



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

TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI

FARG'ONA FILIALI

**“ KOMPYUTER INJINIRINGI ” FAKULTETI
DASTURIY INJINIRING KAFEDRASI
“C++ DA DASTURLASH”
FANIDAN**

Kurs ishi

Bajardi:

630-14 guruh

MAMASOLIYEV A

Qabul qildi:

MUSAYEV X

Farg'ona 2015

**Mavzu: ARM foydalanuvchilarini ro'yhatga olish,
foydalanuvchilarning kitoblardan foydalanish tizimini dasturiy
ta'minotini yaratish.**

Reja:

- I. Kirish**
- II. Nazariy qism**
 - 2.1 C++ tarixi**
 - 2.2 C++ tili va ob'ektlarga mo'ljallangan dasturlash.**
 - 2.3 C++ Builder asoslari**
- III. Asosiy qism**
 - 3.1 Ma'lumotlar bazasini yaratish.**
 - 3.2 Dastur loyihasini yaratish.**
 - 3.3 Dasturdan foydalanish yo'riqnomasi.**
- IV. Xulosa**
- V. Foydalanilgan adabiyotlar**
- VI. Ilova.**

Kirish.

Keyingi yillarda amaliy dasturchilarga juda ko'p integratsion dastur tuzish muhitlari taklif etilayapti. Bu muhitlar u yoki bu imkoniyatlari bilan bir-biridan farq qiladi. Aksariyat dasturlashtirish muhitlarining fundamental asosi C++ tiliga borib taqaladi.

Ob'ektga mo'ljallangan dasturlash.

Ob'yektga mo'ljallangan yondashuv tarixi.

Ob'yektga mo'ljallangan yondashuv bir kunda o'ylab topilgan emas. Uning paydo bo'lishi dasturiy ta'minotning tabiiy rivojida navbatdagi pog'ona, xolos. Vaqt o'tishi bilan qanday uslublar ishlash uchun qulay, qaysinisi noqulay ekanini aniqlash oson bo'lib bordi. OMY eng muvaffaqiyatli, vaqt sinovidan o'tgan uslublarni o'zida mujassam etadi. Dastlab dasturlash anchayin boshqotirma ixtiro

bo'lib, u dasturchilarga dasturlarni kommutatsiya bloki orqali kompyuterning asosiy xotirasiga to'g'ridan-to'g'ri kiritish imkonini berdi. Dasturlar mashina tillarida ikkilik tasavvurda yozilar edi. Dasturlarni mashina tilida yozishda tez-tez xatolarga yo'l qo'yilar, kodni kuzatib borish amalda deyarli mumkin emas edi.

Bundan tashqari, mashina kodlaridagi dastur tushunish uchun g'oyat murakkab edi. Vaqt o'tishi bilan kompyuterlar tobora kengroq qo'llana boshlandi hamda yuqoriroq darajadagi protsedura tillari paydo bo'ldi. Bularning dastlabkisi FORTRAN tili edi. Biroq obyektga mo'ljallangan yondashuv rivojiga asosiy ta'sir keyinroq paydo bo'lgan. Protседura tillari dasturchiga axborotga ishlov berish dasturini pastroq darajadagi bir nechta protseduraga bo'lib tashlash imkonini beradi. Pastroq darajadagi bunday protseduralar dasturning umumiy tuzilmasini belgilab beradi. Ushbu protseduralarga izchil murojaatlar protseduralardan tashkil topgan dasturlarning bajarilishini boshqaradi.

Dasturlashning bu yangi paradigmasi mashina tilida dasturlash paradigmasiga nisbatan ancha ilg'or bo'lib, unga tuzilmashtirishning asosiy vositasi bo'lgan protseduralar qo'shilgan edi. Maydaroq funksiyalarni nafaqat tushunish, balki sozlash ham osonroq kechadi. Biroq boshqa tomondan protsedurali dasturlash koddan takroran foydalanish imkonini cheklab qo'yadi. Har bir protsedurama'lumotlarga kirish usullarini dasturlashi lozim bo'lganligi tufayli ma'lumotlar taqdimotining o'zgarishi dasturning ushbu kirish amalga oshirilayotgan barcha o'rinlarining o'zgarishiga olib kelar edi. Shunday qilib, hatto eng kichik to'g'rilash ham butun dasturda qator o'zgarishlar sodir bo'lishiga olib kelar edi. Modulli dasturlashda, masalan, Modula ikki kabi tilda protsedurali

dasturlashda topilgan ayrim kamchiliklarni bartaraf etishga urinib ko‘rildi. Modulli dasturlash dasturni bir necha tarkibiy bo‘laklarga yoki boshqacha qilib aytganda modullarga bo‘lib tashlaydi. Agar protsedurali dasturlash ma’lumotlar va protseduralarni bo‘lib tashlasa, modulli dasturlash, undan farqli o‘laroq ularni birlashtiradi. Modulma’lumotlarning o‘zidan hamda ma’lumotlarga ishlov beradigan protseduralardan iborat. Dasturning boshqa qismlariga moduldan foydalanish kerak bo‘lib qolsa, ular modul interfeysiga murojaat etib qo‘ya qoladi. Modullar barcha ichki axborotni dasturning boshqa qismlarida yashiradi.

Obyektga mo‘ljallangan dasturlash (OMD) modulli dasturlashdan keyingi mantiqiy pog‘onani egallaydi, u modulga nasldan naslga o‘tishni va polimorfizmni qo‘shadi. Dasturchi OMD dan foydalanar ekan, dasturni bir qator oliy darajali obyektlarga bo‘lish yo‘li bilan tizimlashtiradi. Har bir obyekt hal qilinayotgan muammoning ma’lum bir tomonini modellashtiradi. OMD endilikda dasturni bajarish jarayonini boshqarish uchun dasturchi diqqatini protseduralarni ketma-ketlikda chaqirib olish ro‘yxatini tuzib o‘tirishga qaratmaydi.

Buning o‘rniga obyektlar o‘zaro aloqada bo‘ladi. OMY yordamida ishlab chiqilgan dastur hal qilinayotgan muammoning amaldagi modeli bo‘lib xizmat qiladi. Dasturga obyektlar atamaları bilan ta’rif berish dasturiy ta’minotni ishlab chiqishning eng tushunarli usulidir. Obyektlar hamma narsani obyekt nima qilayotgani nuqtayi nazaridan idrok etishga, ya’ni uning xatti-harakatlarini xayolan modellashtirishga majbur qiladi. Shu tufayli obyektga yondashishda u dasturning bajarilishi jarayonida qanday ishlatiladi degan nuqtayi nazardan biroz e’tiborni chalg‘itishi mumkin. Shunday qilib, dasturni yozish jarayonida haqiqiy dunyoning tabiiy atamalaridan foydalanish mumkin. Dasturni alohida protseduralar va ma’lumotlar shaklida (kompyuter dunyosi atamalarida) qurish o‘rniga obyektlardan iborat dastur qurish mumkin.

Insoniyatning o‘z mehnatini yengillatishga muttasil intilishi ham informatsiyalar rolini oshirmoqda, uni qayta ishlash, tahlil va sintez qilish jarayonlarini hal qilishni taqozo etmoqda. Dunyoning turli nuqtalarida turib o‘zaro hamkorlik ishlarini olib borish, birgalikda yangi texnika va texnologiyalar ustida ish olib borish, umuman olganda o‘zaro tez axborot almashish anchagina muammolar tug‘dirardi. XX asrda qilingan eng katta ishlardan biri bu Internet texnologiyalarining vujudga kelishi bo‘ldi. Endi Internet yordamida insoniyat dunyoning qaysi burchagida faoliyat ko‘rsatishdan qat’iy nazar doimo bir-biri bilan aloqada bo‘lib turadi. Axborot tez almashishi o‘z navbatida inson faolligini ham oshirib yubordi va qisqa vaqtlar ichida juda ham katta yutuqlarga erishildi.

Keyingi yillarda mamlakatimiz ilm-fani ham axborotlashtirishning nazariy asoslariga katta hissa qo'shib kelmoqda, shu bilan birgalikda, hodisalar, jarayonlarni yagona axborot asosida tadqiq etishning ilmiy yo'nalishlarini tahlil va sintez qilish natijasi bo'lgan fan-informatikaning vujudga kelishiga boshlang'ich nuqta qo'yildi. Axborot, energiya, vazn, bo'shliq va vaqtni bir butun holda batafsil o'rganish hozirgi vaqtda inson hayotining barcha jabhalarida muhim ahamiyatga ega bo'lib qolmoqda.

Mamlakatimizda «Kadrlar tayyorlash Milliy dasturi»ning birinchi bosqichi yakunlanib, ikkinchi - sifat bosqichiga o'tildi. O'tgan davrda yaratilgan me'yoriy-xuquqiy hujjatlar fanlar bo'yicha o'quv adabiyotlarining yangi avlodini yaratishga asos bo'lib, o'quv jarayonini sifatini oshirishga xizmat qiladi. O'quv adabiyotlarining yangi avlodini yaratish, ularni tayyorlash borasidagi ilmiy-uslubiy, tashkiliy va iqtisodiy masalalarni hal qilish, uzloqsiz ta'lim tizimida Kadrlar tayyorlash milliy dasturi maqsadlariga erishishni ta'minlashga qaratilgan tadbirlarning ishlab chiqishni talab qiladi.

Axborot jamiyatni rivojlantiruvchi va uning tarakkiyotiga asos buluvchi muxim vosita xisoblanadi. Shu kabi axborot insoniyat tarixida eng muxim iqtisodiy kursatkichlardan biri bulsa, jamiyatni kompyuterlashtirish esa iqtisodiyotni tarkibiy jixatdan qayta qurishda asosiy xarakatlantiruvchi kuchdir. Jamiyatni axborotlashtirish, yangi axborot texnologiyalari bilan ta'minlash insonlarning turli-tuman ma'lumotlarga bo'lgan extiyojini qondirish muxim o'rin tutadi. Inson axborot olami ichra yasharkan, voqea-xodisalar va jarayonlarning bir-biriga aloqadorligini, o'zaro munosabati va mohiyatini taxlil etish, o'z xayotidan kelib chiqayotgan murakkab savollarga ilmiy javob topish maksadida ko'pdan-ko'p dalil va raqamlarga murojaat qiladi. Axborot tufayli nazariya amaliyot bilan birikadi. Amaliyot nazariyasiz, nazariya amaliyotsiz mavjud xam bulmaydi, rivojlanmaydi xam.

Xozirgi kunda xar bir tashkilot, o'quv muassasasi, firma va ishlab chiqarishning barcha soxalarida raxbar va xodimlar faoliyatining samaradorligini oshirish maksadida boshqaruv jarayonlarini ma'lum darajada avtomatlashtirishga oid muammolarni yechish bilan shug'ullanadi. Bunda ular maxsus firmalarning mutaxassislari bilan uchrashadi, ularning faoliyati bilan yaqindan tanishadi, ular ishlab chiqaradigan maxsulotlarni ko'radi va pirovardida o'zida avtomatlashtirish uchun kerak bo'ladigan texnik jixozlarni xarid qiladi. Albatta, tashkilotlarga o'rnatilgan avtomatlashtirish jixozlari yildan-yilga yangilanib, texnik jixatdan takomillashtirib boriladi.

XX asrning so'nggi o'n yili mobaynida axborotlar bilan ishlash va axborotlashtirish juda rivojlandi. Bunga sabab shundaki, kundalik turmushda axborotlar, ularni qayta ishlash va uzatishning ahamiyati ortib bormoqda. Bu esa

o'z navbatida jamiyatning xar bir a'zosidan axborotlashtirish va axborot texnologiyalari sirlarini, uning qoida va qonuniyatlarini mukammal bilishni takozo etadi.

C++ tili tarixi

Birinchi elektron hisoblash mashinalari paydo bo'lishi bilan dasturlash tillari evolyusiyasi boshlanadi. Dastlabki kompyuterlar ikkinchi jahon urushi vaqtida artilleriya snaryadlarining harakat traektoriyasini hisob-kitob qilish maqsadida qurilgan edi. Oldin dasturchilar eng sodda mashina tilini o'zida ifodalovchi kompyuter komandalari bilan ishlaganlar. Bu komandalar nol va birlardan tashkil topgan uzun qatorlardan iborat bo'lar edi. Keyinchalik, insonlar uchun tushunarli bo'lgan mashina komandalarini o'zida saqlovchi (masalan, ADD va MOV komandalari) assembler tili yaratildi.

SHu vaqtlarda BASIC va COBOL singari yuqori sathli tillar ham paydo bo'ldiki, bu tillar tufayli so'z va gaplarning mantiqiy konstruksiyasidan foydalanib dasturlash imkoniyati yaratildi. Bu komandalarni mashina tiliga interpretatorlar va kompilyatorlar ko'chirar edi. Interpretator dasturni o'qish jarayonida uning komandalarini ketma - ket mashina tiliga o'tkazadi. Kompilyator esa yaxlit programma kodini biror bir oraliq forma - ob'ekt fayliga o'tkazadi. Bu bosqich kompilyasiya bosqichi deyiladi. Bundan so'ng kompilyator ob'ektli faylni bajariluvchi faylga aylantiradigan kompanovka dasturini chaqiradi. Interpretatorlar bilan ishlash osonroq, chunki dastur komandalari qanday ketma - ketlikda yozilgan bo'lsa shu tarzda bajariladi. Bu esa dastur bajarilishini nazorat qilishni osonlashtiradi. Kompilyator esa kompilyasiya va kompanovka kabi qo'shimcha bosqichlardan iborat bo'lganligi uchun ulardan hosil bo'ladigan bajariluvchi faylni tahlil qilish va o'zgartirish imkoniyati mavjud emas. Faqatgina kompilyasiya qilingan fayl tezroq bajariladi, chunki

bundagi komandalar kompilyasiya jarayonida mashina tiliga o'tkazilgan bo'ladi.

C++ kabi kompilyasiya qiluvchi dasturlash tillarini yana bir afzalligi hosil bo'lgan dastur kompyuterda kompilyatorsiz ham bajarilaveradi. Interpretatsiya qiluvchi tillarda esa tayyor dasturni ishlatish uchun albatta mos interpretator dasturi talab qilinadi.

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 ta'minoti 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'lchamiga ega.

Butun son ko'rsatkichni saqlash uchun yetarli 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. O'zgaruvchilar uchun nomlantirish shart bo'lmasa-da, biroq qat'iy tavsiya etiladi. Lokal o'zgaruvchini nomlantirmay turib kiritishga deyarli asos yo'q. Ushbu turlarning har qanday kombinatsiyasiga quyidagi arifmetik operatsiyalarni qo'llash mumkin: (unar va binar plus) (unar va binar minus) * (ko'paytirish) / (bo'lish).

Shuningdek ularga quyidagi qiyoslash operatsiyalarini ham qo'llash mumkin:

== (teng)

!=(teng emas)

<(kamroq)

>(ko'proq)

<=(kamroq yoki teng)

>=(ko'proq yoki teng)

C++ tili va ob'ektlarga mo'ljallangan dasturlash.

C++ tili ob'ektga mo'ljallangan dasturlash prinsiplarini qo'llab quvvatlaydi.

Bu prinsiplar quyidagilardir:

- Inkapsulyasiya
- Merosxo'rlik
- Polimorfizm

Inkapsulyasiya.

Agarda muhandis ishlab chiqarish jarayonida rezistorni qo'llasa, u buni yangidan ixtiro qilmaydi, omborga (magazinga) borib mos parametrlarga muvofiq kerakli detalni tanlaydi. Bu holda muhandis joriy rezistor qanday tuzilganligiga e'tiborini qaratmaydi, rezistor faqatgina zavod xarakteristikalariga muvofiq ishlasa etarlidir. Aynan shu tashqi konstruksiyada qo'llaniladigan yashirinlik yoki ob'ektni yashirinligi yoki avtonomligi xossasi inkapsulyasiya deyiladi. Inkapsulyasiya yordamida berilganlarni yashirish ta'minlanadi. Bu juda yaxshi xarakteristika bo'lib foydalanuvchi o'zi ishlatayotgan ob'ektning ichki ishlari haqida umuman o'ylamaydi

Merosxo'rlik

C++ tili merosxo'rlikni himoya qiladi. Bu yangi berilganlar tipi (sinf), oldindan

bo'lgan sinfni kengaytirishdan hosil bo'ladi. Bunda yangi sinf oldingi sinfning

merosxo'ri deb ataladi.

Polimorfizm.

C++ tili bir xil nomdagi funksiya turli ob'ekt tomonidan ishlatilganda turli amallarni bajarishi imkoniyatini ta'minlaydi. Bu funksiya va sinfning polimorfligi deb nomlanadi. Poli – ko'p, morfe – shakl degan ma'noni anglatadi. Polimorfizm – bu shaklning ko'p xilligidir.

C++BUILDER ASOSLARI

Ishlab chiqishning integratsiyalashgan muhiti komponentalar palitrasini birlashtiradi. Shakllar muharriri, kod muharriri, obyektlar noziri, obyektlar xazinasini bular hammasi kod va zaxiralar ustidan to'liq nazoratni ta'minlovchi dasturiy ilovalarni tez ishlab chiqish instrumentlari.

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.

Obyektlar Xazinasini ma'lumotlarning shakl va modullari kabi obyektlarga ega bo'lib, ular ishlab chiqishda muvaqqat sarflarni kamaytirish maqsadida ko'plab ilovalar bilan bo'linadi. C++ Builder ilovalarni qurishning vizual metodikasini Komponentalar palitrasidan kerakli boshqarish elementlarini tanlab olish vositasida joriy etadi. Har bir komponenta bilan ushbu komponenta turini va xulq-atvorini o'zgartiradigan xususiyatlar bog'liq bo'ladi. C++ Builder 32 razryadli

takomillashtirilgan Vizual komponentalar kutubxonasi VCL (Visual Component Library) bilan birgalikda yetkazib 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. Kutubxona Windows va Windows 95

operatsiya tizimlaridagi Foydalanuvchi Grafik Interfeysi standart interfeys obyektlarining to'liq inkapsulatsiyalanishini o'z ichiga oladi. Ular orasida, ixtisoslashgan komponentalar bilan bir qatorda, relyatsion ma'lumotlar bazasini boshqarish uchun mo'ljallangan komponentalar alohida o'rin egallaydi. Ishonchli va samarali dasturlarni yaratishda C++Builder obyektga mo'ljallangan dasturlash (OMD) imkoniyatlaridan to'liq foydalanadi.

C++Builder bosh xususiyati avvalambor 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 biroz 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 funksional jihatdan o'xshash komponentalarni birlashtiradigan qo'shimcha ilovalar bilan bo'lingan. Tanlab olingan komponentaning kontekst menusini unga sichqonchaning o'ng tugmasini bosib ochish mumkin.

Standart komponentalar

TMainMenu

TMainMenu bosh menu buyruqlari panelini va ularga mos keladigan tushib qoladigan menularni yaratadi. Barcha menu buyruqlarining identifikatorlari menuning har qanday konkret buyrug'iga kirish huquqiga ega bo'lgan Items xususiyati bilan aniqlanadi, AutoMerge xususiyati Merge va Unmerge metodlari bilan birgalikda turli shakldagi menularning birlashish jarayonini boshqaradi.

TPopupMenu

TPopUpMenu shakl yoki bironta boshqa komponenta uchun maxsus menu yaratadi. E'tiborga oling, aynan shu maqsad uchun har qanday boshqa komponenta PopUpMenu xususiyatiga ega bo'lib, bu xususiyatda siz bilan bog'liq menuga iqtibos qilishingiz mumkin. Agar siz sichqonchaning o'ng tugmasini shaklga yoki berilgan komponenta mansub bo'lgan biron boshqa elementga bosish bilan maxsus menu ekranda paydo bo'lishini xohlasangiz, AutoPopup xususiyatining true qiymatini o'rnatib. Voqea qayta ishlatgichi OnPopup yordamida bevosita maxsus menuning paydo bo'lishi oldidan bajariladigan protsedurani aniqlash mumkin.

TLabel

TLabel shaklda tahrir qilib bo'lmaydigan statik 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 AutoSize xususiyatining true qiymatini o'rnatib. Kalta soha ichida matnning hammasini ko'rish imkoniga ega bo'lish uchun WordWrap xususiyatining true qiymatini berib. Transparent xususiyatining true qiymatini

oʻrnatilgan, boshqa komponentaning bir qismini toʻgʻri uning ustida joylashtirilgan nom orasidan koʻrinib turadigan qilishingiz mumkin.

TEdit

TEdit axborot yakka satrining tahrir qilinayotgan kiritishidagi toʻrtburchak sohani shaklda aks ettiradi. Tahrir sohasining ichidagi boshlangʻich narsalarni Text xususiyatining qiymati boʻlgan satr aniqlaydi.

TEdit komponentasi TCustomEdit sinfining toʻgʻridan-toʻgʻri hosilasi boʻlib, uning

barcha xususiyatlari, metodlari va voqealariga vorislik qiladi.

TMemo

TMemo axborot koʻplab satrining tahrir qilinayotgan kiritishidagi toʻrtburchak sohani shaklda aks ettiradi. Tahrir sohasining ichidagi boshlangʻich narsalarni Lines xususiyatining qiymati boʻlgan satrlar massivi aniqlaydi. Ushbu xususiyat qiymati ustunida tugmachani bossangiz, roʻyxat elementlari muharririning darchasi ochiladi.

TMemo komponentasi TCustomMemo sinfining toʻgʻridan-toʻgʻri hosilasi boʻlib, uning barcha xususiyatlari, metodlari va voqealariga vorislik qiladi.

TButton

TButton yozuvli toʻrtburchak tugmani yaratadi. Tugmacha bosilganda, dasturda biror bir xatti-harakat nomlanadi (initsiallashtiriladi).

Tugmachalar koʻproq dialogli darchalarda qoʻllanadi. Default xususiyatining true qiymati tomonidan tanlab olingan yashirin tugmacha, dialog darchasida har gal Enter klavishi bosilganda, OnClick voqea qayta ishlatgichini ishga tushiradi. Cancel xususiyatining true qiymati tanlab olgan uzish tugmachasi, dialog darchasida har gal Escape klavishi bosilganda, OnClick voqea qayta

ishlatgichini ishga tushiradi. TVutton komponentasi TButtonControl sinfining hosilasi hisoblanadi.

TCheckBox

TCheckBox ikkita holatga hamda tavsifiy matnga ega bo'lgan kvadrat chekboksni yaratadi (bunda tavsifiy matn chekboksning vazifasini spetsifikatsiya qiladi). Boks holatini bildiruvchi «check» biror bir variantning tanlanishiga mos keladi (boks ustidan tortilgan chiziq 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. TCheckBox komponentasi TButtonControl sinfining hosilasidir.

TRadioButton

TRadioButton ikkita holatga hamda tavsifiy matnga ega bo'lgan yumaloq tugmachani yaratadi (bunda tavsifiy matn yumaloq tugmachaning vazifasini spetsifikatsiya qiladi).

Radio-tugmalar bir-birini istisno qiladigan tanlov variantlarining to'plamidan iborat: ya'ni ushbu vaqt daqiqasida faqat bitta tugma tanlab olinishi mumkin (ichki qora doiracha bilan belgilanadi), avval tanlangan tugmadan esa tanlov avtomatik tarzda olinadi. Radiotugma bosilganda, Checked komponentasining xususiyati ham mos ravishda o'zgaradi va OnCkick voqeasi yuzaga keladi. Odatda radio-tugmalar avvaldan shaklda o'rnatilgan konteyner ichigajoylashtiriladi. Agar bitta tugma tanlangan bo'lsa, ushbu guruhga mansub barcha boshqa tugmalarning tanlovlari avtomatik tarzda olib tashlanadi.

TComboBox

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 dasturning bajarilish paytida tanlab olinishi mumkin bo'lgan element raqami ItemIndex xususiyatining, 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.

TScrollBar

TScrollBar darcha, shakl yoki boshqa komponenta ichidagilarini ko'rib chiqish uchun, masalan, biror bir parametr qiymatini berilgan interval ichida harakatlanishi uchun yugurgichli aylantirish lineykasini yaratadi.

Aylantirilayotgan obyekt xulq-atvorini OnScroll voqealar qayta ishlatgichi aniqlaydi. Foydalanuvchi lineykaning o'zida sichqonchani bosganda (yugurgichning har ikkala tomonida), yugurgich qanchaga surilishi kerakligini LargeChange xususiyatining qiymati aniqlab beradi.

Foydalanuvchi sichqonchani strelkali tugmachalar (lineyka oxiridagi) ustida bosganda yoki pozitsiyalash tugmachalarini bosganda, yugurgich qanchaga surilishi kerakligini SmallChange xususiyatining qiymati aniqlab beradi.

Min va Max xususiyatlarining qiymatlari yugurgichning yo'l qo'yilishi mumkin bo'lgan joy almashinuvlari intervallarini belgi-

laydi. Sizning dasturingiz yugurgichni Position xususiyatining qiymati aniqlab beradigan kerakli pozitsiyaga joylashtirishi mumkin.

SetPcirums metodi bir paytning o'zida Min, Max va Position ga tegishli barcha xususiyatlar qiymatlarini aniqlab beradi.

TPanel

TPanel boshqa komponentalarni o'z ichiga olishi mumkin bo'lgan bo'sh panelni yaratadi. Siz TPanel dan o'z shaklingizda instrumentlar paneli yoki holatlar satrlarini yaratish uchun foydalanishingiz mumkin.

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

Windows komponentalari Windows komponentalari sizning dasturingizga Windows ning 12 ta interfeys elementlarining ulanishini amalga oshiradi.

TTabControl

TTabControl bir-birini qisman yopib turadigan qo'shimcha kartoteka elementlarining to'plamini aks ettiradi. Qo'shimcha ilovalarning nomlari Tabs xususiyatining ro'yxatiga ushbu xususiyat qiymati ustunidagi tugmacha bilan kiritiladi.

TPageControl

TPageControl ko'p varaqli dialogni tashkil etish uchun mo'ljallangan qisman bir-birining ustini yopuvchi qo'shimcha kartoteka ilovalari ko'rinishidagi maydonlar to'plamini aks ettiradi. Tegishli qo'shimcha ilovaga ega bo'lgan yangi dialogli sahifani yaratish uchun berilgan komponentaning kontekst menusidan New Page opsiyasini tanlab oling. Siz konkret sahifani quyidagi usullarning biri yordamida faollashtirishingiz mumkin: ActivPage xususiyatining tushib qolayotgan ro'yxatidan tanlab olingan sichqoncha yordamida shuningdek, kontekst menuning NextPage va Previous Page opsiyalari yordamida qo'shimcha ilovalarni varaqlash vositasida. PageIndex

xususiyati faol sahifa raqamiga ega. Tab Visible xususiyatining false qiymatini o'rnatib, shu sahifani ko'rinmas qilish mumkin.

TImageList

TImageList bir xil o'lchamdagi grafik tasvirlardan iborat kolleksiya uchun konteyner yaratadi. Bu tasvirlarning har bittasini ularning 0 dan n-1 gacha qiymatlar intervalidagi indeks bo'yicha tanlab olish mumkin. Grafik kolleksiyalar bitli obrazlar yoki piktogrammalarning katta to'plamlariga samarali xizmat ko'rsatish uchun qo'llanadi. Ushbu bitli obrazlar va piktogrammalar eni k*n bo'lgan yagona bitli obraz sifatida saqlanadi. Niqobli monoxrom tasvirlar tiniq trafaretlar sifatida aks ettiriladi. TImageList komponentasi saqlanayotgan tasvirlarni yozish, tanlash va chiqarish jarayonlarini osonlashtiruvchi metodlarga ega. THeaderControl THeaderControl dasturning bajarilish jarayonida enini o'zgartirish mumkin bo'lgan ustunlar sarlavhalarining to'plami uchun konteyner yaratadi. Sections xususiyatida sanab o'tilgan sarlavhalarni axborot maydonlari ustida, masalan, TListBox komponentalari ro'yxatlari ustida joylashtirish mumkin. Sarlavhali seksiyalar muharririning darchasi ushbu xususiyat qiymatlarining grafasida tugmacha bilan ochiladi.

TRichEdit

TRichEdit RTF (Rich Text Format) formatidagi ko'p satrli axborotning tahrir qilinadigan kiritish sohasini aks ettiradi. ATF formati shrift atributlari va paragraflarni formatlashning turli variantlarini o'z ichiga oladi. Bu formatni ko'plab professional matniy protsessorlar masalan: MicrosoftWord qabul qiladi.

TBitBtn

TBitBtn bit obrazining tasviri tushirilgan tugmachani yaratadi. Bunday tugmachalar ko‘proq maxsus dialogli darchalarda qo‘llanadi. Grafik tugmachalar bit obrazlari, ularning ko‘rinishi va tugmachada joylashishini spetsifikatsiyalash xususiyatlariga ega bo‘ladi. Siz C++Builder qurilmasiga kirgan alohida tasvirlar katalogidagi grafik tugmalarning tayyor stillaridan foydalanishingiz hamyoki bo‘lmasa tasvirlarni tahrir qilish tizimlarining biri tomonidan yaratilgan suratlardan foydalanishingiz ham mumkin.

TSpeedButton

TSpeedButton odatda ma’lum menu buyruqlarini tez chaqirish yoki rejimlarni o‘rnatish paneli (TPanel) da joylashtiriladigan grafik tugmani yaratadi.

Tezkor tugmaning turli holatlariga (masalan, «bosilgan», «qo‘yib yuborilgan», «taqiqlangan» va h.k.) turli bit obrazlari mos kelishi mumkin. Bir-birining o‘rnini bosadigan tasvirlar va yozuv matnini tanlash uchun xususiyatlar mavjud. Kengayishli tasvirlar fayllari muharririning darchasi Glyph xususiyati qiymatlarining grafasidagi tugma bilan ochiladi. Tez tugmalarning boshqa xususiyatlari ularning biror bir guruhdagi ishini tashkil etadi.

TMaskEdit

TMaskEdit o‘ziga xos formatdagi ma’lumotlarning tahrir qilinadigan nazoratdagi to‘rtburchak sohasini yaratadi. Kiritilayotgan matnning to‘g‘riligi ruxsat etilgan formatlarni kodlovchi niqob vositasida tekshiriladi. Bu formatlarga matn kiritilgan va foydalanuvchiga taqdim etilgan bo‘lishi mumkin (sana, vaqt, telefon raqami va h.k.). EditMask xususiyati joriy niqob kodini saqlaydi. Niqoblar muharriri darchasi ushbu xususiyat qiymatlari grafasida tugma bilan

ochiladi. TMaskEdit komponentasi TCustomMaskEdit sinfining to'g'ridanto'g'ri hosilasidir. U satrlar yoki ustunlar bo'yicha belgili ketmaketliklarni aks ettirish uchun mo'ljallangan muntazam (regulyar) to'rni yaratadi.

TStringGrid

TStringGrid komponentaga tegishli barcha xususiyatlarning nomlari va vazifalari bo'lib, siz ulardan dasturni loyihalash bosqichida to'la foydalanishingiz mumkin. Ular keyingi paragrafda tavsifi berilgan TDrawGrid komponentasi xususiyatlariga to'liq to'g'ri keladi.

Simvolli ketma-ketliklar bilan bog'liq barcha obyektlar kerakli obyektga murojaat qilish imkonini beradigan Objects xususiyatida mujassam bo'lgan. Dastur bajarilish paytida simvolli ketma-ketliklar va setka ustunining ular bilan bog'liq obyektlari Cols xususiyati bilan adreslanadi. Rows xususiyati setka satrlari bilan xuddi shunday ish tutish imkonini beradi. Setkaning barcha simvolli ketma-ketliklari setkaning kerakli uyasini adreslaydigan (manzillaydigan) Cells xususiyatida mujassamdir.

TImage

TImage shaklda grafik tasvir konteynerini yaratadi (bu bit obrazi, piktogramma yoki meta fayl bo'lishi mumkin).

Tasvirlar fayllari muharririning darchasi Picture xususiyati qiymatlari grafasidagi tugma bilan ochiladi. Konteyner o'z o'lchamlarini tasvirni to'liq sig'diradigan qilib o'zgartirishi uchun AutoSize xususiyatining true qiymatini o'rnatish. Kichikroq o'lchamdagi dastlabki tasvir butun konteynerga cho'zilib ketishi uchun Stretch xususiyatining true qiymatini o'rnatish.

Tasvirlar fayllarining dinamik yuklanishi va saqlanishi uchun Picture

obyekt xususiyatining LoadFromFile va SaveToFile metodlaridan quyidagi turlar yordamida foydalaning:

Image->Picture->LoadFromFile(«<fayl nomi>»);

Image->Picture->SaveToFile(«<fayl nomi>»);

TScrollBar

TScrollBar darchada o'lchamlari o'zgaruvchan boksni yaratadi, bu boks shu topdayoq avtomatik tarzda zaruratga ko'ra aylantirish lineykalari bilan ta'minlanadi.

Aylantirib ko'rish boksi yordamida darchaning ayrim sohalarini aylantirib ko'rishdan himoyalash mumkin. Masalan, instrumentlar paneli va holat panelini himoyalash uchun avval darchani aylantirish lineykasini berkitib qo'ying, keyin esa aylantirish boksini mijoz sohasida instrumentlar paneli va holat paneli o'rtasida joylashtiring. Boksni aylantirib ko'rish lineykasi darchaga tegishli bo'lib ko'rinadi, biroq aylantirish faqat boks ichida amalga oshiriladi.

Aylantirib ko'rish bokslaridan yana boshqachasiga ham foydalanish mumkin: ular biror bir darchada ko'plab aylantirib ko'rilayotgan sohalar (turlar) yaratish imkonini beradi. Turlar ko'pincha tijoriy matn protsessorlarida, buxgalteriya dasturlarida va loyihalarni rejalashtirish dasturlarida qo'llanadi. Aylantirib ko'rish boksi boshqa komponentlarga, masalan, Tbutton va TCheckBox ga ham ega bo'lishi mumkin.

Asosiy qism.

3.1 Ma'lumotlar bazasini yaratish.

Har qanday dasturning o'z ma'lumotlar ombori bo'ladi. Ma'lumotlar ombori ning ham turlicha turlari mavjud: Oracle, MySQL, MS Access va boshqalar. Dasturni tuzishda men SQL tiliga asoslangan Paradox ma'lumotlar omboridan foydalandim. Dastur uchun quyidagi jadvallarni tuzib oldim:

- Talaba;
- Kitob;
- Biriktirish;
- MyVariables.

Jadval va undagi ustun nomlarini qo'yishda apostrof(') belgisidan foydalanmaslik maqsadga muvofiq. Aks holda dastur davomida bazaga turli xil Querylar yuborganimizda xatoliklar sodir bo'lishi mumkin. Keltirilgan jadvallarda quyidagi ma'lumotlar saqlanishi ko'zda tutilgan.

Talaba jadvali

Ushbu jadvalda foydalanuvchi talabalarning ID, familiyasi, ismi, otasining ismi, manzili, telefon raqami, guruhi kabi ma'lumotlar saqlanadi.

Kitob jadvali

Ushbu jadvalda ARM dagi kitoblarning ID, nomi, muallifi, chop etilgan yili, jami soni, shundan foydalanuvchilarga berilgan kitoblar soni va ARM da qolgan kitoblar soni kabi ma'lumotlar saqlanadi.

Biriktirish jadvali

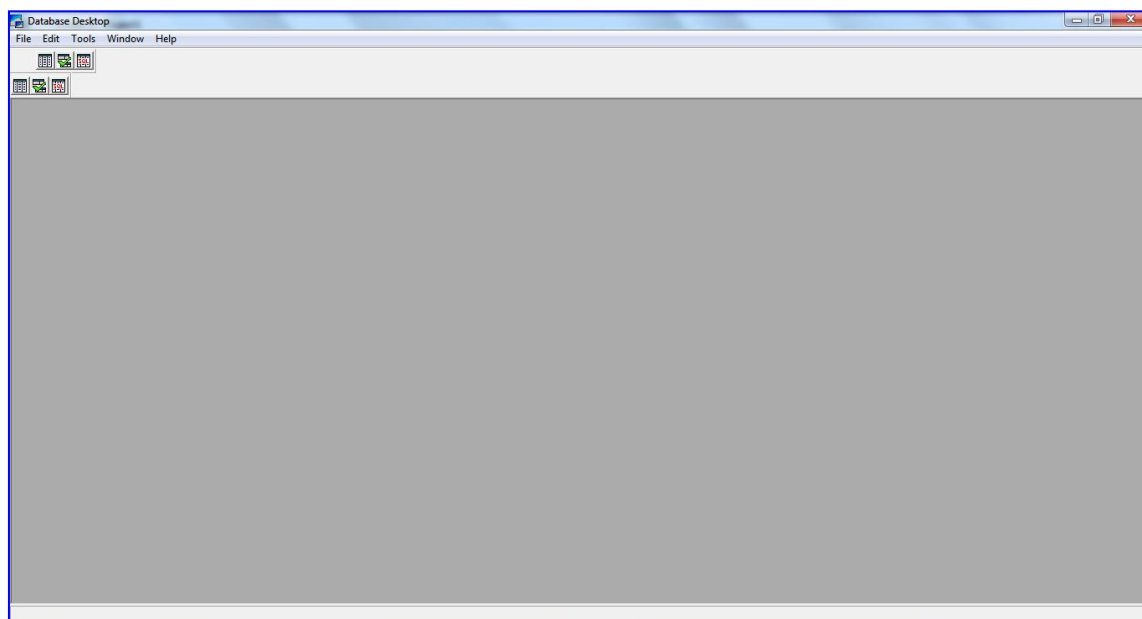
Ushbu jadvalda kitobdan foydalanayotgan foydalanuvchilarning va shu kitobning ID si hamda kitob berilgan vaqt to'g'risidagi ma'lumotlar saqlanadi.

MyVariables jadvali

Ushbu jadvalda dastur foydalanuvchisining login va parol ma'lumotlari saqlanadi.

Ushbu jadvallar quyidagi ketma-ketlikda quriladi.

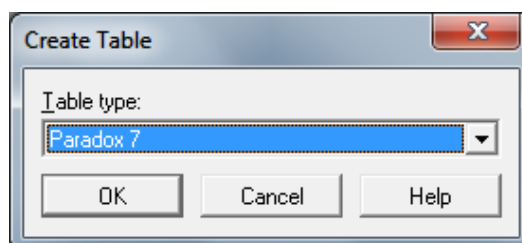
C++ Builder 6 dasturi tarkibidagi Database Desktop dasturi ishga tushiriladi.



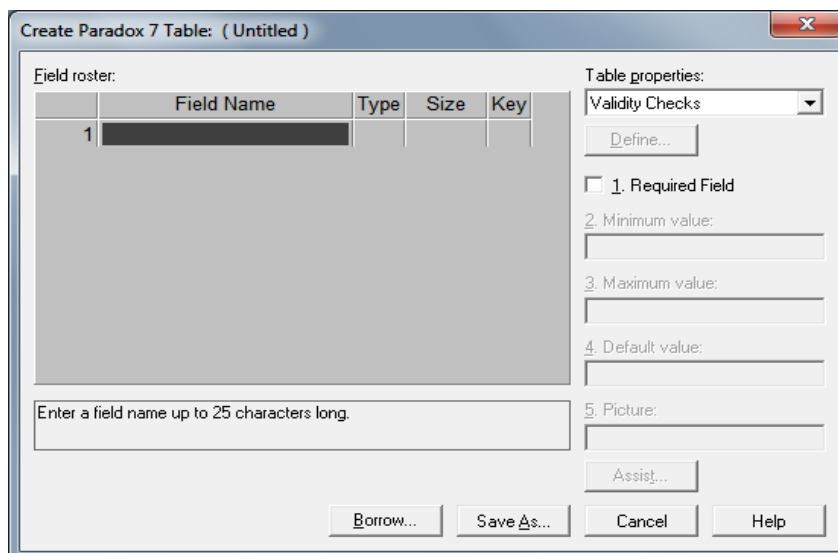
Quyidagi ketma-ketlikdagi buyruqlar beriladi:

File > New > Table.

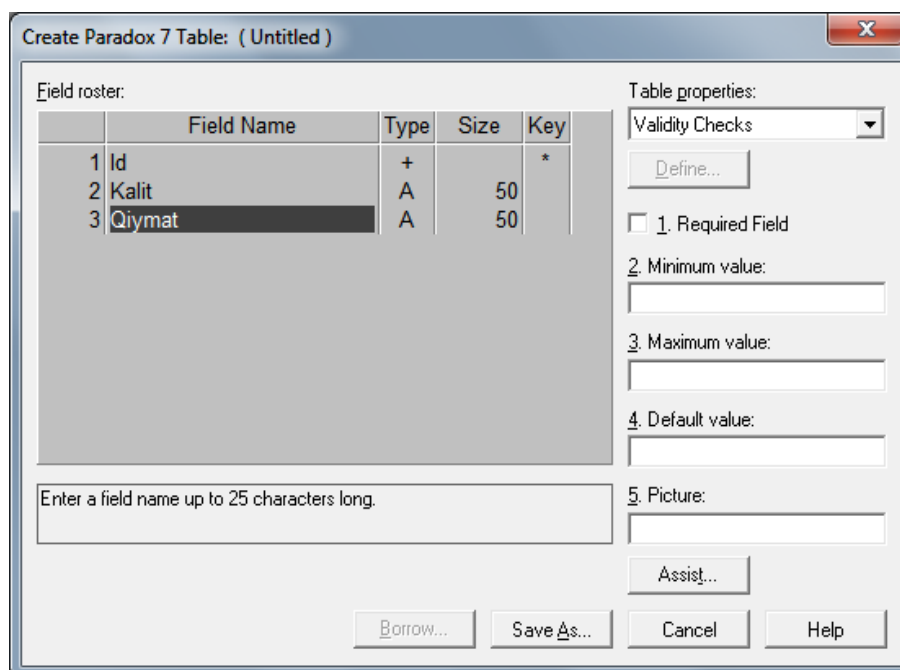
Natijada quyidagi oyna hosil bo'ladi.



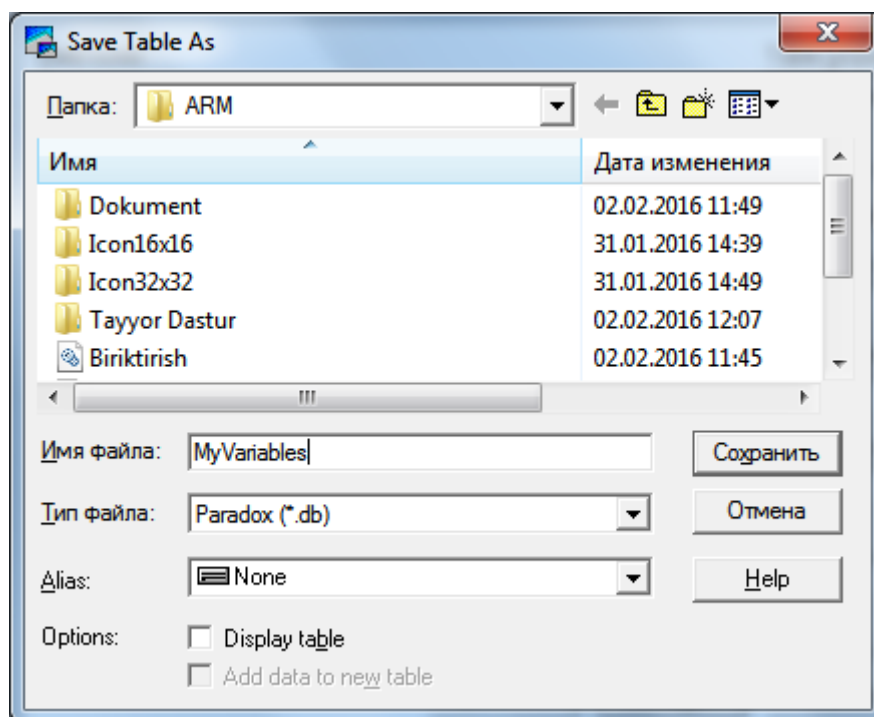
Paradox 7 ni tanlab OK tugmasini bosamiz.



Hosil bo'lgan oynaga kerakli maydonlarni kiritamiz.



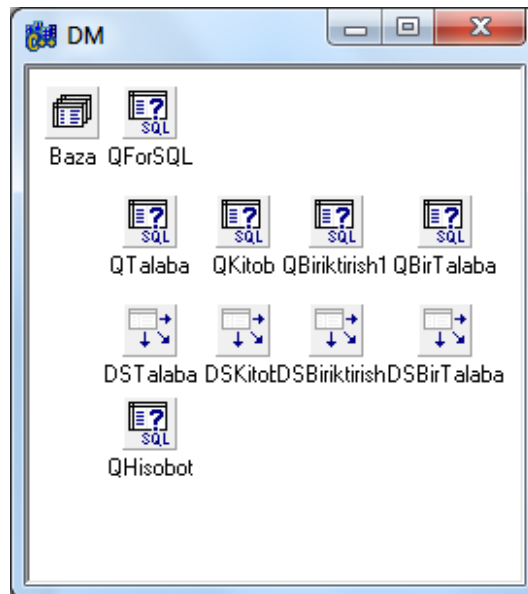
So'ngra "Save As..." tugmasini bosamiz. Natijada saqlash oynasi hosil bo'ladi. Jadvalni kerakli joyga saqlaymiz.



Shu tartibda kerakli jadvallarni kiritamiz.

3.2 Dastur loyihasini yaratish.

Dastur tuzishda kerak bo'lgan ma'lumotlar omborini tuzib oldik. Navbatdagi ishimiz dasturni yaratish. Dasturni yaratishda C++ Builder 6 dasturidan foydalandim. C++ Builder 6 dasturida yangi "Application" hosil qilib olamiz. So'ngra dasturni baza bilan bog'lash uchun Data Module formasini yaratamiz. Data Module oynasiga kerakli komponentlarni yuklaymiz: Database, Query, DataSource



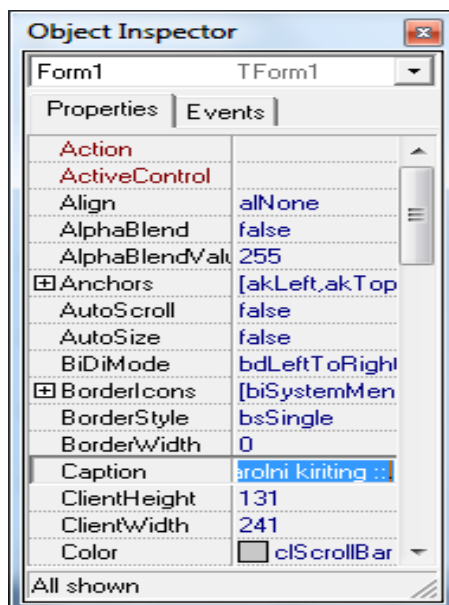
Paradox 7 da tuzilgan jadvallarni dasturga bog'lash uchun quyidagi ishlarni bajaramiz. Buning uchun DataBase komponentasini xususiyatlarini quyidagicha sozlaymiz:

DatabaseName	=	ARM
DriverName	=	STANDARD
Name	=	Baza
LoginPrompt	=	false
Connected	=	true

Shundan so'ng dastur o'zi kompilyatsiya bo'lgan papkadagi *.db kengaytmali jadvallar bilan bog'lanadi.

Query va DataSource komponentalarini tashlab DataBase xususiyatiga Database komponentasining DatabaseName xususiyatiga kiritgan nomimizni tanlaymiz.

Komponent hususiyatlarini Object Inspector oynasidan o'zgartiramiz.



3.3 Dasturdan foydalanish yo'riqnomasi.

Dastur ishga tushganda parol oynasi hosil bo'ladi.



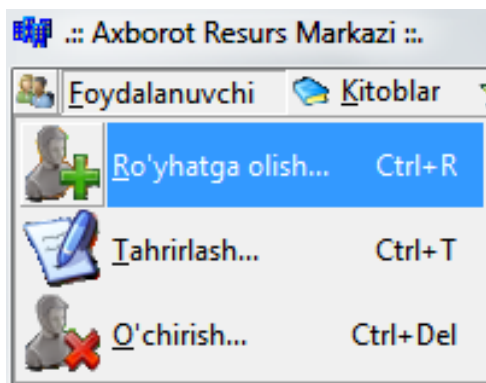
Kerakli login va parolni kiritamiz va asosiy oynaga o'tamiz.



Ushbu oyna dasturning asosiy oynasi hisoblanadi. Bu oyna quyidagi menyulardan iborat:

- Foydalanish
- Kitoblar
- Dastur

Foydalanish menyusining quyidagi SubMenu larga bo'linadi:



Ro'yhatga olish menuasi yangi foydalanuvchi (kitobxon) ni ro'yhatga olish oynasini ochib beradi:

The screenshot shows a dialog box titled 'Foydalanuvchilarni kiritish'. It has two columns of input fields. The left column contains fields for 'Familiya', 'Ismi', and 'Otasining ismi'. The right column contains fields for 'Guruhi', 'Manzili', and 'Telefon raqami'. At the bottom, there are two buttons: 'OK' with a green checkmark and 'Cancel' with a red X.

Kerakli ma'lumotlarni kiritib OK tugmasini bossak, yangi fodalanuvchi ro'yhatga olinadi.

Tahrirlash menusi kiritilgan foydalanuvchilarning ma'lumotlarni o'zgartiradi:

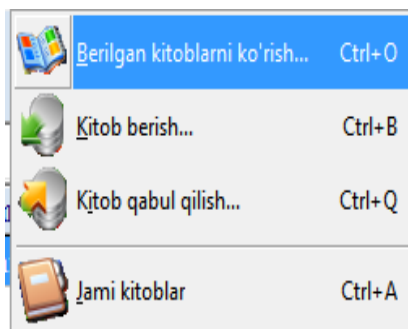
Dialog box titled "Foydalanuvchilarni tahrirlash :::" with the following fields:

Familiya		Guruhi	
Familiya		12345	
Ism		Manzili	
Ism		Farg'ona	
Otasining ismi		Telefon raqami	
Sharif		1234567	

Buttons: OK, Cancel

O'chirish tugmasi esa kiritilgan foydalanuvchilarni ma'lumotlarini o'chiradi.

Kitob menusi quyidagi Submenu lardan iborat.



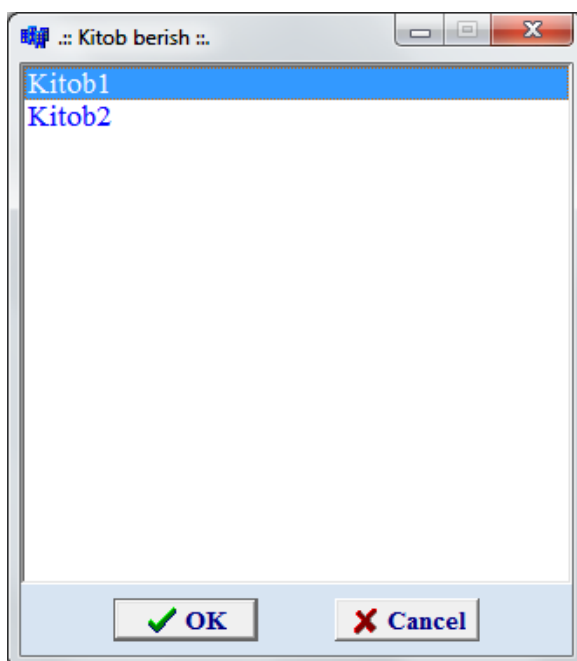
Berilgan kitoblarni ko'rish menyusi tanlab turilgan foydalanuvchi olgan kitoblarni ro'yhatini ko'rsatib beradi. (Foydalanuvchilar ro'yhatiga ham ikki marta chertsak (DbClick xususiyati) ushbu oyna ochiladi).

Dialog box titled "Foydalanuvchiga berilgan kitoblarni ko'rish :::"

Header: *Familiya Ismning ARMdanda qabul qilgan kitoblari*

Kitobning nomi	Muallifi	Chop etilgan yili	Kitob berilgan sana
Kitob1	Muallif1	2001	02.02.2016

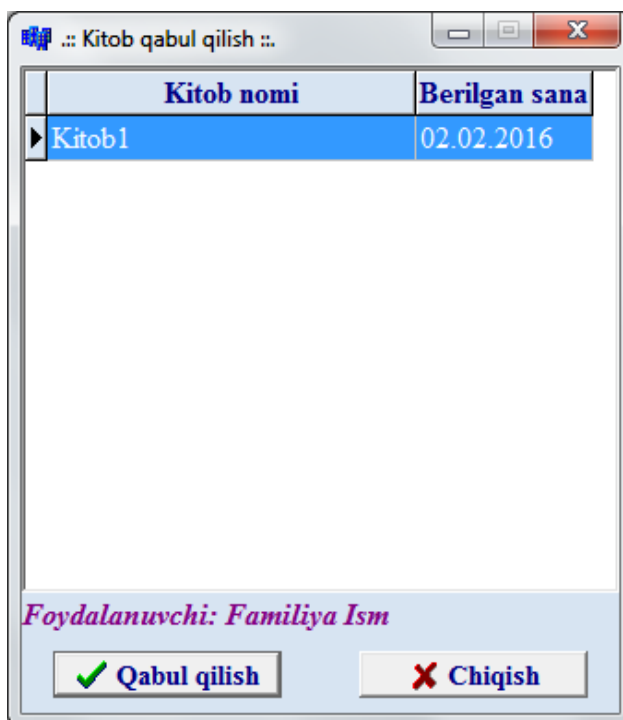
Kitob berish menyusi joriy foydalanuvchiga berilgan kitobni ro'yhatga olish oynasini ochib beradi:



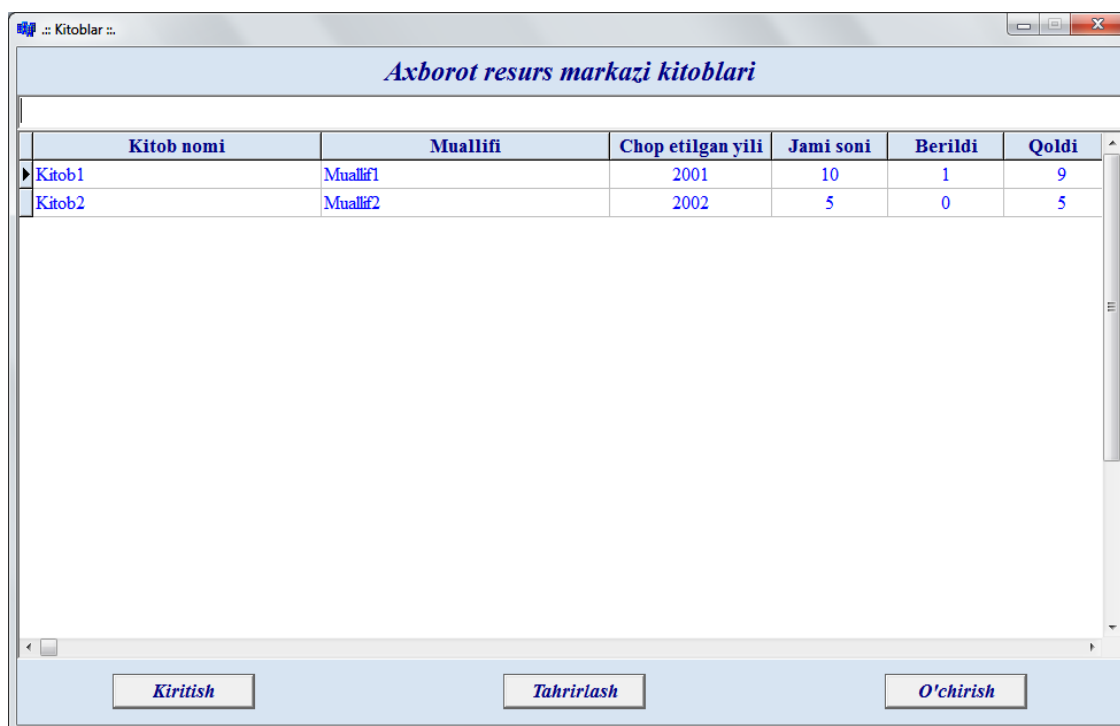
Kerakli kitobni tanlab OK tugmasini bosamiz.

Shundan so'ng tanlagan kitobimiz ushbu foydalanuvchiga biriktiriladi.

Kitob qabul qilish tugmasi orqali foydalanuvchiga biriktirilgan kitoblarni qaytadan ARM ga topshirilganligini saqlab qo'yamiz.



Jami kitoblar menyusidan ARM dagi kitoblar ro'yhati va ular ustida turli amallar bajaradigan oynani ochishimiz mumkin.



The screenshot shows a software window titled "Axborot resurs markazi kitoblari". Inside the window is a table with the following data:

Kitob nomi	Muallifi	Chop etilgan yili	Jami soni	Berildi	Qoldi
Kitob1	Muallif1	2001	10	1	9
Kitob2	Muallif2	2002	5	0	5

At the bottom of the window, there are three buttons: "Kiritish", "Tahrirlash", and "O'chirish".

Bu oynada kitoblarni kiritishimiz, tahrirlashimiz va o'chirishimiz mumkin.



The screenshot shows a dialog box titled "Kitob kiritish ::". It contains five text input fields with the following labels and values:

- Kitob nomi:** Kitob2
- Muallifi:** Muallif2
- Chop etilgan yili:** 2002
- Jami soni:** 5

At the bottom of the dialog box, there are two buttons: "OK" (with a green checkmark icon) and "Cancel" (with a red X icon).

Ro'yhatga sichqonchani chap tugmasini ikki marta tez chertsak joriy kitob qaysi foydalanuvchilarda ekanligini ko'rsatadi.

Kitob nomidan foydalanayotgan foydalanuvchilar ro'yhati

	Familiyasi	Ismi	Otasining ismi	Telefon raqami	Berilgan vaqti
▶	Familiya	Ism	Sharif	1234567	02.02.2016

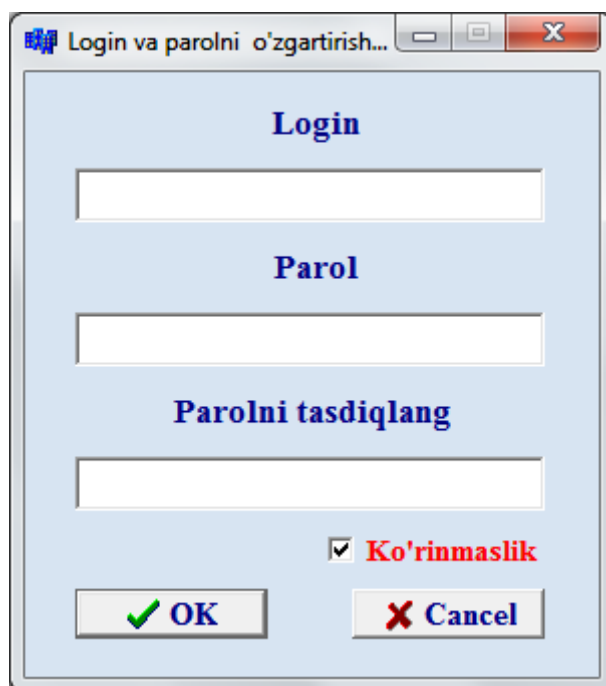
Dastur menyusidagi Hisobot menyusi foydalanuvchilarga berilgan kitoblarning hisobotini chiqarib beradi.

Foydalanuvchilarning kitoblardan foydalanish
RO'YHATI

Familiyasi	Ismi	Otasining ismi	Kitob nomi	Berilgan vaqti	Telefon raqami
Familiya	Ism	Sharif	Kitob1	02.02.2016	1234567
Familiya	Ism	Sharif	Kitob2	02.02.2016	1234567

Page 1 of 1

Dastur menyusining login va parolni o'zgartirish menyusi yordamida dastur foydalanuvchi login va parolini o'zgartirishimiz mumkin.



The image shows a standard Windows-style dialog box titled "Login va parolni o'zgartirish...". The dialog has a light blue background. It contains three text input fields, each preceded by a label: "Login", "Parol", and "Parolni tasdiqlang". Below the third field is a checkbox labeled "Ko'rinmaslik" with a red checkmark icon. At the bottom of the dialog are two buttons: "OK" with a green checkmark icon and "Cancel" with a red X icon.

Dastur menyusining Barcha ma'lumotlarni o'chirish tugmasi orqali ma'lumotlar bazasidagi barcha ma'lumotlarni o'chirishimiz mumkin.

Xulosa

Hammamizga ma'lumki, XXI asr "Axborot texnologiyalari asri" hisoblanadi. Bu asrda dasturlash sohasida misli ko'rilmagan o'zgarishlar bo'ldi. Bunda avtomatlashtirish dasturning asosini tashkil qiladi. Yuqorida yaratilgan dastur ham foydalanuvchiga turli qulayliklar yaratib, hisob ishlaridan ozod qiladi. Men dastur tuzish davomida C++Builderda keng imkoniyatlar yaratilganligini tushundim. Ayniqsa MySQL, Access kabi dasturlar bilan hamkorlikda ishlashi menda katta qiziqish uyg'otdi. Men bu kurs ishini qilish davomida C++Builder dasturi yordamida ma'lumotlar bazasini boshqarishni va dasturda turli interfeyslarda yaratishni, formalar ustida amallar bajarishni o'rgandim.

Men ushbu kurs ishini bajarish davomida C++ dasturlash tili da dasturlash hamda avtomatlashtirish to'g'risidagi bilimlarimni mustaxkamlab oldim. Dastur tuzish davomida C++ builder da ishladim. C++ builder ko'plab komponentalarni ishlatdim. Kurs ishini bajarish davomida dasturdagi ko'plab komponentalar palitrasidan, komponentalardan foydalanishni o'rgandim.

Foydalanilgan adabiyotlar

1. Гради Буч. Объектно-ориентированный анализ и проектирование с примерами приложений на С++. Невский диалект. 2001.
2. И. Грехем. Объектно-ориентированные методы. Принципы и практика. Вильямс. 2004.
3. Г.С.Иванова. Объектно-ориентированное программирование. Учебник. МГТУ им. Баумана. 2003
4. Т. Фейсон. Объектно ориентированное программирование на С++ 4.5. Киев. Диалектика. 1996.
5. www.Ziyonet.uz
6. www.google.uz

Ilovalar

Dasturdagi kodlardan misollar.

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

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

```
#pragma hdrstop
```

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

```
#include "Unit2.h"
```

```
#include "Unit1.h"
```

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

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

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

```
#include "Unit8.h"
```

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

```
#include "Unit9.h"
```

```
#include "Unit10.h"
```

```
#include "Unit12.h"
```

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

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

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

```
TForm3 *Form3;
```

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

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

```
    : TForm(Owner)
```

```

{
}

//-----

void __fastcall TForm3::FormClose(TObject *Sender, TCloseAction &Action)
{
    Form1->Close();
}

//-----

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

//-----

void __fastcall TForm3::Loginvaparolniozgartirish1Click(TObject *Sender)
{
    Form4->ShowModal();
}

//-----

void __fastcall TForm3::FormActivate(TObject *Sender)
{
    for (int i=0;i<6;i++)

```

```

{
    DBGrid1->Columns->Items[i]->Width=DBGrid1->Width/6-6;
}

DBGrid1->Columns->Items[3]->Width=DBGrid1->Columns->Items[3]->Width-
70;

DBGrid1->Columns->Items[4]->Width=DBGrid1->Columns->Items[4]-
>Width+70;
}

//-----

void __fastcall TForm3::Royhatgaolish1Click(TObject *Sender)
{
    Form5->Caption=".: Foydalanuvchilarni kiritish .:";
    Form5->EFam->Text="";
    Form5->EIsM->Text="";
    Form5->ESharif->Text="";
    Form5->EGuruh->Text="";
    Form5->EManzil->Text="";
    Form5->ETel->Text="";
    Form5->ShowModal();
}

//-----

void __fastcall TForm3::Ochirish1Click(TObject *Sender)
{
    if (DM->QTalaba->RecordCount==0)

```

```

{
    ShowMessage("O'chirish uchun ma'lumot yo'q...");
}

else

{
    int                                res=MessageDlg("Chindan                                ham
o'chirmoqchimisiz?",mtWarning,TMsgDlgButtons()<<mbYes<<mbNo,0);

    if (res==mrYes)
    {
        FoyID=DM->QTalaba->FieldByName("id")->AsString;
        DM->QForSQL->Active=false;
        DM->QForSQL->SQL->Clear();
        DM->QForSQL->SQL->Add("delete from Talaba where Id="+FoyID);
        DM->QForSQL->ExecSQL();
        DM->QTalaba->Active=false;
        DM->QTalaba->Active=true;

        //olgan kitoblari ro'yhatini o'chirish kerak

        DM->QBiriktirish1->SQL->Clear();

        DM->QBiriktirish1->SQL->Add("SELECT  biriktirish.talaba_id,  biriktirish.id,
biriktirish.kitob_id, kitob.id, kitob.berildi, kitob.qoldi FROM kitob INNER JOIN
biriktirish ON kitob.id = biriktirish.kitob_id where biriktirish.talaba_id="+FoyID);

        DM->QBiriktirish1->ExecSQL();

        DM->QBiriktirish1->Active=true;

        if (DM->QBiriktirish1->RecordCount!=0)
        {
            AnsiString KitobID,BirikID;

```

```

int ber, qol;
for (int i=0; i<DM->QBiriktirish1->RecordCount; i++)
{
    ber=DM->QBiriktirish1->FieldByName("berildi")->AsInteger-1;
    qol=DM->QBiriktirish1->FieldByName("qoldi")->AsInteger+1;
    KitobID=DM->QBiriktirish1->FieldByName("kitob_id")->AsString;
    BirikID=DM->QBiriktirish1->FieldByName("id")->AsString;
    DM->QForSQL->Active=false;
    DM->QForSQL->SQL->Clear();
    DM->QForSQL->SQL->Add("delete from biriktirish where id="+BirikID);
    DM->QForSQL->ExecSQL();
    DM->QForSQL->SQL->Clear();

    DM->QForSQL->SQL->Add("update          kitob          set
berildi="+IntToStr(ber)+",qoldi="+IntToStr(qol)+" where id="+KitobID);

    DM->QForSQL->ExecSQL();
    DM->QBiriktirish1->Next();
    DM->QKitob->Active=false;
    DM->QKitob->Active=true;
}
}
ShowMessage("Muvaffaqiyatli o'chirildi...");
}
}
}

//-----

```

```
void __fastcall TForm3::Tahrirlash1Click(TObject *Sender)
{
    if (DM->QTalaba->RecordCount==0)
    {
        ShowMessage("Tahrirlash uchun ma'lumot yo'q...");
    }
    else
    {
        Form5->Caption="::: Foydalanuvchilarni tahrirlash :::";
        Form5->EFam->Text=DM->QTalaba->FieldByName("fam")->AsString;
        Form5->EIsim->Text=DM->QTalaba->FieldByName("ism")->AsString;
        Form5->ESharif->Text=DM->QTalaba->FieldByName("sharif")->AsString;
        Form5->EGuruh->Text=DM->QTalaba->FieldByName("guruh")->AsString;
        Form5->EManzil->Text=DM->QTalaba->FieldByName("manzil")->AsString;
        Form5->ETel->Text=DM->QTalaba->FieldByName("tel_raqam")->AsString;
        FoyID=DM->QTalaba->FieldByName("id")->AsString;
        Form5->ShowModal();
    }
}

//-----
```

```
void __fastcall TForm3::Jamikitoblar1Click(TObject *Sender)
{
    Form6->ShowModal();
}
```

//-----

```
void __fastcall TForm3::Kitobbiriktirish1Click(TObject *Sender)
{
    if (DM->QTalaba->RecordCount==0)
    {
        ShowMessage("Foydalanuvchi kiritilmagan...");
    }
    else if (DM->QKitob->RecordCount==0)
    {
        ShowMessage("Kitob kiritilmagan...");
    }
    else
    {
        DM->QForSQL->Active=false;
        DM->QForSQL->SQL->Clear();
        DM->QForSQL->SQL->Add("select * from kitob where qoldi>0");
        DM->QForSQL->ExecSQL();
        DM->QForSQL->Active=true;
        if (DM->QForSQL->RecordCount==0)
        {
            ShowMessage("Barcha kitoblar foydalanuvchilarga berilgan...");
        }
    }
}
```

```

else
{
    Form8->NameList->Items->Clear();
    Form8->ListBox1->Items->Clear();
    Form8->BerildiList->Items->Clear();
    Form8->QoldiList->Items->Clear();
    for (int i=0; i<DM->QForSQL->RecordCount; i++)
    {
        Form8->NameList->Items->Add(DM->QForSQL->FieldByName("nomi")-
>AsString);
        Form8->ListBox1->Items->Add(DM->QForSQL->FieldByName("id")-
>AsString);
        Form8->BerildiList->Items->Add(DM->QForSQL->FieldByName("berildi")-
>AsString);
        Form8->QoldiList->Items->Add(DM->QForSQL->FieldByName("qoldi")-
>AsString);
        DM->QForSQL->Next();
    }
    FoyID=DM->QTalaba->FieldByName("id")->AsString;
    Form8->ShowModal();
}
}
}
//-----

void __fastcall TForm3::Kitobqabulqilish1Click(TObject *Sender)
{

```

```

if (DM->QTalaba->RecordCount==0)
{
    ShowMessage("Foydalanuvchi kiritilmagan...");
}
else
{
    FoyID=DM->QTalaba->FieldByName("id")->AsString;
    DM->QForSQL->Active=false;
    DM->QForSQL->SQL->Clear();

    DM->QForSQL->SQL->Add("select      *      from      biriktirish      where
talaba_id="+FoyID);

    DM->QForSQL->ExecSQL();
    DM->QForSQL->Active=true;
    if (DM->QForSQL->RecordCount==0)
    {
        ShowMessage("Bu foydalanuvchi kitob olmagan...");
    }
    else
    {
        AnsiString    SQLText="SELECT      biriktirish.id,      biriktirish.kitob_id,
biriktirish.talaba_id, biriktirish.sana, kitob.id, kitob.nomi, kitob.muallifi, kitob.yili,
kitob.jami, kitob.berildi, kitob.qoldi FROM talaba, biriktirish INNER JOIN kitob
ON biriktirish.kitob_id = kitob.id where talaba_id="+FoyID;

        DM->QBiriktirish1->Active=false;
        DM->QBiriktirish1->SQL->Clear();
        DM->QBiriktirish1->SQL->Add(SQLText);
        DM->QBiriktirish1->ExecSQL();
    }
}

```

```
DM->QBiriktirish1->Active=true;
```

```
Form9->Panel1->Caption="Foydalanuvchi: "+DM->QTalaba->FieldByName("fam")->AsString+" "+DM->QTalaba->FieldByName("ism")->AsString;
```

```
Form9->ShowModal();
```

```
}
```

```
}
```

```
}
```

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

```
void __fastcall TForm3::Korish1Click(TObject *Sender)
```

```
{
```

```
if (DM->QTalaba->RecordCount==0)
```

```
{
```

```
ShowMessage("Foydalanuvchi kiritilmagan...");
```

```
}
```

```
else
```

```
{
```

```
FoyID=DM->QTalaba->FieldByName("id")->AsString;
```

```
AnsiString SQLText="SELECT biriktirish.id, biriktirish.kitob_id, biriktirish.talaba_id, biriktirish.sana, kitob.id, kitob.nomi, kitob.muallifi, kitob.yili, kitob.jami, kitob.berildi, kitob.qoldi, talaba.fam, talaba.ism, talaba.sharif FROM (biriktirish INNER JOIN kitob ON biriktirish.kitob_id = kitob.id) INNER JOIN talaba ON biriktirish.talaba_id = talaba.id where talaba_id="+FoyID;
```

```
DM->QBiriktirish1->Active=false;
```

```
DM->QBiriktirish1->SQL->Clear();
```

```
DM->QBiriktirish1->SQL->Add(SQLText);
```

```
DM->QBiriktirish1->ExecSQL();
```

```

DM->QBiriktirish1->Active=true;

if (DM->QBiriktirish1->RecordCount==0)
{
    ShowMessage("Foydalanuvchi kitob qabul qilmagan...");
}
else
{
    Form10->Panel1->Caption=DM->QTalaba->FieldByName("fam")->AsString+"
"+DM->QTalaba->FieldByName("ism")->AsString+"ning ARMdan qabul qilgan
kitoblari";

    Form10->ShowModal();
}
}
}

//-----

void __fastcall TForm3::RichEdit1Change(TObject *Sender)
{
    if (RichEdit1->Text.Trim()== "")
    {
        DM->QTalaba->Filtered=false;
    }
    else
    {
        DM->QTalaba->Filtered=false;

        DM->QTalaba->Filter="fam = '"+AddSlash(RichEdit1->Text.Trim(), "'")+ "'";
    }
}

```

```
DM->QTalaba->Filtered=true;

}

}

//-----

void __fastcall TForm3::DBGrid1DblClick(TObject *Sender)
{
    Korish1->Click();
}

//-----

void __fastcall TForm3::Hisobot1Click(TObject *Sender)
{
    DM->QHisobot->Active=false;
    DM->QHisobot->Active=true;
    if (DM->QHisobot->RecordCount==0)
    {
        ShowMessage("Hisobot uchun ma'lumot yo'q...");
    }
    else
    {
        Form12->QuickRep1->Preview();
    }
}

//-----

void __fastcall TForm3::Barchamalumotlarniochirish1Click(TObject *Sender)
{
```

```
int res=MessageDlg("Chindan ham barcha ma'lumotlarni  
o'chirmoqchimisiz?",mtWarning,TMsgDlgButtons()<<mbYes<<mbNo,0);  
  
if (res==mrYes)  
{  
  
DM->QForSQL->Active=false;  
DM->QForSQL->SQL->Clear();  
DM->QForSQL->SQL->Add("delete from talaba");  
DM->QForSQL->ExecSQL();  
DM->QForSQL->SQL->Clear();  
DM->QForSQL->SQL->Add("delete from kitob");  
DM->QForSQL->ExecSQL();  
DM->QForSQL->SQL->Clear();  
DM->QForSQL->SQL->Add("delete from biriktirish");  
DM->QForSQL->ExecSQL();  
DM->QKitob->Active=false;  
DM->QKitob->Active=true;  
DM->QTalaba->Active=false;  
DM->QTalaba->Active=true;  
DM->QBiriktirish1->Active=false;  
DM->QBiriktirish1->Active=true;  
DM->QBirTalaba->Active=false;  
DM->QBirTalaba->Active=true;  
  
ShowMessage("Barcha ma'lumotlar o'chirildi.");  
  
}  
  
}
```