



**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:

620-14 guruh

NIYOZOVA M

Qabul qildi:

MUSAYEV X

Farg'ona 2015

**Mavzu: Omborda mahsulotlarni ro'yhatga olishni
avtomatlashtiruvchi dastur yaratish.**

Reja:

1. Kirish

2. Asosiy qism

2.1 C++ tarixi

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

2.3 C++ Builder asoslari

3. Amaliy qism

4. Xulosa

5. Foydalanilgan adabiyotlar

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'yektga mo'ljallangan yondashuv bir kunda o'ylab topilgan emas.

Uning paydo bo'lishi dasturiy ta'minotning tabiiy rivojidagi 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. Protsedura 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 tuzilmalashtirishning 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 protsedura ma'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. Modul ma'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.

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 (`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 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 sinfnig

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 sichqonchani o‘ng tugmasini bosib ochish mumkin.

Standart komponentalar

TMainMenu

TMainVlenu 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 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

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 voqealariga vorislik qiladi.

TMemo

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

TButton yozuvli to'rtburchak tugmani yaratadi. Tugmacha bosilganda, dasturda biror 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 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 OnClick 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.

TListBox

TListBox tanlash, qo'shish yoki o'chirish uchun mo'ljallangan matn variantlari ro'yxatining to'rtburchak sohasini aks ettiradi.

Agar ro'yxatdagi barcha elementlar ajratilgan sohaga sig'masa, ro'yxatni aylantirish lineykasi yordamida ko'rib chiqish mumkin.

Ro'yxat elementlari Items xususiyatining ichida, dastur bajarilish vaqtida tanlab olinadigan element raqami esa ItemIndex xususiyatining ichida joylashgan bo'ladi. Ro'yxat elementlari matn muharririning darchasi Items xususiyati qiymatining grafasida tugmacha bilan ochiladi. Ro'yxat elementlarini Items obyektining Add, Append, Delete va Insert metodlari yordamida dinamik tarzda qo'shish, o'chirish, orasiga joylash va o'rnini almashtirish mumkin.

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

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

TGroupBox

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 Windowsning bir nomdagi obyektning inkapsulatsiyalanishidan iborat.

TRadioGroup TRadioGroup to'g'ri burchakli ramka ko'rinishidagi konteyner bo'lib, u bir-birini mantiqan istisno qiladigan radio-tugmalar guruhini shaklda vizual birlashtiradi.

Radio-tugmalar bitta konteynerga joylashtirilganda «guruhlanadi».

Bu guruhdan faqat bitta tugmacha tanlab olinishi mumkin. TRadioGroup komponentasiga tugmalarni qo'shish uchun Items xususiyatining tahriri bajarilishi kerak. Items xususiyatining navbatdagi satriga nom berilsa, shu tugma guruhlovchi ramkada paydo bo'ladi. Ushbu daqiqada qaysi tugma tanlab olinishi kerakligini ItemIndex xususiyatining qiymati aniqlab beradi. Columns xususiyatining tegishli qiymatini joylashtirib, siz radiotugmalarni bir necha ustunga guruhlashingiz mumkin.

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.

TTabSheet

qurilma boshqarish komponentasi amalga oshiradi. Agar hamma qo'shimcha ilovalar bir qatorga sig'masa, komponenta aylantirish tugmalarini chiqaradi. Qo'shimcha ilovalarni bir necha qatorda

aks ettirish uchun MultiLine xususiyatining true qiymatini bering.

TTreeView

TTreeView elementlar tartibi tabaqaviy (shajaraviy) ko'rinishga ega bo'lgan maydonni aks ettiradi: hujjatlar sarlavhasi, ko'rsatkichdagi yozuvlar, disklardagi fayllar yoki kataloglar. Bu komponentaning ishini ko'plab Windows ilovalarida ko'rish mumkin.

Items xususiyati shajara elementlarining tahrir qilinayotgan ro'yxatiga ega bo'lgan TTreeNode obyektiga iqtibos (murojaat) qiladi.

TListView

TListView tabaqaviy (shajaraviy) elementlar ro'yxatiga ega bo'lgan maydonni aks ettiradi. ViewStyle xususiyati ro'yxat elementlarini quyidagi tartibda aks etishini belgilaydi: sarlavhali ustunlar bo'yicha, vertikal, gorizontal, katta yoki kichik piktogrammalar bilan.

Items xususiyati ost yozuvlarini qo'shish, o'chirish va modifikatsiyalash, shuningdek, ro'yxat elementlariga piktogrammalarni tanlash imkonini beradi. Ro'yxat muharriri ushbu xususiyat qiymatlari grafasidan tugmacha yordamida chaqiriladi.

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.

TStatusBar

TStatusBar dastur ishlayotgan paytda chiqarib beriladigan maqomli (statusli) axborotni aks ettirish uchun holatlar panellari satrini (bu satr odatda shaklning pastki chegarasi bo'ylab tekislanadi) yaratadi.

Har bir panel Panels xususiyatlari ro'yxatida o'z o'rniga ega.

Panellar 0 indeksidan boshlab, chapdan o'ngga qarab shakllan-

tiriladi. Panellar Muharririning darchasi ushbu xusu-

siyat qiymatlarining grafasida tugmacha bilan ochiladi. SimplePanel

xususiyatidan satr holatining aks etish turini (bir yoki ko'p panelli)

qayta ulash uchun foydalaniladi.

TTrackBar

TTrackBar belgilar va joriy holat regulyatori (rostlagichi) ga ega bo'lgan shkalani (aylantirish lineykasining varianti) yuzaga keltiradi. Min va Max xususiyatlari shkala qiymatlari intervalini o'rnatadi, bunda Position xususiyati regulyatorning berilgan interval ichidagi joriy pozitsiyasini aks ettiradi. Tasvirlanadigan belgilar soni Frequency xususiyatini spetsifikatsiyalaydi. Foydalanuvchi shkalaning o'zida (regulyatorning ikki tomonida) sichqonchani shiqillatganda yoki PageUp va PageDown klavishalarini bosganda, regulyator nechta belgiga surilishi kerakligini PageSize xususiyatining qiymati aniqlaydi. Foydalanuvchi kursorni pozitsiyalash klavishasini bosganda, regulyator nechta belgiga surilishi kerakligini LineSize xususiyatining ko'rsatkichi aniqlaydi. Shkala turishini o'zgartirish uchun TickStyle va TickMarks xususiyatlaridan foydalaning. SelStart va SelEnd xususiyatlarining qiymatlari regulyatorning ruxsat etilgan ko'chishlari chegaralarini o'rnatadi.

TProgressBar

TProgressBar sizning dasturingizdagi qandaydir protseduraning bajarilish jarayonini kuzatib boradi. Protsedura bajarilgan sayin, to'g'ri burchakli indikator chapdan o'ngga qarab asta-sekin berilgan rangga bo'yalib boradi.

Min va Max xususiyatlari indikator qiymatlari indikatorini o'rnatadi. Step xususiyati indikator pozitsiyasi o'zgarishi bilan har gal Position xususiyati qiymatining o'zgarish qadamini belgilaydi.

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.

TDrawGrid

TDrawGrid tuzilma holiga keltirilgan grafik ma’lumotlarni satrlar yoki ustunlar bo‘yicha aks ettirish uchun muntazam setka yaratadi. RowCount va ColCount xususiyatlari vertikal va gorizontal bo‘yicha setka uyalarining sonini belgilaydi.

Options xususiyatining qiymatlari setkaning turi (masalan, ustunlar orasida ajratuvchi chiziqlarga ega bo‘lgan setka turi) va uning xulq-

atvorini (masalan, ustundan ustunga Tab klavishasi bo'ylab o'tish) o'zgartirish imkonini beradi. Setkadagi ajratish chiziqlarining eng GridLineWidth xususiyati tomonidan belgilanadi, aylantirish chiziqchalari esa ScrollBars xususiyati tomonidan qo'shiladi. FixedCol va FixedRows xususiyatlari ustunlar va satrlarning aylantirilishini taqiqlab qo'yish imkonini beradi, FixedColor xususiyati esa barcha ustun va satrlarga ma'lum rang beradi.

DefaultDrawing xususiyatining true qiymati setka uyalarining ichidagilarini avtomatik tarzda chizib ko'rsatadi, bunda uning foni, asosi va rangi yashirin tanlanadi. DefaultDrawing xususiyatining false qiymatini o'rnatish hamda setka uyalarini «qo'lda» to'ldirish uchun mo'ljallangan OnDrawCell voqeasi qayta ishlatgichining yozilishini talab qiladi. DefaultColWidths va DefaultRowHeights xususiyatlari yordamida yashirin tanlanayotgan barcha ustunlar va satrlarning enini o'rnatish mumkin. ColWidth va RowHeight xususiyatlari konkret ustun enini va konkret satr bo'yini spetsifikatsiyalaydi.

Dastur ishlash paytida siz CellRect metodi yordamida biror bir uyaning rasmini chizish uchun ma'lum sohani o'z ixtiyoringizga olishingiz mumkin. MouseToCell metodi ustun raqami va sichqoncha kursori o'rnatilgan satr uyasining koordinatalarini qaytarib beradi. Setkaning tanlab olingan uyasi Selection xususiyatining qiymati bo'lib qoladi

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 foydalanish:

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

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

TShape

`TShape` aylana va ellips, kvadrat va to'g'ri to'rtburchak (burchaklarini yumaloqlash mumkin) kabi oddiy geometrik shakllarning rasmini chizadi.

Tanlab olingan geometrik shaklning turini `Shape` xususiyati, rang va bo'yash usulini `Brush` komponentasiga joylangan ikkita `Color` va `Style` xususiyatlari aniqlaydi. Shakllarning o'lchamlarini ham tegishli xususiyatlar aniqlaydi.

TBevel

`TBevel` xuddi iskana bilan o'yilgandek hajmli ko'rindigan chiziqlar, bokslar yoki ramkalarni yaratadi.

Komponenta chizayotgan obyektning `Shape` xususiyati aniqlaydi, `Style` xususiyatining qiymati esa obyekt ko'rinishini o'zgartirib, uni bo'rtiq yoki botiq holga keltiradi. Foydalanuvchi shakl o'lchamlarini o'zgartirganda ham obyektning nisbiy holatini o'zgarish qoldirish

uchun Align xususiyatining true qiymatini o'rnatish.

ScrollBar

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.

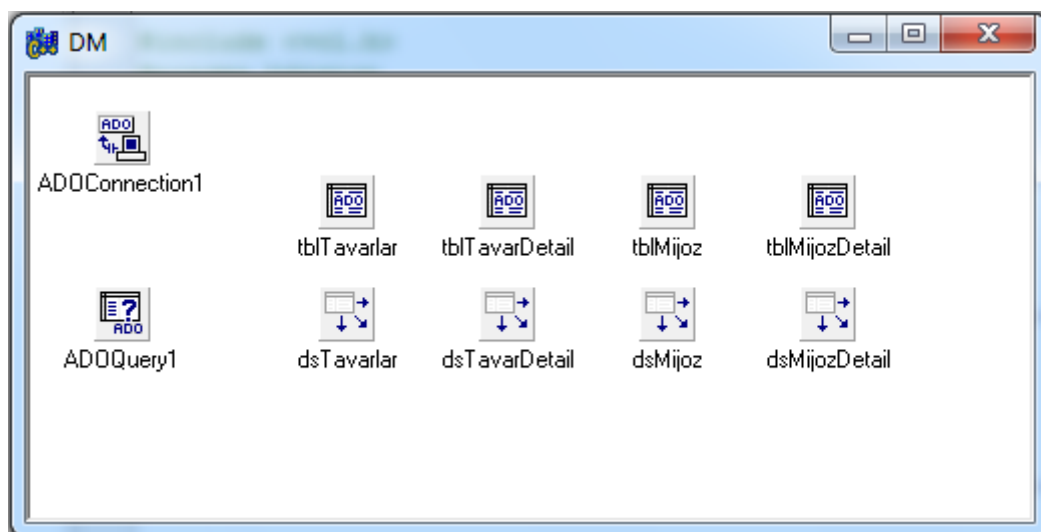
Aylantirib ko'rish bokslaridan yana boshqachasiga ham foydalanish mumkin: ular biror bir darchada ko'plab aylantirib ko'rilyotgan sohalar (turlar) yaratish imkonini beradi. Turlar ko'pincha tijoriy matn protsessorlarida, buxgalteriya dasturlarida va loyihalarni rejalashtirish dasturlarida qo'llanadi.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

Amaliy qism.

Dasturni yaratishda avvalo ma'lumotlar bazasini, undagi jadvallarni, jadvaldagi ustunlarni to'g'ri kiritish lozim. Jadval, ustun nomlarini yozishda probel, apostriv(') belgilaridan foydalanmaslik zarur.

Bolalar o'yinchoqlari do'koni avtomatlashtirilgan dasturini yaratishda oyinchoqbaza nomli baza tashkil qildim. Bazada myb, maxsulot va sotish nomli jadvallar yaratdim.

Ma'lumotlar bazasini to'liq tuzib olganimizdan so'ng dasturni yaratishni boshlaymiz. Dasturni yaratishda C++ Builder 6 dasturidan foydalandim. Dasturni ishga tushiramiz va C++ Builder Application Form yaratib olamiz. Bu asosiy forma hisoblanadi. Bazani dasturga bo'g'lash uchun Data Module formani hosil qilib olamiz. Bazaga bog'lash uchun ADOConnection komponentasidan, jadvaldagi ma'lumotlarni olish uchun ADOTable komponentalaridan, ADOTable komponentalaridagi ma'lumotlarni dasturga bo'g'lash uchun DataSource komponentasidan foydalanamiz. Data Modulenii foydalanishda qulaylik uchun dm deb nomladim.



Dastur ishlatilganda asosiy forma hosil bo'ladi. Men asosiy formaga quyidagicha dizaynda hosil qildim.

Ombor_only key_believable

Tovarlar **Omborchilar** **Hisobot**

TavarTuri

► Oziq-ovqat

Kiyim -kechak

TavarNomi	TavatNarxi	TavarV.	OK
► non	1000	02.02.2014 23:20:08	
tvarog	2000	02.02.2014 23:20:08	
sut	1000	02.02.2014 23:20:08	
qatiq	1800	05.02.2014 23:20:08	
go'sht	20000	01.02.2014 23:20:08	
kolbasa	15000	30.01.2014 23:20:08	
tuxum	5000	08.02.2015 23:20:08	

FIO

► Aliyev Jaloliddin

Manzili

Farg'ona shahar A.Navoiy 32a-uy

Azimova Zarina

Fargona shahar Terakmozor dahasi

Hamidov Bahtiyor

Andijon viloyati Jaloquduq tumani

TavarNomi

► Malibu

Dasturimda myb va unga tegishli bo'lgan maxsulot va unga tegishli sotish ma'lumolari mavjud jadvallar mavjud. Yangi myb qo'shmoqchi bo'lsam, ni aktivlashtiraman va myb jadvaliga bir marta sichqoncha bilan chertaman va tugmasini bosaman. Natijada quyidagicha xodim va uni lavozimini kiritish darchasi xosil bo'ladi.

Mijozlar haridini tahrirlash oynasi...

Aliyev Jaloliddin

Tovar turini tanlang:

Tovarni tanlang:

Tovar narxi:

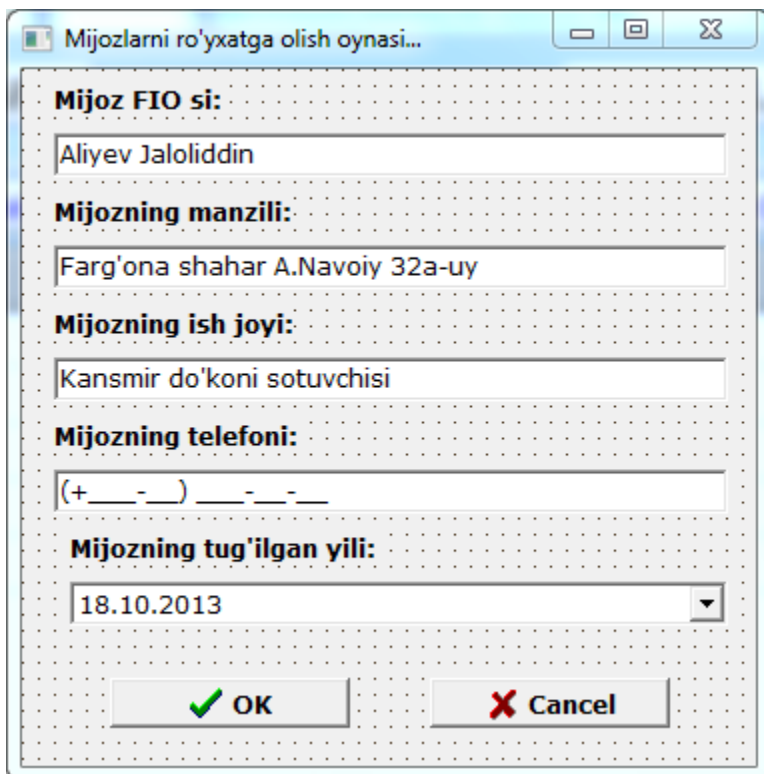
Sotib olingan sana:

19.10.2013

dbgrd1: TDBGrid
Origin: 1, 42; Size: 591 x 231
Tab Stop: True; Order: 2

Maydonlar kiritiladi va tugmasi bosiladi.

Myb tahrirlamoqchi bo'lsam  ni aktivlashtiraman, xodimlar jadvalini chertib,  tugmasini bosaman.



Mijozlarni ro'yxatga olish oynasi...

Mijoz FIO si:
Aliyev Jaloliddin

Mijozning manzili:
Farg'ona shahar A.Navoiy 32a-uy

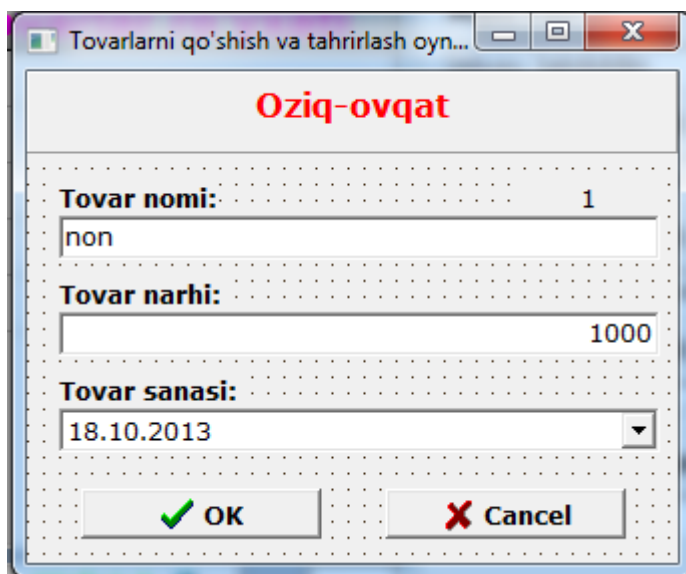
Mijozning ish joyi:
Kansmir do'koni sotuvchisi

Mijozning telefoni:
(+__-__)__-__-__

Mijozning tug'ilgan yili:
18.10.2013

Yangi mybninh yetkazib bergan maxsuloti to'grisidagi ma'lumotni qo'shmoqchi bo'lsam,  ni aktivlashtiraman va maxsulot jadvaliga bir marta sichqoncha bilan chertaman va  tugmasini bosaman. Natijada quyidagicha maxsulot kiritish darchasi xosil bo'ladi.



Tovarlarni qo'shish va tahrirlash oynasi...

Oziq-ovqat

Tovar nomi: 1
non

Tovar narhi:
1000

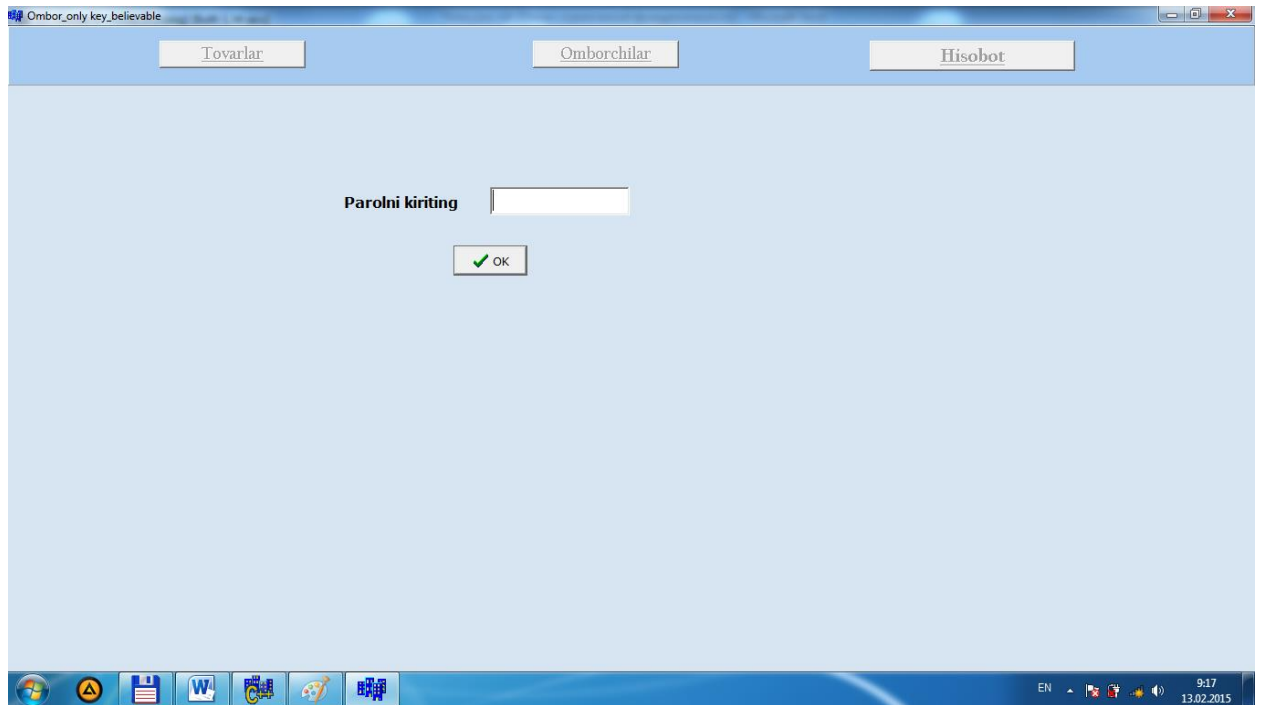
Tovar sanasi:
18.10.2013

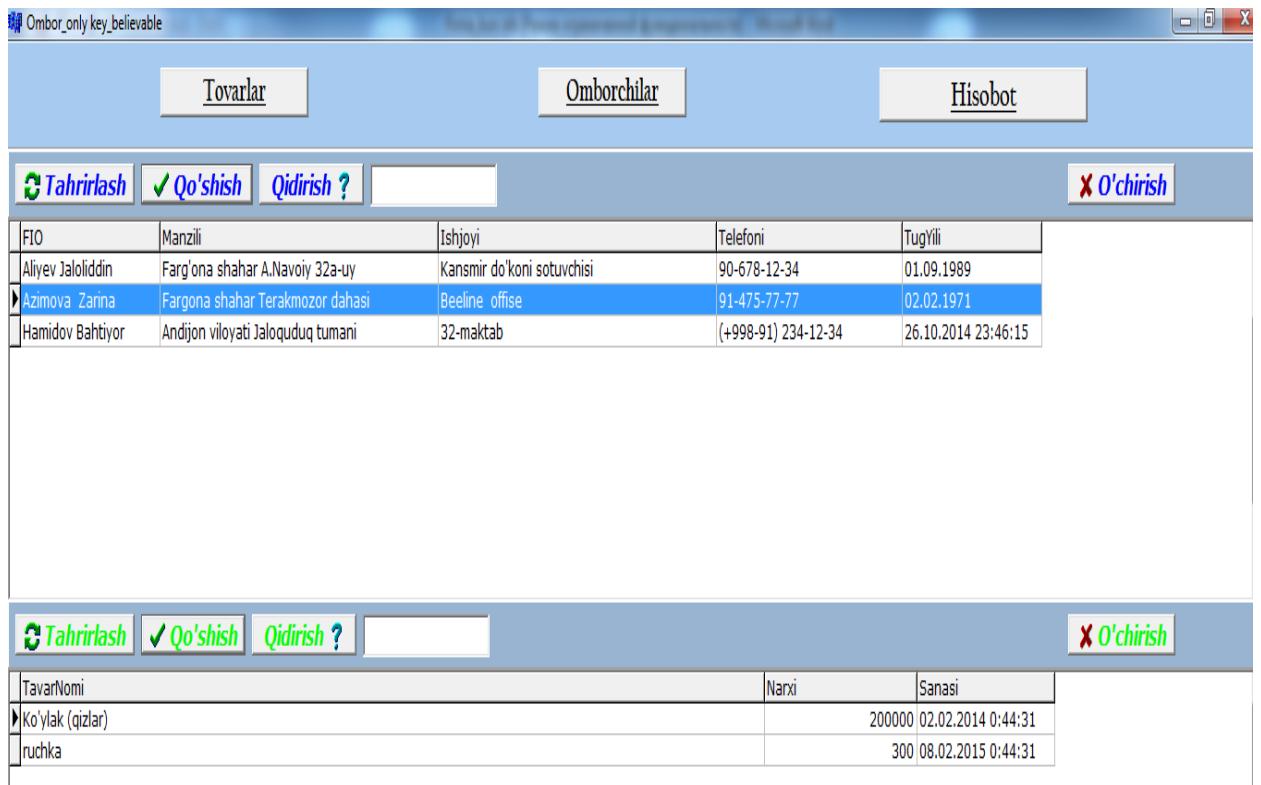
Maxsulot nomi bo'limiga maxsulotning nomi kiritiladi va maxsulotning soni va 1 donasi uchun narxi kiritiladi va OK tugmasi bosiladi.

Dastur ishlashidan lavhalar:

Asosiy oyna:

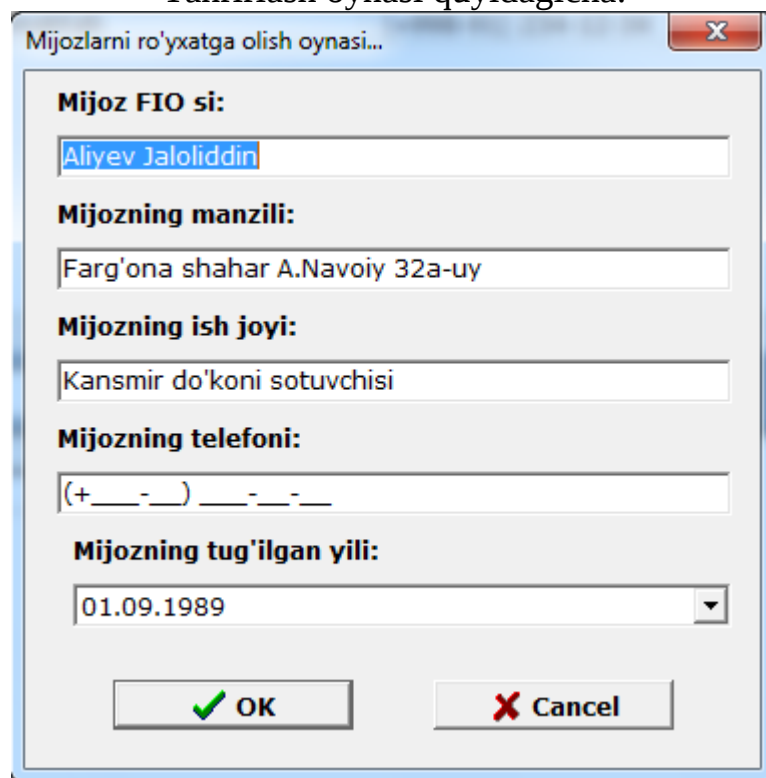


Dastur ikki bo'limdan iborat. Bular **Tovarlar** va **Omborchilar** bo'limlaridir. Foydalanuvchi bo'limni tanlab ish boshlaydi.



Mijozlar bo'limi quyidagi tuzilishda

Tahrirlash oynasi quyidagicha:



Mijozlarni ro'yxatga olish oynasi...

Mijoz FIO si:
Aliyev Jaloliddin

Mijozning manzili:
Farg'ona shahar A.Navoiy 32a-uy

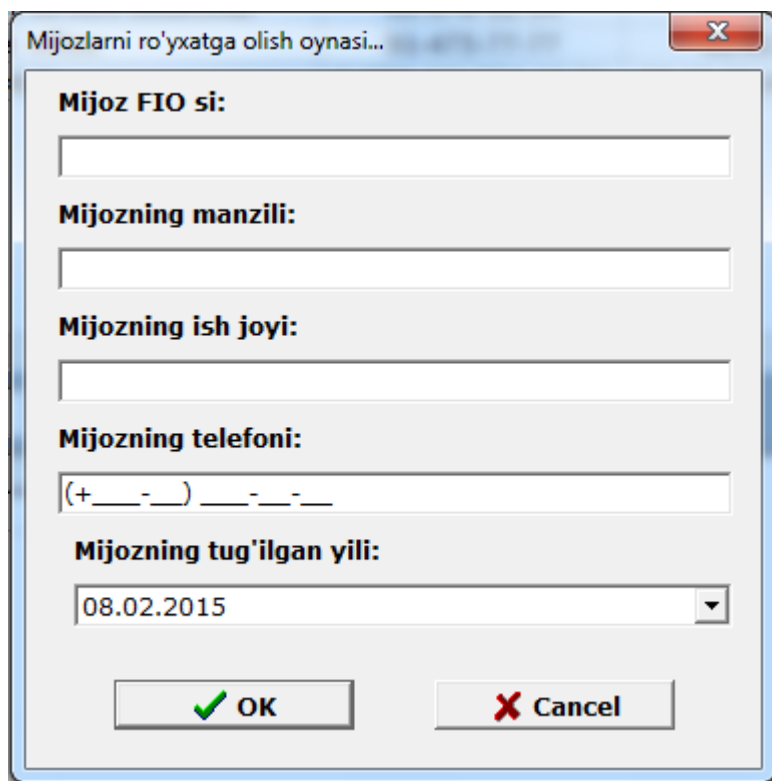
Mijozning ish joyi:
Kansmir do'koni sotuvchisi

Mijozning telefoni:
(+__-__) __-__-__

Mijozning tug'ilgan yili:
01.09.1989

OK Cancel

Mijozlarni ro'yxatga olish quyidagicha amalgam oshiriladi.



Mijozlarni ro'yxatga olish oynasi...

Mijoz FIO si:

Mijozning manzili:

Mijozning ish joyi:

Mijozning telefoni:
(+__-__) __-__-__

Mijozning tug'ilgan yili:
08.02.2015

OK Cancel

Dasturda qidiruv funksiyasi ham yaratilgan bo'lib uning ishlash prinsipi: kiritilgan Text oynasidagi text bor ismli omborchi yoki tovarni qidirib kursorni topilgan natijaga qo'yadi.

Ombor_only key_believable

Tovarlar Omborchilar Hisobot

Tahrirlash Qo'shish Qidirish ? ali O'chirish

FIO	Manzili	Ishjoyi	Telefoni	TugYili
Aliyev Jaloliddin	Farg'ona shahar A.Navoiy 32a-uy	Kansmir do'koni sotuvchisi	90-678-12-34	01.09.1989
Azimova Zarina	Fargona shahar Terakmozor dahasi	Beeline offise	91-475-77-77	02.02.1971
Hamidov Bahtiyor	Andijon viloyati Jaloquduq tumani	32-maktab	(+998-91) 234-12-34	26.10.2014 23:46:15

Tahrirlash Qo'shish Qidirish ? O'chirish

TavarNomi	Narxi	Sanasi
Malibu	100000000	27.12.2014

Dasturimizning yana bir qulaylik qismlaridan biri bu uning sotilgan tovarlar haqida hisobot yaratishidadir.

Print Preview

OMBORDAGI TOVARLARNING UMUMIY RO'YHATI

Nomi	Narxi	Qo'yilgan sanasi	Omborga qo'yg'an shaxs FIO
Malibu	1000	27.12.2014	Aliyev Jaloliddin
ruchka	1000	08.02.2015 0:44:31	Azimova Zarina
Ko'ylak (qizlar)	1000	02.02.2014 0:44:31	Azimova Zarina
Lacceti	1000	08.02.2015 0:44:31	Hamidov Bahtiyor
Kompyuter	1000	08.02.2015 0:44:31	Hamidov Bahtiyor

Xulosa.

Men ushbu kurs ishimni bajarish davomida C++ dasturlash tilida dasturlash hamda avtomatlashtirish to'g'risidagi bilimlarimni mustahkamlab oldim. Dastur tuzish davomida C++ Builder dasturida ishladim. C++ Builderning ko'plab komponentalari bilan mukammal amaliy ishlar olib bordim. Men ushbu kurs ishini bajarish davomida juda ko'p narsalarni o'rgandim.

Obyektli dastur tuzish bu ikki jarayonni birlashtirish demakdir. Dasturi tuzilayotgan soha yoki tizim va algoritmik ammalardir.

Men shuni anglab yetdimki dasturchi bu barcha sohaning yetakchisi bo'lmog'i lozim. U shundagina yetuk dasturchi bo'lib yetishadi.

Foydalanilgan adabiyotlar

1. Гради Буч. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. 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 C++ 4.5. Киев. Диалектика. 1996.
5. www.ziyonet.uz
6. www.C++.ru
7. www.dastur.uz

Ilova

Dasturdagi kodlardan misollar.

Asosiy forma kodlari:

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

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

```
#pragma hdrstop
```

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

```
#include "DMUnit.h"
```

```
#include "TovarUnit.h"
```

```
#include "TovarDetailUnit.h"
```

```
#include "MijozUnit.h"
```

```
#include "MijozDetailUnit.h"
```

```
#include "Hisobot.h"
```

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

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

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

```
TmainForm *mainForm;
```

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

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

```
    : TForm(Owner)
```

```
{
```

```
}
```

```

//-----
void __fastcall TmainForm::BitBtn1Click(TObject *Sender)
{
    DM->tblTavarlar->Edit();
    TovarForm->ShowModal();
}

//-----

void __fastcall TmainForm::btn1Click(TObject *Sender)
{
    DM->tblMijoz->Edit();
    MijozForm->dtpTYili->Date=DM->tblMijozTugYili->AsDateTime;
    MijozForm->MaskEdit1->Text=DM->tblMijozTelefoni->AsString;
    MijozForm->ShowModal();
}

//-----

void __fastcall TmainForm::btnTovarClick(TObject *Sender)
{
    pnlTovarlar->Visible=True;
    pnlMijozlar->Visible=False;
    pnlTovarlar->Align=alClient;
}

//-----

void __fastcall TmainForm::btnMijozlarClick(TObject *Sender)
{
    pnlMijozlar->Visible=True;

```

```

    pnlTovarlar->Visible=False;

    pnlMijozlar->Align=alClient;
}

//-----

void __fastcall TmainForm::BitBtn2Click(TObject *Sender)
{
    DM->tblTavarlar->Insert();
    TovarForm->ShowModal();
}

//-----

void __fastcall TmainForm::BitBtn3Click(TObject *Sender)
{
    int Res;

    Res=MessageDlg("Yozuvni
o'chirmoqchimisiz",mtWarning,TMsgDlgButtons())<<mbYes<<mbNo,0);

    if (Res==mrYes)
    {
        DM->tblTavarDetail->First();
        for(int i=0;i<DM->tblTavarDetail->RecordCount;i++)
        {
            DM->tblTavarDetail->Delete();
            DM->tblTavarDetail->Next();
        }
        DM->tblTavarlar->Delete();

        DM->tblTavarDetail->Active=False;
    }
}

```

```

    DM->tblTavarDetail->Active=True;

    DM->tblTavarlar->Active=False;

    DM->tblTavarlar->Active=True;
}

}

//-----

void __fastcall TmainForm::BitBtn6Click(TObject *Sender)
{
    int Res;

    Res=MessageDlg("Yozuvni
o'chirmoqchimisiz",mtWarning,TMsgDlgButtons()<<mbYes<<mbNo,0);

    if (Res==mrYes)
    {
        DM->tblTavarDetail->Delete();

        DM->tblTavarDetail->Active=False;

        DM->tblTavarDetail->Active=True;
    }
}

//-----

void __fastcall TmainForm::btn3Click(TObject *Sender)
{
    int Res;

    Res=MessageDlg("Yozuvni
o'chirmoqchimisiz",mtWarning,TMsgDlgButtons()<<mbYes<<mbNo,0);

    if (Res==mrYes)

```

```

{
    DM->tblMijozDetail->First();
    for(int i=0;i<DM->tblMijozDetail->RecordCount;i++)
    {
        DM->tblMijozDetail->Delete();
        DM->tblMijozDetail->Next();
    }
    DM->tblMijoz->Delete();

    DM->tblMijozDetail->Active=False;
    DM->tblMijozDetail->Active=True;
    DM->tblMijoz->Active=False;
    DM->tblMijoz->Active=True;
}
}

//-----

void __fastcall TmainForm::btn6Click(TObject *Sender)
{
    int Res;

    Res=MessageDlg("Yozuvni
o'chirmoqchimisiz",mtWarning,TMsgDlgButtons()<<mbYes<<mbNo,0);
    if (Res==mrYes)
    {
        DM->tblMijozDetail->Delete();
        DM->tblMijozDetail->Active=False;
        DM->tblMijozDetail->Active=True;
    }
}

```

```

    }

}

//-----

void __fastcall TmainForm::BitBtn5Click(TObject *Sender)
{
    DM->tblTavarDetail->Insert();

    TavarDetailForm->dtpSanasi->Date=Now();

    TavarDetailForm->ShowModal();
}

//-----

void __fastcall TmainForm::BitBtn4Click(TObject *Sender)
{
    DM->tblTavarDetail->Edit();

    TavarDetailForm->dtpSanasi->Date=DM->tblTavarDetailTavarVaqti-
>AsDateTime;

    TavarDetailForm->ShowModal();
}

//-----

void __fastcall TmainForm::btn2Click(TObject *Sender)
{
    DM->tblMijoz->Insert();

    MijozForm->dtpTYili->Date=Now();

    MijozForm->MaskEdit1->Clear();

    MijozForm->ShowModal();
}

//-----

```

```
void __fastcall TmainForm::btn5Click(TObject *Sender)
```

```
{
```

```
MijozdetailForm->cbbTuri->Text="";
```

```
MijozdetailForm->cbbTavarlar->Text="";
```

```
MijozdetailForm->Edit1->Text="";
```

```
MijozdetailForm->dtpSotilganSana->Date=Now();
```

```
DM->tblMijozDetail->Insert();
```

```
MijozdetailForm->ShowModal();
```

```
}
```

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

```
void __fastcall TmainForm::btn4Click(TObject *Sender)
```

```
{
```

```
DM->tblMijozDetail->Edit();
```

```
MijozdetailForm->cbbTavarlar->Text=DM->tblMijozDetailTavarNomi->AsString;
```

```
MijozdetailForm->Edit1->Text=DM->tblMijozDetailNarxi->AsString;
```

```
MijozdetailForm->dtpSotilganSana->Date=DM->tblMijozDetailSanasi->AsDateTime;
```

```
MijozdetailForm->ShowModal();
```

```
}
```

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

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

```
{
```

```
DM->tblMijozDetail->First();
```

```
Form2->QuickRep1->Preview();
```

```
}
```

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

```
void __fastcall TmainForm::BitBtn11Click(TObject *Sender)
```

```
{
```

```
DM->tblMijoz->Edit();
```

```
MijozForm->dtpTYili->Date=DM->tblMijozTugYili->AsDateTime;
```

```
MijozForm->MaskEdit1->Text=DM->tblMijozTelefoni->AsString;
```

```
MijozForm->ShowModal();
```

```
}
```

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

```
void __fastcall TmainForm::BitBtn12Click(TObject *Sender)
```

```
{
```

```
DM->tblMijoz->Insert();
```

```
MijozForm->dtpTYili->Date=Now();
```

```
MijozForm->MaskEdit1->Clear();
```

```
MijozForm->ShowModal();
```

```
}
```

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

```
void __fastcall TmainForm::BitBtn14Click(TObject *Sender)
```

```
{  
MijozdetailForm->cbbTuri->Text="";  
MijozdetailForm->cbbTavarlar->Text="";  
MijozdetailForm->Edit1->Text="";  
MijozdetailForm->dtpSotilganSana->Date=Now();  
DM->tblMijozDetail->Insert();  
MijozdetailForm->ShowModal();  
}  
//-----
```

```
void __fastcall TmainForm::BitBtn13Click(TObject *Sender)  
{  
DM->tblMijozDetail->Edit();  
MijozdetailForm->cbbTavarlar->Text=DM->tblMijozDetailTavarNomi->AsString;  
MijozdetailForm->Edit1->Text=DM->tblMijozDetailNarxi->AsString;  
MijozdetailForm->dtpSotilganSana->Date=DM->tblMijozDetailSanasi->AsDateTime;  
MijozdetailForm->ShowModal();  
}  
//-----
```

```
void __fastcall TmainForm::BitBtn7Click(TObject *Sender)  
{  
TLocateOptions Opts;
```

```
Opts.Clear();
```

```
Opts << loPartialKey;
```

```
DM->tblTavarlar->Locate("TavarTuri", Edit1->Text, Opts);
```

```
}
```

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

```
void __fastcall TmainForm::BitBtn8Click(TObject *Sender)
```

```
{
```

```
TLocateOptions Opts;
```

```
Opts.Clear();
```

```
Opts << loPartialKey;
```

```
DM->tblTavarDetail->Locate("TavarNomi", Edit2->Text, Opts);
```

```
}
```

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

```
void __fastcall TmainForm::BitBtn9Click(TObject *Sender)
```

```
{
```

```
TLocateOptions Opts;
```

```
Opts.Clear();
```

```
Opts << loPartialKey;
```

```
DM->tblMijoz->Locate("FIO", Edit3->Text, Opts);
```

```
}
```

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

```
void __fastcall TmainForm::BitBtn10Click(TObject *Sender)
```

```
{
```

```
TLocateOptions Opts;
```

```
Opts.Clear();
```

```
Opts << loPartialKey;
```

```
DM->tblMijozDetail->Locate("TavarNomi", Edit4->Text, Opts);
```

```
}
```

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

```
void __fastcall TmainForm::BitBtn15Click(TObject *Sender)
```

```
{
```

```
if (Edit5->Text=="851") {
```

```
btnMijozlar->Enabled=true;
```

```
btnTovar ->Enabled=true;
```

```
Button1->Enabled=true;
```

```
Label2->Visible=true;
```

```
Label3->Visible=true;
```

```
Label1->Visible=false;
```

```
Edit5->Visible=false;
```

```
BitBtn15->Visible=false;
```

```
}
```

else

{

ShowMessage("Parol hat");

Edit5 ->SetFocus();

}

}

//-----

void __fastcall TmainForm::FormActivate(TObject *Sender)

{

btnMijozlar->Enabled=false;

btnTovar->Enabled=false;

Button1->Enabled=false;

}

//-----

Mijozlar bilan ishlash formalaridagi dastur kodlari.

void __fastcall TMijozdetailForm::FormActivate(TObject *Sender)

{

cbbTuri->Items->Clear();

lstId->Items->Clear();

DM->tblTavarlar->First();

for (int i=0;i<DM->tblTavarlar->RecordCount;i++)

{

cbbTuri->Items->Add(DM->tblTavarlarTavarTuri->AsString);

lstId->Items->Add(DM->tblTavarlarId->AsString);

DM->tblTavarlar->Next();

```

    }
}
//-----
void __fastcall TMijozdetailForm::cbbTuriClick(TObject *Sender)
{
    int i=cbbTuri->ItemIndex;
    String Id=lstId->Items->Strings[i];

    DM->ADOQuery1->Active=False;
    DM->ADOQuery1->SQL->Clear();
    DM->ADOQuery1->SQL->Add("Select * From TavarDetail Where
TavarId="+Id);
    DM->ADOQuery1->ExecSQL();
    DM->ADOQuery1->Active=True;

    cbbTavarlar->Items->Clear();
    lstNarxi->Items->Clear();
    DM->ADOQuery1->First();
    for (int j=0;j<DM->ADOQuery1->RecordCount;j++)
    {
        lstNarxi->Items->Add(DM->ADOQuery1->FieldByName("TavatNarxi")-
>AsString);
        cbbTavarlar->Items->Add(DM->ADOQuery1->FieldByName("TavarNomi")-
>AsString);
        DM->ADOQuery1->Next();
    }
}
//-----
void __fastcall TMijozdetailForm::cbbTavarlarClick(TObject *Sender)

```

```

{
    int i=cbbTavarlar->ItemIndex;
    String S=lstNarxi->Items->Strings[i];
    Edit1->Text=S;
}

//-----

void __fastcall TMijozdetailForm::BitBtn1Click(TObject *Sender)
{
    DM->tblMijozDetail->FieldByName("MijozId")->AsString=dbtxtId->Caption;
    DM->tblMijozDetail->FieldByName("TavarNomi")->AsString=cbbTavarlar->Text;

    DM->tblMijozDetail->FieldByName("Narxi")->AsCurrency=Edit1->Text;
    DM->tblMijozDetail->FieldByName("Sanasi")->AsDateTime=dtgSotilganSana->Date;

    DM->tblMijozDetail->Post();
    DM->tblMijozDetail->Active=False;
    DM->tblMijozDetail->Active=True;
}

//-----

void __fastcall TMijozdetailForm::BitBtn2Click(TObject *Sender)
{
    DM->tblMijozDetail->Active=False;
    DM->tblMijozDetail->Active=True;
}

//-----

```