi. Har bir komponenta (masalan, tugmacha) bilan ushbu komponenta turini va xulq-atvorini o'zgartiradigan xususiyatlar bog'liq bo'ladi. Har qanday komponenta ushbu komponentaning turli xildagi ta'sirlarga reaksiyasini (munosabatini) aniqlab beradigan voqealar seriyasini keltirib chiqarishi mumkin. Bunday keyin => belgilari biz C ++Builder muhitida amalga oshiradigan xatti-harakatlarni bildiradi.

=>C++Builder ni chaqiring va bosh menyudagi File | New Application komandasi bo'yicha yangi ilovalar ustida ishlashni boshlang.

=>sichqonchani Komponentalar Palitrasining qo'shimcha ilovalari ustida bosib, foydalanuvchi ish ko'radigan dastur interfeysi elementlarining mavjud assortimentini ko'rib chiqing.

Palitraning bir qo'shimcha ilovasidan ikkinchisiga o'tib, kirish mumkin bo'lgan komponentlar to'plami o'zgarayotganining guvohi bo'lishimiz mumkin. Sichqoncha kursori komponentlar belgisi ustida to'xtaganda, aytib turish nomi paydo bo'ladi. Agar F1 klavishasini bossak, tizimning ma'lumotnomalar xizmati tanlab olingan komponenta haqida to'liq ma'lumot chiqarib beradi.

Xususiyatlar, metodlar va voqealar

Ilovalarning tez ishlab chiqilishi ob'ektlilik asosidagi dasturlash doirasida xususiyatlar, metodlar va voqealarning qo'llab-quvvatlanishini bildiradi. **Xususiyatlar** komponentalarning nomlar, matniy aytilishlar yoki ma'lumotlar manbalari kabi turli xildagi tavsiflarini osongina o'rnatish imkonini beradi. **Metodlar** (a'zo-funksiyalar) komponentadagi ob'ekt ustida ma'lum operatsiyalarni amalga oshiradi. Bunday operatsiyalar jumlasida qayta tiklash yoki multimedia qurilmasini qayta o'rash kabi murakkab operatsiyalarni ham ko'rsatish mumkin. **Voqealar** komponentalarga foydalanuvchi ko'rsatayotgan faollashtirish (aktivizatsiya), tugmalarni bosish yoki tahrir qilinadigan kiritish kabi ta'sirlarni ushbu ta'sirlarga bizning munosabat kodlaringiz bilan bog'laydi. Bundan tashqari voqealar komponentalar holatlarida sodir bo'ladigan ayrim o'ziga xos o'zgarishlar

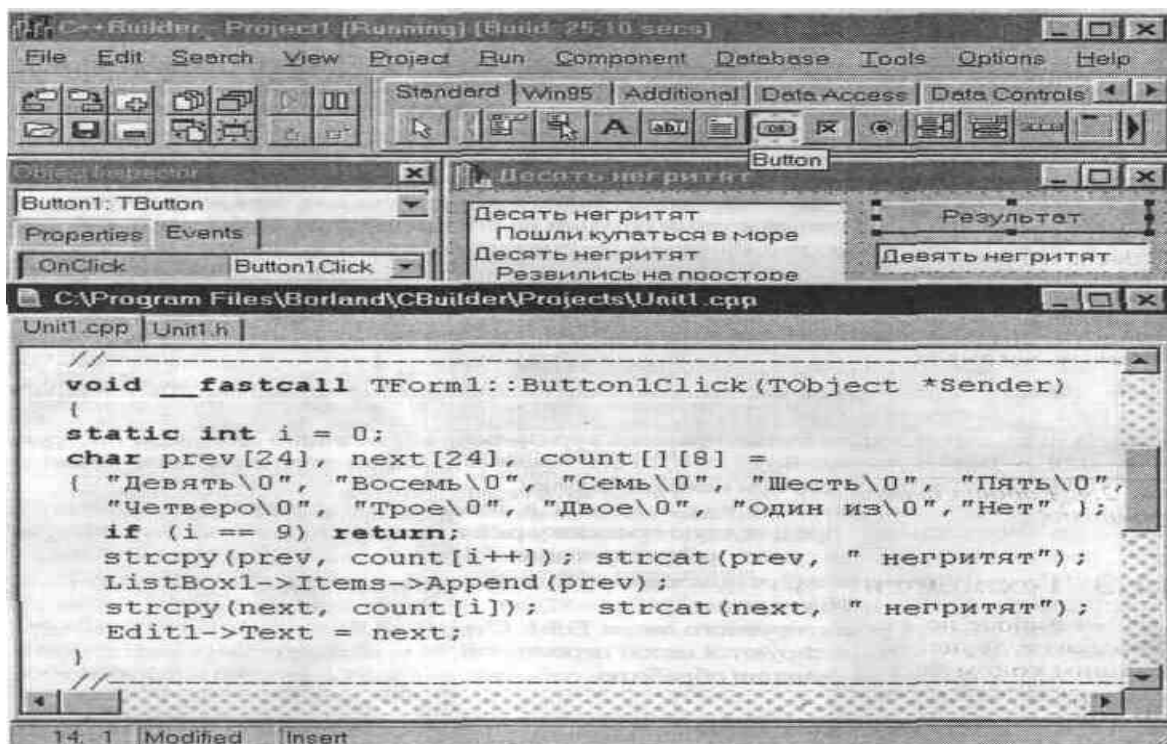
paytida ham yuzaga kelishi mumkin. Bunday o'ziga xos o'zgarishlar qatorida ma'lumotlar bazasiga kirishning interfeysli elementlarida ma'lumotlarni yangilashni ko'rsati o'tish kifoya. Xususiyatlar, metodlar va voqealar, birgalikda ish olib borar ekan, ular Windows uchini shonchli ilovalarni intuitiv tarzda dasturlash muhiti - RAD ni hosil qiladi.

=> Tanlangan ob'ekt bilan assotsiatsiyalanadigan (birgalikda yodga olinadigan) voqealarni ko'rish uchun, Ob'ektlar Nozirida Voqealar (Events) qo'shimcha ilovasini ko'rsating.

=> O'zingiz shaklga joylashtirgan tugma komponentasini sichqoncha bilan ikki marta uring

=> Ochilgan Kod Muharriri darchasida kursor ButtonIClick funktsiyasi tanasiga instruktsiyalarni kiritish uchun pozitsiyani ko'rsatadi. Bu funktsiya esa tugmachani bosishda yuzaga keladigan OnClick voqeasiga ishlov berish uchun mo'ljallangan.

2.2-rasmda oddiy kod ko'rsatilgan bo'lib, u «Natija» tugmasini yana bir bor bosilishiga javoban avval turgan plev aytib berishini ro'yxat oxiriga, hamda navbatdagi next aytib berishini tahrir qilish maydoniga qo'shadi. ListBoxI->Items->Append(prev) yo'riqnomasi, Append metodi yordamida, rrev satrini ListBoxI ro'yxati ob'ektining Items xususiyatiga qo'shadi. EditI->Text=next yo'riqnomasi tahrir qilinayotgan EditI kiritish ob'ektining Text xususiyatiga next satrini taqdim etadi. Aytib berish satrlari ikki o'lchamli count massivida saqlanadi va static turdagi butun o'zgaruvchi tomonidan indekslanadi. Bu o'zgaruvchi esa ButtonI tugmasini bosish bilan yuzaga keladigan voqeaga ishlov berish funktsiyasining chaqirilishlari o'rtasida o'zining joriy qiymatini saqlaydi.



2.2-rasm. Kod muharriri bajarilayotgan modul matnining Unit1.cpp faylida kiritilishi va tahrir qilinishini ta'minlaydi.

Birinchi versiyali ilovani loyihalash bosqichi shuning bilan tugallanadi va ishchi dasturni yaratishga kirishish mumkin bo'ladi.

=>Run | Run bosh menyusi komandasi bilan ilovani kompilyatsiya qilish (ko'chirish) va yig'ish jarayonini ishga tushirib yuboring

=>Dastur chaqirilgach, bir necha marta «Natija» («Rezultat») tugmasini bosing.

2.3. C++Builder dasturining asosiy komponentlari.

C++ Builder 32 razryadli takomillashtirilgan Vizual Komponentalar Kutubxonasi VCL (Visual Component Library) bilan birgalikda etkazib beriladi. Bu kutubxona eng murakkab ilovalarni qurish uchun mo'ljallangan 100 dan ortiq takroran qo'llanadigan komponentalardan iborat. Kutubxonaning asosiy komponentalari Palitralar komponentalarining instrumental Panelida berilgan. Komponentalar belgilari dasturingiz shakliga olib o'tiladi.

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:

- mavjud komponenta turiga vorislik;
- yangi xususiyatlar, metodlar va voqealarni aniqlash;
- yaratilgan komponentani qayd etish.

Qidirish oson bo'lishi uchun, Palitra 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.

Standart komponentalar

Komponentalar palitrasining Standard qo'shimcha ilovalari komponentalari bizning dasturingizga Windows standart interfeysli elementlarning 14 tasining ulanishini amalga oshiradi.

TMainVlenu

Bosh menyu komandolari panelini va ularga mos keladigan tushib qoladigan menyularni yaratadi. Barcha menyu komandalarining identifikatorlari menyuning har qanday konkret komandasiga kirish huquqiga ega bo'lgan Items xususiyati bilan aniqlanadi, AutoMerge xususiyati Merge va Unmerge metodlari bilan birgalikda turli shakldagi menyularning birlashish jarayonini boshqaradi.

TPopUpMenu

Shakl yoki bironta boshqa komponenta uchun maxsus menyu yaratadi. E'tiborga oling, aynan shu maqsad uchun har qanday boshqa komponenta PopUpMenu xususiyatiga ega bo'lib, bu xususiyatda biz uning bilan bog'liq menyuga iqtibos qilishingiz mumkin.

Agar biz sichqonchaning o'ng tugmasini shaklga yoki berilgan komponenta mansub bo'lgan biron boshqa elementga bosish bilan maxsus menyu ekranda paydo bo'lishini xohlasangiz, AutoPopup xususiyatining true qiymatini o'rnatib. Veqea qayta ishlatgichi - OnPopup yordamida bevosita maxsus menyuning paydo bo'lishi oldidan bajariladigan protsedurani aniqlash mumkin.

Tlabel

Shaklda tahrir qilib bo'lmaydigan satik matnning to'rtburchak sohasini aks ettiradi. Odatda matn boshqa komponenta nomidan iborat bo'ladi.

Nom matni Caption xususiyatining qiymatidir. Alignment xususiyati matnni tekislash usulini aniqlaydi. Shrift o'lchami avtomatik tarzda sohaning maksimal to'ldirilishiga mos kelishi uchun, AutoBize xususiyatining true qiymatini o'rnatib. Kalta soha ichida matnning hammasini ko'rish imkoniga ega bo'lish uchun, WordWrap xususiyatining true qiymatini bering. Transparent xususiyatining true qiymatini o'rnatib, boshqa komponentaning bir qismini to'g'ri uning ustida joylashtirilgan nom orasidan ko'rinib turadigan qilishingiz mumkin.

Tedit

Axborot yakka satrining tahrir qilinayotgan kiritishidagi to'rtburchak sohani shaklda aks ettiradi. Tahrir sohasining ichidagi boshlang'ich narsalarni Text xususiyatining qiymati bo'lgan satr aniqlaydi.

TEdit komponentasi TCustomEdit sinfining to'g'ridan-to'g'ri hosilasi bo'lib, uning barcha xususiyatlari, metodlari va voqelariga vorislik qiladi.

TMemo

Axborot ko'plab satrining tahrir qilinayotgan kiritishidagi to'rtburchak sohani shaklda aks ettiradi. Tahrir sohasining ichidagi boshlang'ich narsalarni Lines xususiyatining qiymati bo'lgan satrlar massivi aniqlaydi. Ushbu xususiyat qiymati ustunida tugmachani bossangiz, ro'yxat elementlari muharririning darchasi ochiladi.

TMemo komponentasi TCustomMemo sinfining to'g'ridan-to'g'ri hosilasi bo'lib, uning barcha xususiyatlari, metodlari va voqealariga vorislik qiladi.

TButton

Yozuvli to'rtburchak tugmani yaratadi. Tugmacha bosilganda, dasturda biron-bir xatti-harakat nomlanadi (initsiallashtiriladi).

Tugmachalar ko'proq dialogli darchalarda qo'llanadi. Default xususiyatining true qiymati tomonidan tanlab olingan yashirin tugmacha, dialog darchasida har gal Enter klavishasi bosilganda, OnClick voqea qayta ishlatgichini ishga tushiradi. Cancel xususiyatining true qiymati tanlab olgan uzish tugmachasi, dialog darchasida har gal Escape klavishasi bosilganda, OnClick voqea qayta ishlatgichini ishga tushiradi.

TButton komponentasi TButtonControl sinfining hosilasi hisoblanadi.

TCheckBox

Ikkita holatga hamda tavsifiy matnga ega bo'lgan kvadrat chek-boksni yaratadi (bunda tavsifiy matn chek-boksning vazifasini spetsifikatsiya qiladi).

Boks holatini bildiruvchi «check» biron-bir variantning tanlanishiga mos keladi (boks ustidan tortilgan chiiq bilan belgilanadi), «unchecked» holati esa tanlov olib tashlanishiga mos keladi - bunda Checked komponentasining xususiyati mos ravishda o'zgaradi hamda OnSlick voqeasi yuzaga keladi. Tavsifiy matn Caption xususiyatida saqlanadi. AllowGrayed xususiyatining true qiymatini o'rnatib, boksni to'qroq rangli (masalan, kulrang) qilish mumkin. State xususiyati joriy holatni va boks rangini aks ettiradi.

TComboBox

Tahrir sohasi hamda matn variantlarining tushib qoladigan ro'yxati kombinatsiyasini tanlash uchun yaratadi.

Text xususiyatining qiymati bevosita tahrir sohasiga kiritib qo'yiladi. Foydalanuvchi tanlab olishi mumkin bo'lgan ro'yxat elementlari Items xususiyatining ichida bo'ladi, dasturning bajarilish paytida tanlab olinishi mumkin bo'lgan element raqami ItemIndex xususiyatining ichida bo'ladi, tanlab olingan matnning o'zi esa SelText xususiyatining ichida bo'ladi. SelStart va SelLength xususiyatlari matnning qaysi qismini tanlab olishni belgilab berish yoki matnning qaysi qismi tanlab olinganini bilish imkonini beradi.

Items ob'ekting Add, Append, Delete va Insert metodlari yordamida ro'yxat elementlarini dinamik tarzda qo'shish, o'chirish orasiga qo'yish va o'rnini almashtirish mumkin, masalan:

ComboBox1->Items->Insert(0, «Ro'yxatdagi birinchi element»);

Sorted xususiyatining true elementi ro'yxat elementlarini alifbo tartibida navlarga ajratilishini ta'minlaydi. TComboBox komponentasining turini Style xususiyatidan tanlab olish mumkin.

TComboBox komponentasi TCustomComboBox sinfining hosilasi bo'lib uning barcha xususiyatlari, metodlari va voqealariga vorislik qiladi.

TGroupBox

To'g'ri burchakli ramka ko'rinishidagi konteyner bo'lib, u qandaydir bir interfeys elementlarining mantiqan bog'langan guruhini shaklda vizual birlashtiradi. Bu komponenta Windows ning bir nomdagi ob'ektning inkapsulalanishidan iborat.

TPanel

Boshqa komponentlarni o'z ichiga olishi mumkin bo'lgan bo'sh panelni yaratadi. Biz TPanel dan o'z shaklingizda instrumentlar paneli yoki holatlar satrlarini yaratish uchun foydalanishingiz mumkin.

TPanel panel komponentasi TCustomPanel sinfining hosilasi bo'lib, uning barcha xususiyatlar, metodlari va voqealari to'liq vorislik qiladi.

C++ da o'zgaruvchilar turlari va tavsiflari

Har bir nom va har bir ifoda o'z turiga ega bo'lib, bu tur o'z ustida amalga oshirilishi mumkin bo'lgan operatsiyalarni belgilab beradi. Masalan, int "i" tavsifini olaylik. U "i" ning int turga ega ekanini, ya'ni "i" butun o'zgaruvchi ekanini belgilaydi. Tavsif — dasturga nom kiritadigan operator. Tavsif ushbu nomga tur bag'ishlaydi. Tur nomning yoki ifodaning to'g'ri qo'llanishini belgilaydi. Butunlar uchun +, —, * va / kabi operatsiyalar belgilangan.

Apparat taminoti vositalariga yaqindan bevosita javob beradigan asosiy turlar quyidagilardir: char short int long float double. Dastlabki to'rtta tur butun sonlarni, keyingi uchta esa suzuvchi nuqtali sonlarni, ya'ni kasrlarni ifodalashda qo'llanadi. Char turdagi o'zgaruvchi ushbu mashinada simvolni saqlash uchun mo'ljallangan o'lchamga ega, int turdagi o'zgaruvchining o'lchamlari esa ushbu mashinadagi butun arifmetikaga mos keladi (bu, odatda, so'z bo'ladi). Tur taqdim etishi mumkin bo'lgan butun sonlar diapazoni ushbu turning o'lchamlariga bog'liq bo'ladi (uni sizeof operatori yordamida hisoblab chiqarish mumkin).

C++da o'lchamlar char turdagi ma'lumotlar o'lchamlari birligida o'lchanadi, shuning uchun char ta'rifga binoan bir o'lchamga ega. Butun son ko'rsatkichni saqlash uchun etarli ekani hamma mashinalar uchun ham birdek to'g'ri kelavermaydi. Asosiy turga nisbatan "const" sifatini qo'llash mumkin. Bunda dastlabki turga xos bo'lgan xususiyatlarga ega turni olish imkonini beradi. Bunda faqat bitta istisno bor. U shundaki, const turidagi o'zgaruvchilarning qiymati initsiallashtirish (nomlashtirish) dan so'ng o'zgara olmaydi. Bitta tishli qo'shtirnoqqa olingan belgi (simvol) belgi konstantasi hisoblanadi. E'tibor bering, shunday yo'l bilan aniqlangan konstanta xotirani egallamaydi. Shunchaki, konstanta qiymati kerak bo'lib qolgan o'rinda u bevosita qo'llanishi mumkin bo'ladi.

Inkapsulyatsiyalash

Inkapsulyatsiyalash dasturni qandaydir monolit, bo'linmas narsa sifatida olib qaramay, ko'plab mustaqil elementlarga bo'lish imkonini beradi. Har bir element

o'z funksiyalarini boshqa elementlardan mustaqil ravishda bajara oladigan alohida modul sifatida olib qaraladi. Aynan inkapsulyasiyalash tufayli mustaqillik darajasi ortadi, chunki ichki detallar interfeys ortida yashiringan bo'ladi. Inkapsulyasiyalash modullikning ob'ektga mo'ljallangan tavsifidir. Inkapsulyasiyalash yordamida dasturiy ta'minotni ma'lum funksiyalarni bajaruvchi modullarga bo'lib tashlash mumkin. Bu funksiyalarni amalga oshirish detallari esa tashqi olamdan yashirin holda bo'ladi. Mohiyatan inkapsulyasiyalash atamasi «germetik berkitilgan; tashqi ta'sirlardan himoyalangan dastur qismi» degan ma'noni bildiradi. Agar biron-bir dasturiy ob'ektga inkapsulyasiyalash qo'llangan bo'lsa, u holda bu ob'ekt qora quti sifatida olib qaraladi. Siz qora quti nima qilayotganini uning tashqi interfeysini ko'rib turganingiz uchungina bilishingiz mumkin. Qora quti biron narsa qilishga majburlash uchun, unga xabar yuborish kerak. Qora quti ichida nima sodir bo'layotgani ahamiyatli emas, qora quti yuborilgan xabarga adekvat (mos ravishda) munosabatda bo'lishi muhimroqdir.

Interfeys tashqi olam bilan tuzilgan o'ziga xos bitim bo'lib, unda tashqi ob'ektlar ushbu ob'ektga qanday talablar yuborishi mumkinligi ko'rsatilgan bo'ladi. Interfeys – ob'ektni boshqarish pulti.

Polimorfizm

Agar inkapsulyasiyalash va vorislikni OMYO ning foydali vositalari sifatida olib qarash mumkin bo'lsa, polimorfizm - eng universal va radikal vositadir. Polimorfizm inkapsulyasiyalash va vorislik bilan chambarchas bog'liq, boz ustiga, polimorfizmsiz OMYO samarali bo'lolmaydi. Polimorfizm - OMYO paradigmasida markaziy tushunchadir. Polimorfizmni egallamay turib, OMYO dan samarali foydalanish mumkin emas.

Polimorfizm shunday holatki, bunda qandaydir bitta narsa ko'p shakllarga ega bo'ladi. Dasturlash tilida «ko'p shakllar» deyilganda, bitta nom avtomatik mexanizm tomonidan tanlab olingan turli kodlarning nomidan ish ko'rishi tushuniladi. Shunday qilib, polimorfizm yordamida bitta nom turli hulq-atvorni bildirishi mumkin.

Vorislik polimorfizmning ayrim turlaridan foydalanish uchun zarurdir. Aynan o'rindoshlik imkoniyati mavjud bo'lgani uchun, polimorfizmdan foydalanish mumkin bo'ladi. Polimorfizm yordamida tizimga to'g'ri kelgan paytda qo'shimcha funksiyalarni qo'shish mumkin. Dasturni yozish paytida hatto tahmin qilinmagan funkcionallik bilan yangi sinflarni qo'shish mumkin, buning ustiga bularning hammasini dastlabki dasturni o'zgartirmay turib ham amalga oshirish mumkin. Yangi talablarga osongina moslasha oladigan dasturiy vosita deganda, mana shular tushuniladi.

Polimorfizmning uchta asosiy turi mavjud:

- Qo'shilish polimorfizmi
- Parametrik polimorfizm
- Ortiqcha yuklanish.

Qo'shilish polimorfizmini ba'zida sof polimorfizm deb ham ataydilar. Qo'shilish polimorfizmi shuning bilan qiziqarlilik, uning tufayli tarmoq sinf nushalari o'zini turlicha tutishi mumkin. Qo'shilish polimorfizmidan foydalanib, yangi tarmoq sinflarni kiritgan holda, tizimning hulq-atvorini o'zgartirish mumkin. Uning bosh afzalligi shundaki, dastlabki dasturni o'zgartirmay turib, yangi hulq-atvorni yaratish mumkin.

Aynan polimorfizm tufayli joriy qilishdan takroran foydalanishni vorislik bilan aynanlashtirish kerak emas. Buning o'rniga vorislikdan avvalam bor o'zaro almashinish munosabatlari yordamida polimorfizm hulq-atvorga erishish uchun foydalanish lozim. Agar o'zaro almashinish munosabatlari to'g'ri belgilansa, buning ortidan albatta takroran qo'llash chiqib keladi. Qo'shilish polimorfizmidan foydalanib, bazaviy sinfdan, har qanday avloddan, shuningdek bazaviy sinf qo'llaydigan metodlardan takroran foydalanish mumkin.

Parametrik polimorfizmdan foydalanib, turdosh metodlar va turdosh (universal) turlar yaratish mumkin. Turdosh metodlar va turlar dalillarning ko'plab turlari bilan ishlay oladigan dasturni yozish imkonini beradi. Agar qo'shilish polimorfizmidan foydalanish ob'ektni idrok etishga ta'sir ko'rsatsa, parametrik polimorfizmdan foydalanish qo'llanayotgan metodlarga ta'sir ko'rsatadi.

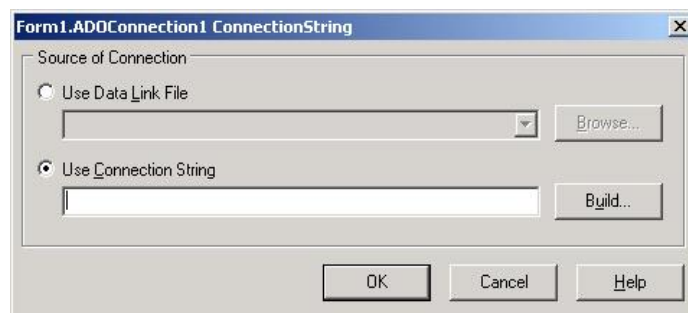
Parametrik polimorfizm yordamida, parametr turini bajarilish vaqtigacha e'lon qilmay turib, turdosh metodlar yaratish mumkin. Metodlarning parametrik parametrlari bo'lganidek, turlarning o'zi ham parametrik bo'lishi mumkin. Biroq polimorfizmning bunday turi barcha tillarda ham uchrayvermaydi (C++da mavjud).

2.4. C++ Builder dasturlash tilini ma'lumotlar ombori bilan bog'lash.

C++ Builder dasturlash tilini ma'lumotlar ombori bilan bog'lash turlari ko'p bo'lib ulardan biz ADO komponentalar to'plamiga tegishli bo'lgan ADOConnection, ADOTable va DataAccses komponentalar to'plamiga tegishli bo'lgan DataSource komponentalaridan foydalanamiz.

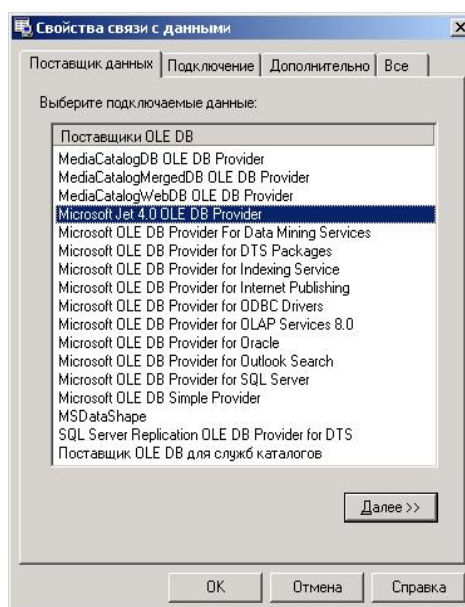
Komponentning- ko'rinishi	Komponentning- nomi	Vazifasi
	ADOConnection	Bu komponent ma'lumotlar bazasini Delphi dasturlashtili bilan bir-biriga bog'laydi.
	ADOTable	Bu komponent ma'lumotlar bazasidagi jadvalni Delphi dasturlash tili bilan bir-biriga bog'laydi.
	DataSource	Bu komponent ma'lumotlar bazasidagi jadvaldagi ma'lumotni inson ko'ra oladigan ko'rinishga keltiradi va aksincha

Bu komponentlarni Delphi dasturlash tilining proekt oynasiga ya'ni formaga o'rnatamiz. Komponentlardan ADOConnectionni tanlab Object Inspector oynasidagi Properties (xususiyat) bo'limidagi ConnectionString  bandini tanlaymiz. Bu banddagi  tugmani chertamiz. Shunda quyidagi oyna xosil bo'ladi.(2.3– rasm)



2.3– rasm. Bazani C++Builder dasturlash tili
bilan bo'g'lash oynasi.

Bu oynadagi **Build...** tugmasini chertamiz. Shunda quyidagi oyna xosil bo'ladi. (2.4 – rasм)



2.4 – rasm. Bazani C++Builder dasturlash tili
bilan bog'lash uchun bazani turini ko'rsatish
oynasi.

Bu oynadagi **Подключение** oynasiga o'tib bazani qayerda turganini ko'rsatamiz va **OK** tugmasini bosamiz.

Shundan so'ng ADOTable komponentini aktivlashtiramiz. Uning xususiyatlar oynasidan Connection xususiyatini tanlab komponentani ADOConnection komponentasi bilan bog'laymiz. DataSource komponentasi xususiyatlar oynasidan DataSet xususiyatini tanlaymiz va ADOTable komponentasini bir-biriga bog'laymiz.

Ma'lumotlar ombori bilan ishlaydigan komponentalar ro'yxati.

Data Access bo'lami



DataSource – ma'lumotlarni yoki komponentlarni bir-biriga bog'lash.



Table – ma'lumotlar omborini (faylni) bog'lash.



Query – ma'lumotlar omboridagi yozuvlarni boshqarish.



StoredProc –serverdan ma'lumotlar omborini yuklash.



Database –yagona ma'lumotlar omborini bog'lash.

Data Controls bo'limi



DBGrid – ma'lumot omboridagi yozuvlarni jadval ko'rinishida chiqarish.



DBNavigator – ma'lumotlar omboridagi yozuvlarni taxrirllovchi komponent.



DBText – ma'lumotlar omboridagi matnli maydon ma'lumotlarini chiqarish.



DBEdit – ma'lumotlar omboridagi biror maydonni tahrirlash.



DBMemo – ma'lumotlar omboridagi memo tipidagi ma'lumotlarni tahrirlash.



DBImage – ma'lumotlar omborida joylangan tasvirlarni ko'rsatish.



DBListBox – ma'lumotlar omboridagi ma'lumotlarni listga chiqarish.



DBComboBox-ma'lumotlarni kombinatsiyali tanlash.

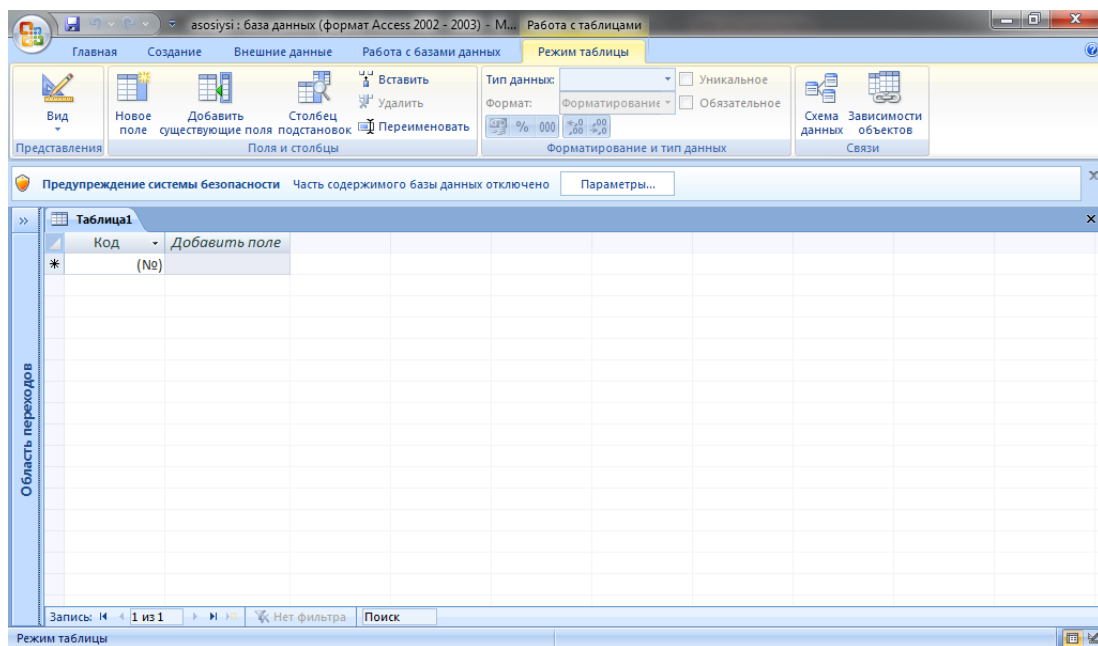


DBRichEdit – ma'lumotlarni tahrirlashning memo ganisbatan kengroq imkoniyati.

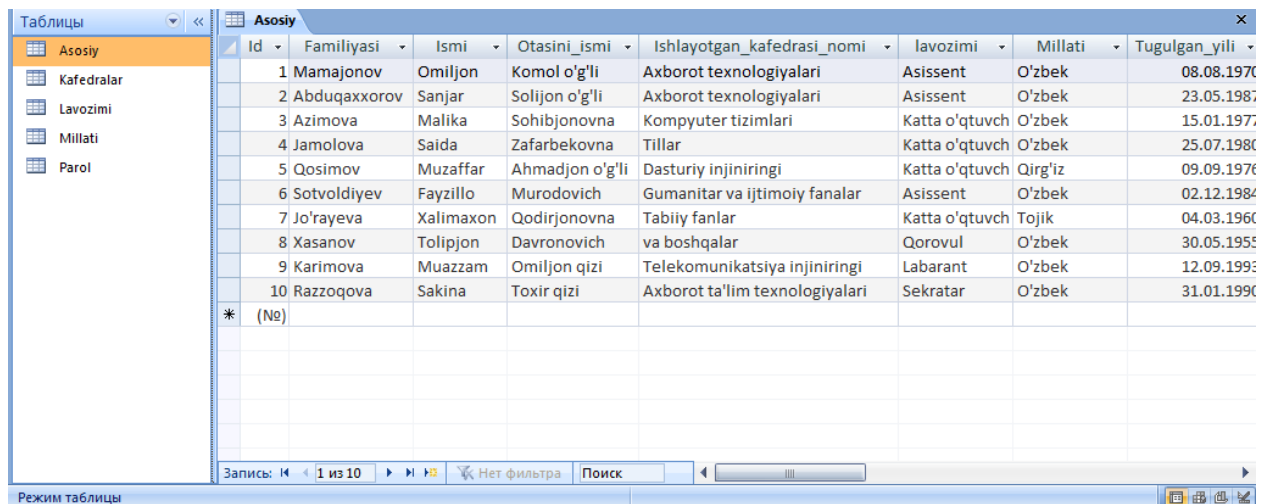
Amaliy qism.

Dasturni yaratishda avvalo ma'lumotlar bazasini, undagi jadvallarni, jadvaldagi ustunlarni to'g'ri kiritish lozim. Jadval, ustun nomlarini yozishda probel, apostrof(') belgilaridan foydalanmaslik zarur. Chunki dasturda SQL so'rovlardan foydalanganda xatoliklar sodir berishi mumkin.

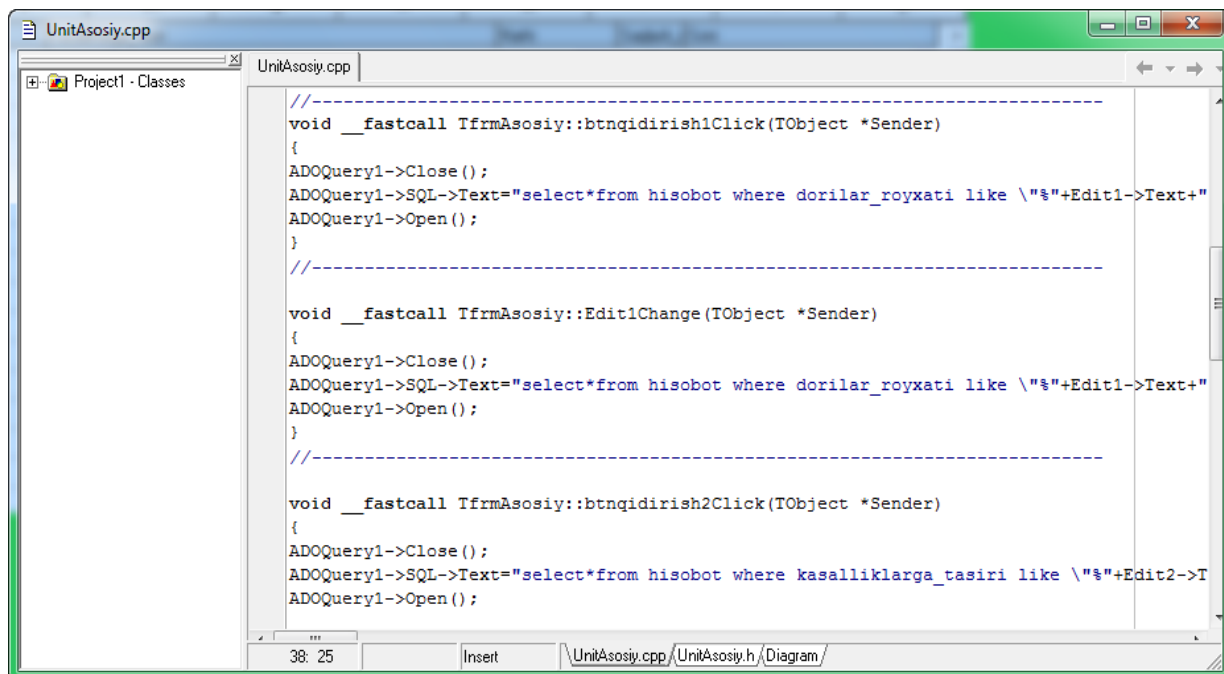
MBBT Access programmani ishga tushirish uchun: Pusk menyusiga kirib, u yerdan Microsoft office tanlab olamiz. So'ngra Microsoft Accessni tanlaymiz.



U yerdan bizga kerakli bo'lgan Ma'lumotlar bazasini tuzib olamiz



“TATU Farg’ona filiali kadrlar bo’limi” dasturini tuzishda “asosiysi” nomli ma’lumotlar bazasi xosil qilib oldim. “asosiysi” nomli bazada “to’rtta” tablitsa mavjud ya’ni bular: “Asosiy”, “kafedralar”, “lavozimi”, “millati” deb nomlangan tablitsalar mavjud. Asosiy tablitsasida TATUFF kadrlar bo’limini aks ettirgan unda barcha xodimlarni ma’lumotlari kiritiladi qolgan tablitsalar yordamchi tablitsa xisoblanib ular xam Asosiy tablitsasiga bog’langan. Bu “asosiysi” bazamizni tayyorlab olgandan so’ng uni foydalanuvchilarga qulay bo’lishi uchun dasturlaymiz , dasturni yaratishda C++ Builder 6 dasturidan foydalanamiz. Dasturni ishga tushiramiz va C++ Builder Application Form yaratib olamiz. Bu asosiy forma hisoblanadi.



Bazaga bog’lash uchun ADOConnection komponentasidan, jadvaldagi ma’lumotlarni olish uchun ADOQuery va ADOTable komponentalaridan, ADOQuery va ADOTable komponentalaridagi ma’lumotlarni dasturga bo’g’lash uchun DataSource komponentasidan foydalanamiz. Dasturda Panel, BitBtn, Label, ComboBox Edit, DbBGrid, GroupBox komponentalaridan ham foydalanib, uning dizaynini yaratib oldik.

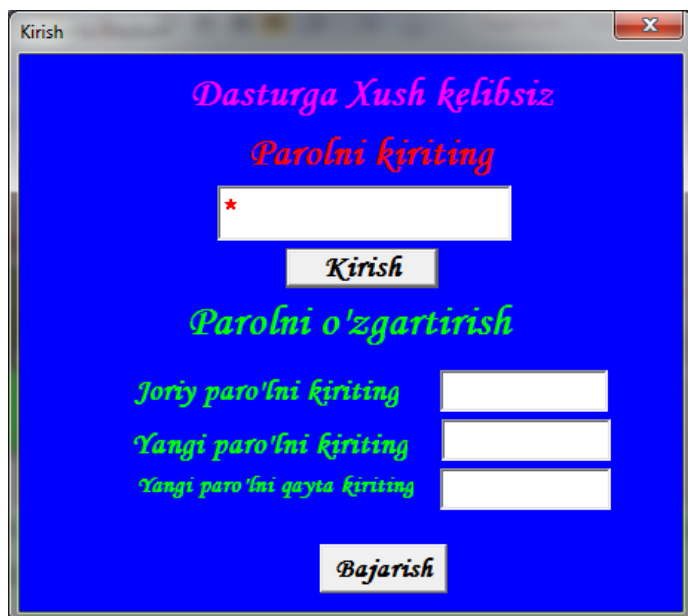
Komponentalar palitrasi ilovalarni qurishda taklif qilinadigan 100 dan ortiq takroran qo‘llanadigan komponentalardan iborat. Shakllar muharriri dasturning

foydalanuvchi bilan interfeysini yaratish uchun mo'ljallangan. Kod muharriri dastur matnini, xususan, voqealarga ishlov berish funksiyalarini yozish uchun mo'ljallangan. Obyektlar noziri qotib qolgan chigal dasturlash zaruratisiz obyektlar xususiyatlarini vizual o'rnatish imkonini beradi hamda shunday voqealarni o'z ichiga oladiki, bu voqealarni ularning paydo bo'lishiga nisbatan obyektlar reaksiyasi kodlari bilan bog'lash mumkin bo'ladi.

Dastur ishlatilganda quyidagi oyna hosil bo'ladi.

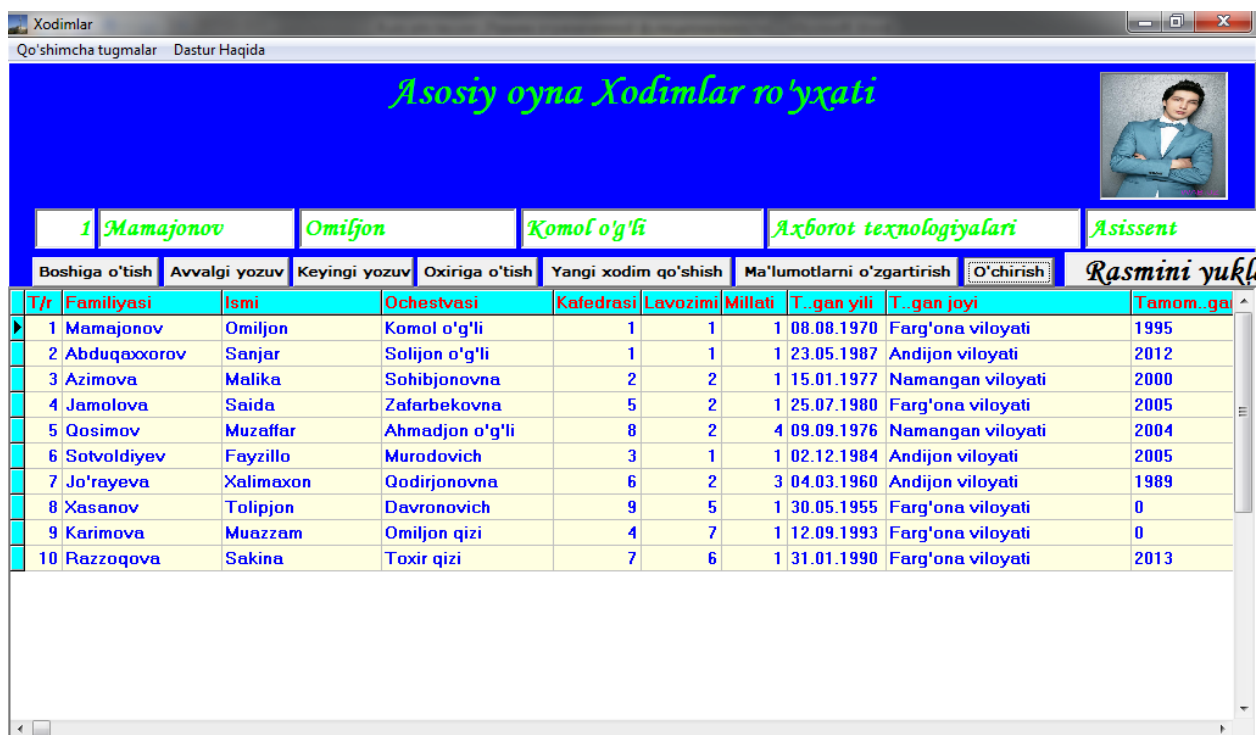


Yuqoridagi oynada kadrlar bo'limi deb nomlangan tugma bilan keyngi oyna xosil bo'ladi



Bu oynada, parol orqali dasturga kirishimiz mumkin va parolni yangilashimiz xam mumkin bular juda qulay va tushunarli ishlangan.

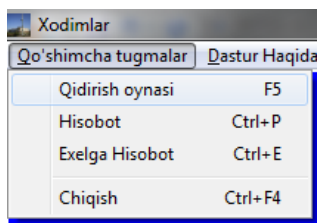
Bajarish tugmasi bosilganda quyidagi oyna ochiladi:



2-rasm

Bu oynada bazadagi asosiy tablitsasi va qo'shimcha xizmatlarni bajaradigan tugmalar xosil bo'ladi. Bu oynada ma'lumotlarni taxrirlash, yangi ma'lumotlar qo'shish, ularni o'chirish va rasmlarni yuklash kabi ishlarni qilish mumkin yuqori chekkasida joylashgan qo'shimcha tugmalar deb nomlangan tugma mavjud, bu

tugmada Qidiruv oynasi, Hisobot, Exelga Hisobot va Dasturdan CHiqish tugmalari joylashgan.



3-rasm

Qidiruv oynasi tugmasini bosilganda quyidagi oyna hosil bo'ladi:



4-rasm

Hisobot tugmasini bosganda hisobot kabi fe'llarni chop etish mumkin.

Bu turdagi imkoniyatdan kadrlar bo'limini ma'lumotlarini qog'ozga chop etish imkoniyatini beradi.

 Toshkent Axborot Texnologiyalari Universiteti Farg'ona Filiali Kadrlar bo'limi								
Id	Familiyasi	Ismi	Otasini ismi	Kafedrası	Lavozimi	Millati	Tug'ulgan sanaRasmi	
1	Mamajonov	Omiljon	Komol o'g'li	1	1	1	08.08.1970	
2	Abduqaxxorov	Sanjar	Solijon o'g'li	1	1	1	23.05.1987	
3	Azimova	Malika	Sohibjonovna	2	2	1	15.01.1977	
4	Jamolova	Saida	Zafarbekovna	5	2	1	25.07.1980	
5	Qosimov	Muzaffar	Ahmadjon o'g'li	8	2	4	09.09.1976	
6	Sotvoldiyev	Fayzillo	Murodovich	3	1	1	02.12.1984	

5-rasm

Exelga Hisobot tugmasida Microsoft Exelga import qilib beradi

Книга1 - Microsoft Excel										
Главная Вставка Разметка страницы Формулы Данные Рецензирование Вид										
Общий Число Стили Ячейки										
B1 fx										
A	B	C	D	E	F	G	H	I	J	K
id	Familiyasi	Ismi	Otasini_ismi	Ishlayotgan_kafedrası_nomi	Lavozimi	Millati	Tugulgan_yili	Tugulgan_joyi	OTM_tugatgan_yili	Tugatgan
1	Mamajonov	Omiljon	Komol o'g'li		1	1	08.08.1970	Farg'ona viloyati	1995	TATU
2	Abduqaxxorov	Sanjar	Solijon o'g'li		1	1	23.05.1987	Andijon viloyati	2012	TaTu
3	Azimova	Malika	Sohibjonovna		2	2	15.01.1977	Namangan viloyati	2000	Namanga
4	Jamolova	Saida	Zafarbekovna		5	2	25.07.1980	Farg'ona viloyati	2005	Johon til
5	Qosimov	Muzaffar	Ahmadjon o'g'li		8	2	09.09.1976	Namangan viloyati	2004	Namanga
6	Sotvoldiyev	Fayzillo	Murodovich		3	1	02.12.1984	Andijon viloyati	2005	Andijon i
7	Jo'rayeva	Xalimaxon	Qodirjonovna		6	2	04.03.1960	Andijon viloyati	1989	Toshkent
8	Xasanov	Tolipjon	Davronovich		9	5	30.05.1955	Farg'ona viloyati	0	OTM o'qi
9	Karimova	Muazzam	Omiljon qizi		4	7	12.09.1993	Farg'ona viloyati	0	Xali tamc
10	Razzoqova	Sakina	Toxir qizi		7	6	31.01.1990	Farg'ona viloyati	2013	TATU Far

6-rasm

Bu dasturim Toshkent Axborot Texnologiyalar Universiteti Farg'ona Filiali karlar bo'limini ish jarayonini yengilatadi chunki ko'p imkoniyatga ega yuqorida aytib o'tgan imkoniyatlari xammasi emas dasturni ko'rib chiqib o'zingiz amin bo'lasiz.

Xulosa:

Men ushbu kurs ishini bajarish davomida C++ dasturlash tili da dasturlash hamda avtomatlashtirish to'g'risidagi bilimlarimni mustaxkamlab oldim. Dastur tuzish davomida MICROSOFT ACCESS dasturi yordamida Ma'lumotlar bazasini hosil qildim.

Ma'lumotlar bazasini hosil qilgach, uni C++ builder dasturiga bog'lab oldim. C++ builder ko'plab komponentalarni ishlatdim. Kurs ishini bajarish davomida dasturdagi ko'plab komponentalar palitrasidan, komponentalardan foydalanish, unda qidiruv tizimini tashkil qilishni, tahrirlash oynasini hosil qilish va ma'lumotlar bazasini shakllantirishni o'rgandim. Men dastur tuzish davomida MICROSOFT ACCESS dasturi yordamida Ma'lumotlar bazasini tuzdim, C++Builderda keng imkoniyatlar yaratilganligini tushundim. Microsoft Office kabi dasturlar bilan hamkorlikda ishlashi menda katta qiziqish uyg'otdi. Undan tashqari C++ builder dasturidagi imkoniyatlar, qulayliklarni o'zlashtirib oldim. Ushbu kurs ishini bajarish davomida olgan bilimlarimni mustahkamlab olishga erishdim.

Foydalanilgan adabiyotlar:

1. Каримов И.А. Баркамолавлод - Ўзбекистон таракқиётининг пойдевори. Т., Шарк, 1998.
2. Каримов И.А.- Ўзбекистон ХХІ аср бўсағасида: хавфсизликка тахдид, баркарорлик шартлари ва таракқиёт кафолатлари, Т.:, Ўзбекистон, 1997.
3. Каримов И.А. Ўзбекистон иқтисодий салохатларни чуқурлаштириш йўлида. Тошкент. “Ўзбекистон” 1995й.
4. Кнут Д. “Искусство программирования” 1-т. М.: Мир, 1976 г.
5. Арипов М.М. “Информатика ва ахборот технологиялари”. Тошкент, “Ўқитувчи”, 2002 й.
6. Интернет. Delphi для начинающих. Delphi Beginner.chm.
7. ”С++ dasturlash ”fanidan ma’ruza matn.
8. С++ uslubiy qo’llanma.
9. www.ziyonet.uz.
10. PharmedSanoat.uz
11. <http://www.uzvip.uz/download.php?id=2106>
12. <http://dasturchi.uz>
13. www.embarcadero.com

ILOVA

Dasturdagi kodlardan misollar.

Dorilarni nomi bo'yicha Qidiruv oynasidagi kodlar

```
//-----
```

```
#include <vcl.h>
```

```
#include <ComObj.hpp>
```

```
#include <utilcls.h>
```

```
#pragma hdrstop
```

```
#include "Unit3.h"
```

```
#include "Unit4.h"
```

```
#include "Unit5.h"
```

```
#include "Unit6.h"
```

```
#include "Unit7.h"
```

```
//-----
```

```
#pragma package(smart_init)
```

```
#pragma resource "*.dfm"
```

```
TForm3 *Form3;
```

```
Variant app, books, book, sheets, sheet, cell;
```

```
//-----
```

```
__fastcall TForm3::TForm3(TComponent* Owner)
```

```
    : TForm(Owner)
```

```
{
```

```
}
```

```
//-----
```

```
void __fastcall TForm3::Button1Click(TObject *Sender)
```

```
{
```

```
    ADOTable1->First();
```

```
}
```

```

//-----
void __fastcall TForm3::Button2Click(TObject *Sender)
{
    ADOTable1->Prior();
}

//-----
void __fastcall TForm3::Button3Click(TObject *Sender)
{
    ADOTable1->Next();
}

//-----
void __fastcall TForm3::Button4Click(TObject *Sender)
{
    ADOTable1->Last();
}

//-----
void __fastcall TForm3::Button7Click(TObject *Sender)
{
    switch(MessageDlg("Bu xodimni o'chirishni xoxlaysizmi?",mtWarning,TMsgDlgButtons()
    <<mbOK<<mbCancel, 0))
    {
        case mrOk:ADOTable1->Delete();
            break;
    }
}

//-----
void __fastcall TForm3::Button5Click(TObject *Sender)
{
    ADOTable1->Insert();
    Form4->ShowModal();
}

//-----
void __fastcall TForm3::Button6Click(TObject *Sender)
{

```

```

ADOTable1->Edit();
Form5->ShowModal();
}
//-----
void __fastcall TForm3::Qidirishoynasi1Click(TObject *Sender)
{
Form6->ShowModal();
}
//-----
void __fastcall TForm3::ExelgaHisobot1Click(TObject *Sender)
{
app=CreateOleObject("Excel.Application");
app.OlePropertySet("SheetsInNewWorkbook", 4);
books=app.OlePropertyGet("Workbooks");
books.OleProcedure("Add");
book= books.OlePropertyGet("Item", 1);
sheets=book.OlePropertyGet("Worksheets");
sheet=sheets.OlePropertyGet("Item", 1);
sheet.OlePropertySet("Name", "asosiysi");

cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",2/*qator*/,1/*ustun*/);
cell.OlePropertySet("Value", "id");

cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",2,2);
cell.OlePropertySet("Value", "Familiyasi");

cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",2,3);
cell.OlePropertySet("Value", "Ismi");

cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",2,4);
cell.OlePropertySet("Value", "Otasini_ismi");

cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",2,5);

```

```
cell.OlePropertySet("Value", "Ishlayotgan_kafedraasi_nomi");
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",2,6);  
cell.OlePropertySet("Value", "Lavozimi");
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",2,7);  
cell.OlePropertySet("Value", "Millati");
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",2,8);  
cell.OlePropertySet("Value", "Tugulgan_yili");
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",2,9);  
cell.OlePropertySet("Value", "Tugulgan_joyi");
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",2,10);  
cell.OlePropertySet("Value", "OTM_tugatgan_yili");
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",2,11);  
cell.OlePropertySet("Value", "Tugatgan_OTM_nomi");
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",2,12);  
cell.OlePropertySet("Value", "Mutaxasisligi");
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",2,13);  
cell.OlePropertySet("Value", "Ilmiy_darajasi");
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",2,14);  
cell.OlePropertySet("Value", "Jinsi");
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",2,15);  
cell.OlePropertySet("Value", "Xozirgi_yashash_joyi");
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",2,16);
```

```
cell.OlePropertySet("Value", "Oyligini_qo'lga_tegishi");
```

```
ADOTable1->First();
```

```
for(int i=1;i<=ADOTable1->RecordCount;i++)
```

```
{
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",i+2,1);
```

```
cell.OlePropertySet("Value", i);
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",i+2,2);
```

```
cell.OlePropertySet("Value", ADOTable1->FieldByName("Familiyasi")->AsVariant);
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",i+2,3);
```

```
cell.OlePropertySet("Value", ADOTable1->FieldByName("Ismi")->AsVariant);
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",i+2,4);
```

```
cell.OlePropertySet("Value", ADOTable1->FieldByName("Otasini_ismi")->AsVariant);
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",i+2,5);
```

```
cell.OlePropertySet("Value", ADOTable1->FieldByName("Ishlayotgan_kafedraasi_nomi")->AsVariant);
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",i+2,6);
```

```
cell.OlePropertySet("Value", ADOTable1->FieldByName("Lavozimi")->AsVariant);
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",i+2,7);
```

```
cell.OlePropertySet("Value", ADOTable1->FieldByName("Millati")->AsVariant);
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",i+2,8);
```

```
cell.OlePropertySet("Value", ADOTable1->FieldByName("Tugulgan_yili")->AsVariant);
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",i+2,9);
```

```
cell.OlePropertySet("Value", ADOTable1->FieldByName("Tugulgan_joyi")->AsVariant);
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",i+2,10);
```

```
cell.OlePropertySet("Value", ADOTable1->FieldByName("OTM_tugatgan_yili")->AsVariant);
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",i+2,11);
```

```
cell.OlePropertySet("Value", ADOTable1->FieldByName("Tugatgan_OTM_nomi")->AsVariant);
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",i+2,12);
```

```
cell.OlePropertySet("Value", ADOTable1->FieldByName("Mutaxisisligi")->AsVariant);
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",i+2,13);
```

```
cell.OlePropertySet("Value", ADOTable1->FieldByName("Ilmiy_darajasi")->AsVariant);
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",i+2,14);
```

```
cell.OlePropertySet("Value", ADOTable1->FieldByName("Jinsi")->AsVariant);
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",i+2,15);
```

```
cell.OlePropertySet("Value", ADOTable1->FieldByName("Xozirgi_yashash_joyi")->AsVariant);
```

```
cell=sheet.OlePropertyGet("Cells").OlePropertyGet("Item",i+2,16);
```

```
cell.OlePropertySet("Value", ADOTable1->FieldByName("Oyligini_qo'lga_tegishi")->AsVariant);
```

```
ADOTable1->Next();
```

```
}
```

```
app.OlePropertySet("Visible", true);
```

```
}
```

```
//-----
```

```
void __fastcall TForm3::Hisobot1Click(TObject *Sender)
```

```
{
```

```
Form7->QuickRep1->Preview();
```

```
}
```

```
//-----
```

```

void __fastcall TForm3::DasturHaqida1Click(TObject *Sender)
{
    ShowMessage("Bu Dastur\n612-13 GURUH talabasi\nJalilov Omadjon\nTomonidan
    tuzilgan\nDasturda TATU Farg'ona Filiali\nKadrlar bo'limini ishlash strukturasi\nnishlab
    chiqilgan.");
}

//-----

void __fastcall TForm3::Button8Click(TObject *Sender)
{
    if(OpenPictureDialog1->Execute())
    {
        ADOTable1->Edit();
        DBImage1->Picture->Graphic->LoadFromFile(OpenPictureDialog1->FileName);
        ADOTable1->Post();
    }
}

//-----

void __fastcall TForm3::Chiqish1Click(TObject *Sender)
{
    Form3->Close();
}

//-----

Qidirish oynasi qodlari

//-----

#include <vcl.h>

#pragma hdrstop

#include "Unit6.h"
#include "Unit1.h"
#include "Unit3.h"

//-----

#pragma package(smart_init)

```

```
#pragma resource "*.dfm"
```

```
TForm6 *Form6;
```

```
//-----
```

```
__fastcall TForm6::TForm6(TComponent* Owner)
```

```
    : TForm(Owner)
```

```
{
```

```
}
```

```
//-----
```

```
void __fastcall TForm6::Button1Click(TObject *Sender)
```

```
{
```

```
    if(Edit1->Text!="")
```

```
    {
```

```
        Form3->ADOQuery1->Close();
```

```
        Form3->ADOQuery1->SQL->Text="select * from Asosiy where Tugulgan_yili like \"%"+Edit1->Text+"%\"";
```

```
        Form3->ADOQuery1->Open();
```

```
        Edit2->Visible=True;
```

```
        Edit3->Visible=True;
```

```
        Label4->Visible=True;
```

```
        Label5->Visible=True;
```

```
        Label6->Visible=True;
```

```
        BitBtn1->Visible=True;
```

```
    }
```

```
}
```

```
//-----
```

```
void __fastcall TForm6::Button2Click(TObject *Sender)
```

```
{
```

```
    if(Edit1->Text!="")
```

```
    {
```

```
        Form3->ADOQuery1->Close();
```

```
        Form3->ADOQuery1->SQL->Text="select * from Asosiy where Familiyasi like \"%"+Edit1->Text+"%\"";
```

```
        Form3->ADOQuery1->Open();
```

```
Edit2->Visible=False;
```

```
Edit3->Visible=False;
```

```
Label4->Visible=False;
```

```
Label5->Visible=False;
```

```
Label6->Visible=False;
```

```
BitBtn1->Visible=False;
```

```
}
```

```
}
```

```
//-----
```

```
void __fastcall TForm6::Button5Click(TObject *Sender)
```

```
{
```

```
if(Edit1->Text!="")
```

```
{
```

```
Form3->ADOQuery1->Close();
```

```
Form3->ADOQuery1->SQL->Text="select * from Asosiy where Ismi like \"%" + Edit1->Text + "%\"";
```

```
Form3->ADOQuery1->Open();
```

```
Edit2->Visible=False;
```

```
Edit3->Visible=False;
```

```
Label4->Visible=False;
```

```
Label5->Visible=False;
```

```
Label6->Visible=False;
```

```
BitBtn1->Visible=False;
```

```
}
```

```
}
```

```
//-----
```

```
void __fastcall TForm6::Button6Click(TObject *Sender)
```

```
{
```

```
if(Edit1->Text!="")
```

```
{
```

```
Form3->ADOQuery1->Close();
```

```
Form3->ADOQuery1->SQL->Text="select * from Asosiy where Otasini_ismi like \"%" + Edit1->Text + "%\"";
```

```

Form3->ADOQuery1->Open();
Edit2->Visible=False;
Edit3->Visible=False;
Label4->Visible=False;
Label5->Visible=False;
Label6->Visible=False;
BitBtn1->Visible=False;
}
}
//-----
void __fastcall TForm6::Button3Click(TObject *Sender)
{
if(Edit1->Text!="")
{
Form3->ADOQuery1->Close();
Form3->ADOQuery1->SQL->Text="select * from Asosiy where Tugulgan_joyi like \"%" + Edit1->Text + "%\"";
Form3->ADOQuery1->Open();
Edit2->Visible=False;
Edit3->Visible=False;
Label4->Visible=False;
Label5->Visible=False;
Label6->Visible=False;
BitBtn1->Visible=False;
}
}
//-----
void __fastcall TForm6::Button4Click(TObject *Sender)
{
if(Edit1->Text!="")
{
if(Edit1->Text=="Axborot texnologiyalari")

```

```
{
Form3->ADOQuery1->Close();
Form3->ADOQuery1->SQL->Text= "select * from Asosiy where Ishlayotgan_kafedراسي_nomi=1" ;
Form3->ADOQuery1->Open();
}else
if(Edit1->Text=="Kompyuter tizimlari")
{
Form3->ADOQuery1->Close();
Form3->ADOQuery1->SQL->Text= "select * from Asosiy where Ishlayotgan_kafedراسي_nomi=2" ;
Form3->ADOQuery1->Open();
}else
if(Edit1->Text=="Gumanitar va ijtimoiy fanalar")
{
Form3->ADOQuery1->Close();
Form3->ADOQuery1->SQL->Text= "select * from Asosiy where Ishlayotgan_kafedراسي_nomi=3" ;
Form3->ADOQuery1->Open();
}else
if(Edit1->Text=="Telekommunikatsiya injiniringi")
{
Form3->ADOQuery1->Close();
Form3->ADOQuery1->SQL->Text= "select * from Asosiy where Ishlayotgan_kafedراسي_nomi=4" ;
Form3->ADOQuery1->Open();
}else
if(Edit1->Text=="Tillar")
{
Form3->ADOQuery1->Close();
Form3->ADOQuery1->SQL->Text= "select * from Asosiy where Ishlayotgan_kafedراسي_nomi=5" ;
Form3->ADOQuery1->Open();
}else
if(Edit1->Text=="Tabiiy fanlar")
{
Form3->ADOQuery1->Close();
```

```
Form3->ADOQuery1->SQL->Text= "select * from Asosiy where Ishlayotgan_kafedra_nomi=6" ;
Form3->ADOQuery1->Open();
}else
if(Edit1->Text=="Axborot ta'lim texnologiyalari")
{
Form3->ADOQuery1->Close();
Form3->ADOQuery1->SQL->Text= "select * from Asosiy where Ishlayotgan_kafedra_nomi=7" ;
Form3->ADOQuery1->Open();
}else
if(Edit1->Text=="Dasturiy injiniringi")
{
Form3->ADOQuery1->Close();
Form3->ADOQuery1->SQL->Text= "select * from Asosiy where Ishlayotgan_kafedra_nomi=8" ;
Form3->ADOQuery1->Open();
}else
if(Edit1->Text=="va boshqalar")
{
Form3->ADOQuery1->Close();
Form3->ADOQuery1->SQL->Text= "select * from Asosiy where Ishlayotgan_kafedra_nomi=9" ;
Form3->ADOQuery1->Open();
}else
{ShowMessage("Kafedra nomini to'liq kiriting!");
}
Edit2->Visible=False;
Edit3->Visible=False;
Label4->Visible=False;
Label5->Visible=False;
Label6->Visible=False;
BitBtn1->Visible=False;
}}
//-----
void __fastcall TForm6::Button7Click(TObject *Sender)
```

```
{
if(Edit1->Text!="")
{
if(Edit1->Text=="Asissent")
{
Form3->ADOQuery1->Close();
Form3->ADOQuery1->SQL->Text= "select * from Asosiy where lavozimi=1" ;
Form3->ADOQuery1->Open();
}else
if(Edit1->Text=="Katta o'qtuvch")
{
Form3->ADOQuery1->Close();
Form3->ADOQuery1->SQL->Text= "select * from Asosiy where lavozimi=2" ;
Form3->ADOQuery1->Open();
}else
if(Edit1->Text=="Datsent")
{
Form3->ADOQuery1->Close();
Form3->ADOQuery1->SQL->Text= "select * from Asosiy where lavozimi=3" ;
Form3->ADOQuery1->Open();
}else
if(Edit1->Text=="Professor")
{
Form3->ADOQuery1->Close();
Form3->ADOQuery1->SQL->Text= "select * from Asosiy where lavozimi=4" ;
Form3->ADOQuery1->Open();
}else
if(Edit1->Text=="Qorovul")
{
Form3->ADOQuery1->Close();
Form3->ADOQuery1->SQL->Text= "select * from Asosiy where lavozimi=5" ;
Form3->ADOQuery1->Open();
}
```

```
}else  
if(Edit1->Text=="Skratar")  
{  
    Form3->ADOQuery1->Close();  
Form3->ADOQuery1->SQL->Text= "select * from Asosiy where lavozimi=6" ;  
Form3->ADOQuery1->Open();  
}else  
if(Edit1->Text=="Labarant")  
{  
    Form3->ADOQuery1->Close();  
Form3->ADOQuery1->SQL->Text= "select * from Asosiy where lavozimi=7" ;  
Form3->ADOQuery1->Open();  
}else  
if(Edit1->Text=="Direktor")  
{  
    Form3->ADOQuery1->Close();  
Form3->ADOQuery1->SQL->Text= "select * from Asosiy where lavozimi=8" ;  
Form3->ADOQuery1->Open();  
}else  
if(Edit1->Text=="Fakultet dekani")  
{  
    Form3->ADOQuery1->Close();  
Form3->ADOQuery1->SQL->Text= "select * from Asosiy where lavozimi=9" ;  
Form3->ADOQuery1->Open();  
}else  
if(Edit1->Text=="Dekan o'rinbosar")  
{  
    Form3->ADOQuery1->Close();  
Form3->ADOQuery1->SQL->Text= "select * from Asosiy where lavozimi=10" ;  
Form3->ADOQuery1->Open();  
}else  
if(Edit1->Text=="O'quv ishlari bo'yicha direktor o'rinbosari")
```

```

{
    Form3->ADOQuery1->Close();
    Form3->ADOQuery1->SQL->Text= "select * from Asosiy where lavozimi=11" ;
    Form3->ADOQuery1->Open();
}else
if(Edit1->Text=="Ma'naviyat ishlari bo'yicha direktor o'rinbosari")
{
    Form3->ADOQuery1->Close();
    Form3->ADOQuery1->SQL->Text= "select * from Asosiy where lavozimi=12" ;
    Form3->ADOQuery1->Open();
}else
if(Edit1->Text=="va boshqalar")
{
    Form3->ADOQuery1->Close();
    Form3->ADOQuery1->SQL->Text= "select * from Asosiy where lavozimi=13" ;
    Form3->ADOQuery1->Open();
}
else
{ShowMessage("To'liq nomini kiriting!");
}
Edit2->Visible=False;
Edit3->Visible=False;
Label4->Visible=False;
Label5->Visible=False;
Label6->Visible=False;
BitBtn1->Visible=False;
}
}
//-----
void __fastcall TForm6::BitBtn1Click(TObject *Sender)
{
    if(Edit2->Text!="" && Edit3->Text!="")

```

```
{  
    Form3->ADOQuery1->Close();  
  
    Form3->ADOQuery1->SQL->Text="select * from Asosiy where Tugulgan_yili between #"+Edit2->Text+"# and #"+Edit3->Text+"# order by Tugulgan_yili";  
  
    Form3->ADOQuery1->Open();  
}  
}  
  
//-----
```

```
void __fastcall TForm6::Label7MouseMove(TObject *Sender, TShiftState Shift,  
    int X, int Y)  
{  
    Label7->Font->Color=clLime;  
}  
  
//-----
```

```
void __fastcall TForm6::Label7MouseLeave(TObject *Sender)  
{  
    Label7->Font->Color=clRed;  
}  
  
//-----
```

```
void __fastcall TForm6::Label7Click(TObject *Sender)  
{  
    Button4->Visible=True;  
    Button7->Visible=True;  
    Label7->Visible=False;  
}  
  
//-----
```

```
void __fastcall TForm6::Button8Click(TObject *Sender)  
{
```

```

if(Edit1->Text!="")
{
Form3->ADOQuery1->Close();

Form3->ADOQuery1->SQL->Text="select * from Asosiy where Ishlayotgan_kafedراسي_nomi like
\"%"+Edit1->Text+"%\"";

Form3->ADOQuery1->Open();

Edit2->Visible=False;
Edit3->Visible=False;
Label4->Visible=False;
Label5->Visible=False;
Label6->Visible=False;
BitBtn1->Visible=False;
}
}
//-----

```

```

void __fastcall TForm6::Button9Click(TObject *Sender)
{
if(Edit1->Text!="")
{
Form3->ADOQuery1->Close();

Form3->ADOQuery1->SQL->Text="select * from Asosiy where lavozimi like \"%"+Edit1-
>Text+"%\"";

Form3->ADOQuery1->Open();

Edit2->Visible=False;
Edit3->Visible=False;
Label4->Visible=False;
Label5->Visible=False;
Label6->Visible=False;
BitBtn1->Visible=False;
}
}
//-----

```

