



**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
MADAMINOVA S**

Qabul qildi:

MUSAYEV X

Farg'ona 2015

Reja:

Kirish

I. Nazariy qism

1.1 C++ tarixi

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

1.3 C++ Builder asoslari

II. Amaliy qism

2.1. Ma'lumotlar bazasini loyihalashtirish

2.2. Kamolot nomli supermarket uchun dasturiy taminot ishlab chiqish.

2.3. Dastur imkoniyatlari

III. Xulosa

IV. Foydalanilgan adabiyotlar

V. Ilova.

KIRISH

Hozirgi kunda ahborot texnologiyalari jamiyatimizning barcha sohalariga kirib bormoqda. Jumladan yosh avlodga ta'lim berish sohasida ham qator yangiliklar, o'zgarishlar amalga oshirilmoqda. Bu kurs ishida boshlang'ich sinf o'quvchilarining matematikadan bilimlarini sinashga mo'ljallangan test dasturi yaratildi. Dastur asosan arifmetik amallarga asoslangan holda tuzilgan. Bizga ma'lumki, boshlang'ich sinf o'quvchilarida turli xil musobaqalarga, o'yinlarga qiziqish yuqori bo'ladi. Yaratilgan dastur yordamida o'quvchilar o'z bilimlarini sinash bilan birga o'z bilimlarini tekshirish imkoniyatiga ham ega bo'ladilar.

Zamonaviy kompyuterlarda turli dasturlash tillari keng qo'llanilmoqda. Bu dasturlar iqtisodiyot, boshqarish, xizmat ko'rsatish va ishlab chiqarishning turli sohalarida muhim ahamiyatga molik masalalarni hal qilishda ba'zan yagona omil bo'lib qolmoqda. Zamonaviy ishlab chiqarish korxonalarida ishlab chiqarish jarayonini takomillashtirish va avtomatlashtirishda dasturlashning ahamiyati katta. Sanoatning turli sohalarida ishlatilayotgan avtomatlashtirilgan jihozlar, robot texnikalari ham turli algoritmik tillarda yozilgan ma'lum dasturlar asosida kompyuterlar yordamida boshqariladi. Bu esa o'z navbatida, ishlab chiqarish korxonalarida xizmat qilayotgan, nafaqat boshqaruv xodimlari va muhandislariga, balki ishchilarga ham dasturlash tillarini puxta egallashlarini taqozo etadi. Obyektga yo'naltirilgan dasturlash dasturlashni endi boshlaganlar uchun qiyinroq mavzu bo'lishi mumkin, lekin uni amaliyotda qo'llash bir qator qulayliklarni yaratishi ko'plarni bu mavzuga jalb etib kelmoqda. Obyektga yo'naltirilgan dasturlash (Object-oriented programming, OOP) real dunyoni sodda shaklda tasvirlash imkonini beradi. Tushinib olish uchun quyidagi masalani ko'ramiz. Deylikki binolarni qurishda g'isht teruvchilar, suv quvurlari quruvchilari, elektriklar va boshqa kasb egalari mehnat qilishadi. Bunda g'isht teruvchilar faqatgina g'isht terishni bilishsa bo'lgani, ular elektr simlari qayerdan o'tadiyu, suv quvurlari qayerdan o'tishini bilishlari shart emas. Ular o'z ishlarini yaxshi bajarishsa bo'lgani. Bosh arxitektor qurilishdagi barcha ishlarni mutannosib kelishini kafolatlab qo'yadi. OYDT da ham dasturning har bir komponenti o'z

vazifasini bajarsa umumiy loyiha ko'ngildagidek ishlashi kafolatlanishi mumkin. Biz dasturlashda OYDTdan foydalanmasdan protsedurali dastur tuzishimiz mumkin. Agar biz OYDTdan foydalansak ham yoki protsedurali dastur tuzsak ham dastur tuzib bo'lgandan so'ng quyidagi mumamolardan biriga yoki bir nechtasiga duch kelishimiz mumkin:

- Loyihada qandaydir o'zgarish bo'lishi
- Dastur tuzishda qandaydir xotolik yuzaga kelishi va boshqa kamchiliklar

Agar loyiha dasturi 100000 satrdan ko'p bo'ladigan bo'lsa xatolarni izlab topish va tuzatish uchun ketgan vaqt qariyb yangi dastur tuzish uchun ketgan vaqtga teng bo'lib qoladi. Bunday mumamaga yana duch kelsak unda nima bo'ladi?

Agarda OPP dan foydalansak juda qisqa vaqtda loyihaga o'zgartirish kirishish, xatolarni izlab topib tuzatish imkoni bo'ladi.

C++ Builder dasturi SDI ilova dasturi hisoblanib, asosan sozlovchi asboblar paneli (chap tomonda) va komponentlar palitrasini (o'ng tomonda) o'z ichiga oladi. Bundan tashqari C++ Builder yuklanayotganda ob'yektlar inspektori (chap tomonda) hamda yangi ilova dasturi shakli (o'ng tomonda) paydo bo'ladi. Yangi ilova dasturi shakli orqasida kodni tahrirlovchi oyna ham mavjud bo'ladi.

Shakllar C++ builder ning asosi hisoblanadi. Foydalanuvchi ilova dasturi interfeysini yaratish uchun oynaga C++ Builder ob'yekti formasi elementlarini, ya'ni komponentlarni qo'yish bilan amalga oshiriladi. C++ Builder komponentlari o'ng tomondagi komponentlar palitrasida joylashgan bo'ladi. C++ Builderning eng yaxshi tomonlaridan biri shuki, unda dasturchi o'zi hohlagan yangi komponent yaratishi va uni komponentlar palitrasiga sozlab olishi mumkin. Bundan tashqari turli loyihalar uchun yangi komponentlar yaratish mumkin.

I. NAZARIY QISM

1.1. C++ tili dasturlash tiil haqida tushuncha

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'ektlil 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++ dasturlash tili C dasturlash tiliga

asoslangan. C dasturlash tili 1972 yili ilk bor Bell Laboratoriesda, DEC PDP-11 klmpyuterida qo'llangan. C dasturlash tili o'zidan oldingi tillarning juda ko'p muhim tomonlarini o'z ichiga olish bilan bir qatorda, o'zgaruvchilarni tiplashtirish va yana bir qator yangiliklar qo'shib oldi. Boshlanishda S dasturlash tili asosan UNIX sistemalarida keng tarqaldi. Hozirda operatsion tizimlarning asosiy qismi C/C++ dasturlash tillarida yozilmoqda. 1983 yili C dasturlash tili keng tarqalganligi sababli, uni standartlash harakati boshlandi. Buning uchun Amerika Milliy Standartlar Komiteti (ANSI) qoshida X3J11 texnik komitet tuzildi. Harakatlar natijasida 1989 yil ushbu standart qabul qilindi. Standartni dunyo bo'yicha keng tarqatish maqsadida 1990 yilda ANSI va Dunyo Standartlar Tashkiloti (ISO) hamkorlikda C dasturlash tilining ANSI/ISO 9899:1990 standarti qabul qilindi. C++ dasturlash tili 1980 yillar boshida Bjarne Stroustrup tomonidan C dasturlash tiliga asoslangan holda yaratildi. C++ dasturlash tili juda ko'p qo'shimchalarni o'z ichiga olgan dasturlash tilidir. Bu qo'shimchalarning eng asosiysi ob'ektlar bilan ishlash olish imkoniyatidir. Dasturlarni tez va sifatli yozish hozirgi kunda katta ahamiyat kasb etmoqda. Buni ta'minlash uchun ob'ektli dasturlash g'oyasi ilgari surildi. Huddi 70-chi yillar boshida strukturali dasturlash kabi, dasturlarni hayotdagi jismlarni modellashtiruvchi ob'ektlar orqali tuzish, dasturlash sohasida inqilob qildi. C++ dasturlash tilidan tashqari boshqa ko'p ob'ektga yo'naltirilgan dasturlash tillari paydo bo'ldi. Shulardan eng ko'zga tashlanadigani Xeroxning Palo Altoda joylashgan ilmiy – qidiruv markazida (PARC) tuzilgan Smalltalk dasturlash tilidir. Smalltalk dasturlash tilida hamma narsa ob'ektlarga asoslangan. C++ dasturlash tili esa gibrid tildir. Unda C dasturlash tiliga o'xshab strukturali dasturlash yoki yangicha, ob'ektlar bilan dasturlash mumkin. Yangicha deyishimiz ham nisbiydir. Ob'ektli dasturlash falsafasi paydo bo'lganiga ham qariyb o'ttiz yil bo'ldi. C++ dasturlash tili funksiya va ob'ektlarning juda boy kutubxonalariga ega. Ya'ni C++ dasturlash tilida dasturlashni o'rganish ikki qismga bo'linadi. Birinchisi bu C++ dasturlash tilini o'zini o'rganish, ikkinchisi esa C++ dasturlash tilining standart kutubxonasidagi tayyor ob'ekt va funksiyalarini qo'llashni o'rganishdir.

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)

1.2. Borland C++ obyektli dasturlash tili muhitida ishlash.

Dastur yaratish umumlashgan muhiti Redaktor form – Shakllar muharriri, Inspektor ob'ektov – Ob'ektlar inspektori, Palitra komponentov – Komponentlar palitrasi, Administrator proekta – Proekt administratori va to'la umumlashgan Redaktor koda – Kodlar muharriri hamda kodlar va resurslar ustidan to'liq nazoratni ta'minlaydigan , dastur ilovalarini tezkor yaratadigan Otladchik - instrumentov - Sozlash-instrumentlari kabilarni birlashtiradi.

Komponentlar

Komponentalarni shaklga o'rnatish uchun komponentlar palitrasidagi kerakli piktogramma tanlanadi, so'ngra shaklning komponenta joylanishi kerak bo'lgan joyi tanlanadi. Shundan so'ng komponentalar xossalarini ob'ektlar inspektori yordamida tahrirlash mumkin. Properties bandida komponentalar xossalarining ro'yxati (chapda) va bu xossalarning qiymatlar ro'yxati (o'ngda) joylashgan.

Komponentalar ko'rinadigan (vizual) va ko'rinmaydigan (vizual bo'lmagan) larga bo'linadi. Vizual komponentalar bajarilish paytida proektlash paytidagidek paydo bo'ladi. Bunga tugmalar va tahrirlanuvchi maydonlar misol bo'la oladi. Vizual bo'lmagan komponentalar proekt ishlagan vaqtida shakldagi piktogramma ko'rinishida paydo bo'ladi. Ular bajarilish paytida hech qachon ko'rinmaydi, ammo ma'lum funkcionallikga ega bo'ladi (masalan, berilganlarga murojatni ta'minlaydi, Windowsning standart mulokatlarini chaqiradi).

Xossalar

Xossalar komponentalarning tashqi ko'rinishi va tabiatini aniqlovchi atributlar hisoblanadi. Xossalar ustunidagi ko'p xossalar komponentalari oldindan o'rnatilgan (по умолчанию) qiymatlarga ega bo'ladi (masalan, tugmalarlar balandligi). Komponentalar xossalari xossalar varag'i (Properties) da aks ettiriladi. Ob'ektlar inspektori komponentalarning nashr etilgan (published) xossalarini aks ettiriladi. Published-xossalardan tashqari komponentalar umumiy (public), faqat ilovalarning bajarilish paytidagina murojat qilish mumkin bo'lgan nashr qilingan xossalarga ega bo'ladi. Xossalar ro'yxati ob'ektlar inspektori xossalar varag'ida joylahadi. Xossalarni proektlash paytida aniqlash mumkin yoki ilovalarning bajarilish paytida ko'rinishini o'zgartirish uchun kod yozish mumkin. Komponenta xossalarini proektlash paytida aniqlash uchun shakldagi komponenta tanlanadi, ob'ektlar inspektori xossalari varag'i ochiladi, aniqlanadigan xossa tanlanadi va zarur bo'lsa xossalar muharriri yordamida o'zgartiriladi (bu kiritish uchun oddiy maydon yoki son, osilib tushuvchi ro'yxat, ochiluvchi ro'yxat, muloqat paneli va boshqalar bo'lishi mumkin).

Biror komponentaning xossalarini dasturning bajarilish paytida o'zgartirish uchun «Имя Компонента» → «Название свойства» tavsifiga o'zgaruvchidek murojat qilish kerak, ya'ni qiymatlarni o'zimiz hohlagandek o'qishimiz yoki almashtirishimiz mumkin.

Xodisalar

Ob'ektlar inspektorining xodisalar varag'i (Events) komponentalar tomonidan taniladigan xodisalar ro'yxatini ko'rsatadi. Har bir komponenta o'zining shaxsiy xodisalarni qayta ishlovchi naborga ega bo'ladi. C++ Builder da xodisalarni qayta ishlovchi funksiyalarni yozish va xodisalarni bu funksiya bilan bog'lashga to'g'ri keladi. Biror bir xodisaga qayta ishlovchi yozib, siz dasturga bu xodisa ro'y berganda yozilgan funksiyaning bajarilishini topshirasiz.

Xodisani qayta ishlovchini qo'shish uchun shaklda xodisani qayta ishlovchi komponenta tanlanadi. So'ngra xodisalar varag'ida ob'ektlar inspektori ochilib (Event bandi) xodisaning qatoridagi qiymatlar ustunida sichqonning chap tugmasi ikki marta bosiladi. Bu bilan C++ Builder ni xodisalarni qayta ishlash prototipini

generatsiya qilishga va uni kodlar muharririda ko'rinishiga majbur qiladi. Bu holda bo'sh funksiya nomi generatsiya qilinadi va muharrir kod kiritilishi zarur bo'lgan joyda ochiladi. Kursor buyruqlar qavslari ichiga joylashadi { ... }. So'ngra xodisa sodir bo'lganda bajarilishi kerak bo'lgan kod kiritiladi. Xodisalarni qayta ishlovchi funksiya nomidan keyin ko'rsatiladigan parametrlarga ega bo'lishi mumkin.

Quyida xodisalarni qayta ishlovchi protseduraning shunday bo'sh karkasi ko'rsatilgan:

```
void __fastcall TForm1::Button2Click(TObject *Sender)  
{  
  
}
```

1.3 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, ixtisos-lashgan 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 ishlashjarayonida nafaqat tayyor komponentalardan foydalanish, balki yangikomponentalarni yaratish qobiliyatida ham namoyon bo'ladi. Yangikomponentalar, 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 menyusini unga sichqonchani o'ng tugmasini bosib ochish mumkin.

Standart komponentalar

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 shakl yoki birona boshqa komponenta uchun maxsus menu yaratadi. E'tiborga oling, aynan shu maqsad uchunhar 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'rnating. Voqea qayta ishlatgichi OnPopup yordamida bevosita maxsus menuning paydo bo'lishi oldidan bajariladigan protsedurani aniqlash mumkin.

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'rnating. Kalta soha ichida matnning hammasini ko'rish imkoniga ega bo'lish uchun WordWrap xususiyatining true qiymatini bering. Transparent xususiyatining true qiymatini lo'rnatsangiz, boshqa komponentaning bir qismini to'g'ri uning ustidajoylashtirilgan nom orasidan ko'rinib turadigan qilishingiz mumkin.

TEdit axborot yakka satrining tahrir qilinayotgan kiritishidagi to'rtburchak sohani shaklda aks ettiradi. Tahrir sohasining ichidagi boshlang'ich narsalarni Text xususiyatining qiymati bo'lgan satr aniqlaydi. TEdit komponentasi TCustomEdit sinfining to'g'ridan-to'g'ri hosilasi bo'lib, uning barcha xususiyatlari, metodlari va voqealariga vorislik qiladi.

TMemo axborot ko'plab satrining tahrir qilinayotgan kiritishidagito'rtburchak sohani shaklda aks ettiradi. Tahrir sohasining ichidagiboshlang'ich narsalarni Lines xususiyatining qiymati bo'lgan satrlarmassivi aniqlaydi. Ushbu xususiyat qiymati ustunida tugmachanibossangiz, ro'yxat elementlari muharririning darchasi ochiladi. TMemo komponentasi TCustomMemo sinfining to'g'ridan-to'g'rihosilasi bo'lib, uning barcha xususiyatlari, metodlari va voqealarigavorislik qiladi.

TButton yozuvli to'rtburchak tugmani yaratadi. Tugmacha bosil-ganda, dasturda biror bir xatti-harakat nomlanadi (initsiallashtiriladi). Tugmachalar ko'proq dialogli darchalarda qo'llanadi. Defaultxususiyatining true qiymati tomonidan tanlab olingan yashirin tugmacha, dialog darchasida har gal Enter klavishasi bosilganda, OnClick voqea qayta ishlatgichini ishga tushiradi. Cancel xususiyatining true qiymati tanlab olgan uzish tugmachasi, dialog darchasida har gal Escape klavishasi bosilganda, OnClick voqea qayta ishlatgichini ishga tushiradi. TVutton komponentasi TButtonControl sinfining hosilasi hisoblanadi.

TCheckBox ikkita holatga hamda tavsifiy matnga ega bo'lgan kvadrat chekboksni yaratadi (bunda tavsifiy matn chekboksning vazifasini spetsifikatsiya qiladi). Boksholatini 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 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 tugmatanlab 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.

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 bajarilishvaqtida tanlab olinadigan element raqami esa ItemIndex xususiyatiningichida 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 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 elementlariItems xususiyatining dasturning bajarilish paytida tanlab olinishi mumkin bo'lgan element raqami ItemIndex xususiyatining, tanlabolingan 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 olinganinibilish imkonini beradi.

TScrollBar darcha, shakl yoki boshqa komponenta ichidagilariniko'rib chiqish uchun, masalan, biror bir parametr qiymatini berilgan interval ichida harakatlanishi uchun yugurgichli aylantirish lineykasiniyaratadi. 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 (lineykaoxiridagi) ustida bosganda yoki pozitsiyalash tugmachalarini bosganda, yugurgich qanchaga surilishi kerakligini SmallChange xususiyatiningqiymati 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.

TGroupBox to'g'ri burchakli ramka ko'rinishidagi konteyner bo'lib, u qandaydir bir interfeys elementlarining mantiqan bog'langanuruhini shaklda vizual birlashtiradi. Bu komponenta Windowsningbir 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 tahriribajarilishi 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, sizradiotugmalarni bir necha ustunga guruhlashingiz mumkin.

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 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 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 usullarningbiri yordamida faollashtirishingiz mumkin: ActivPage xususiyatiningtushib 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.

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 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 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 tasvirlartiniq 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 konteyneryaratadi. Sections xususiyatida sanab o'tilgan sarlavhalarni axborotmaydonlari ustida, masalan, TListBox komponentalari ro'yxatlari

ustidajoylashtirish mumkin. Sarlavhali seksiyalar muharririning darchasi ushbu xususiyat qiymatlarining grafasida tugmacha bilan ochiladi.

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 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 shakllantiriladi. Panellar Muharririning darchasi ushbu xususiyat qiymatlarining grafasida tugmacha bilan ochiladi. SimplePanel xususiyatidan satr holatining aks etish turini (bir yoki ko'p panelli) qayta ulash uchun foydalaniladi.

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 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'yaliq 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 bit obrazining tasviri tushirilgan tugmachani yara-tadi. 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 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 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 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 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 klavishi bo'ylab o'tish) o'zgartirish imkonini beradi. Setkadagi ajratish chiziqlarining eng GridLineWidth xususiyati tomonidan belgilanadi, aylantirish chiziq-chalari esa ScrollBars xususiyati tomonidan qo'shiladi. FixedCols 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 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'rnatib. Kichikroq o'lchamdagi dastlabki tasvir butun konteynerga cho'zilib ketishi uchun Stretch xususiyatining true qiymatini o'rnatib. Tasvirlar fayllarining dinamik yuklanishi va saqlanishi uchun Picture obyekt xususiyatining LoadFromFile va SaveToFile metodlaridan quyidagi turlar yordamida foydalanib: Image->Picture->LoadFromFile(«<fayl nomi>»); Image->Picture->SaveToFile(«<fayl nomi>»);

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 xuddi iskana bilan o'yilgandek hajmli ko'rinadigan chiziqlar, bokslar yoki ramkalarni yaratadi. Komponenta chizayotgan obyektni 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'zgarmas qoldirish uchun Align xususiyatining true qiymatini o'rnatib.

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.

II Amaliy qism.

2.1 Ma'lumotlar bazasini loyihalashtirish

Ma'lumotlar bazasi (date base) - bu EHM ning tashqi xotirasida saqlanadigan, har qanday jismoniy, ijtimoiy, statistik, tashkiliy va boshqa ob'ektlar, jarayonlar, holatlarning o'zaro bog'liq va tartiblashtirilgan majmuidir.

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

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

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

Belgi - identifikatorlarning bosh vazifasi - ob'yektni guruhdagi turdosh ob'ektlardan farqlash. Ob'yektning identifikatori, umumiy aytganda, ob'yektning xususiyati haqida yoki, ya'ni shunga o'xshash, uning u yoki bu sinfga tegishligi haqida hech qanday axborot bermasligi mumkin.

Ma'lumotlar. Belgili shaklda ifodalanagan ob'yekt yoki ob'ektlarning muomalasi haqidagi axborotlar ma'lumotni tashkil qiladi.

Bu ma'lumotlar inson yoki qandaydir texnik qurilma tomonidan qabul qilinishi va tegishli tarzda interpretatsiya qilinishi mumkin. Ma'lumotlarning tavsifli xususiyati shu hisoblanadiki, ya'ni ularni bir belgili tizimdan boshqasiga qayta kodlash axborotni yuqotmagan holda o'tkazish mumkin. Belgili ifodalanishning bunaqangi xususiyatining ahamiyati - mavjud predmetli holatning

qabul qiluvchiga yoʻnaltirilgan, belgilarning har xil tizimlarida ifodalanish imkonini beradi.

Maʼlumotlar bazasini qurishda insonga yoʻnaltirilgan mantiqiy ifodalanish haqida va uzoq muddatli xotira qurilmasiga yoʻnaltirilgan jismoniy ifodalanish haqida gapirish anʼanaga aylangan.

MAʼLUMOTLAR BAZASI - maʼlum masalaga yoki biror faoliyatga taallukli oʻzaro bogʻlangan va alohida kurinishda tashkil etilgan maʼlumotlar.

MB misollar: – Xar xil katalog va spravochniklarda saklanadigan axborotlar; – Poezdlar jadvali; – Darslar jadvali; – Slovarlar;

MB maʼlumotlarni xsusiyatlari: – tuliklik; – aktuallik; – kulaylik; – Tugri tashkil etish.

Maʼlumotlar banki - bir nechta MB, texnik vosita va dastur taʼminotidan iborat avtomatlashtirilgan tizim.

MB tashkil qilish, ularga qoʻshimcha maʼlumotlarni kiritish va mavjud MBdan foydalanish uchun maxsus MBlar bilan ishlaydigan programmalar zarur boʻladi. Bunday programmalar majmui maʼlumotlar bazasini boshqarish sistemalari (MBBT) deb yuritiladi. Aniqroq qilib aytganda, MBBT–bu koʻplab foydalanuvchilar tomonidan MBni yaratish, unga qoʻshimcha maʼlumotlarni kiritish va MBni birgalikda ishlatish uchun zarur boʻlgan programmalar majmuidir. MBBTning asosiy tarkibiy qismi–maʼlumotlar boʻlsa, boshqa tarkibiy qismi–foydalanuvchilardir. Bularan tashqari Hardware- texnik va Software-dasturiy taʼminoti ham MBBTning samarali ishlashini taʼminlovchi tarkibiy qismlar hisoblanadi. Hardware tashqi qoʻshimcha qurilmadan iborat boʻlsa, programma qismi esa MB bilan foydalanuvchi oʻrtasidagi muloqotni tashkil qilishni amalga oshiradi. MBning tuzilishi oʻrganilayotgan obʼektning maʼlumotlari koʻrinishi, maʼnosi, tuzilishi va hajmiga bogʻliq boʻladi.

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

Relyatsion modellarda esa ob'ektlar va ularning o'zaro aloqalari ikki o'lchovli jadval ko'rinishida tasvirlanadi. Ma'lumotlarning bunday ko'rinishda tasvirlanishi ob'ektlarning o'zaro aloqalarini yaqqol tasvirlanishiga asos bo'ldi. MBBT alohida olingan quyidagi modullardan tashkil topadi:

MBni boshqarish bloki-disklardagi ma'lumotlar bilan foydalanuvchi programmasi va sistemaning so'rovi (query) orasidagi interfeysni aniqlaydi; Fayl menedjeri-ma'lumotlar tuzilmasi bilan disklar o'rtasidagi bog'lanishni boshqaradi; Query protsessor-ingliz tilida yozilgan query gaplarini MBni boshqarish bloki tushunadigan tilga o'tkazadi;

MB qo'yiladigan talablar qo'yilgan:

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

New Database

General

Database Name:

Character set:

Collation:

OK Cancel

2.1 rasm. Malumot bazasining ko'rinishi

Untitled @jaxongir (kurs_ishi_) - Table *

File Edit Window Help

New Save Save As Add Field Insert Field Delete Field Primary Key Move Up Move Down

Fields Indexes Foreign Keys Triggers Options Comment SQL Preview

Name	Type	Length	Decimals	Allow Null
soni	int			☒
narxi	int			☒
summa	int			☒
* vaqti	date			☒

Default:

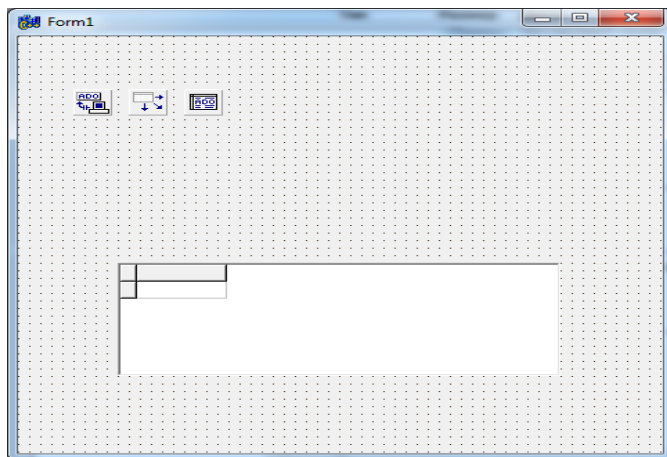
Comment:

Number of Field: 3

2.2 rasm Natijalarni yozish uchun

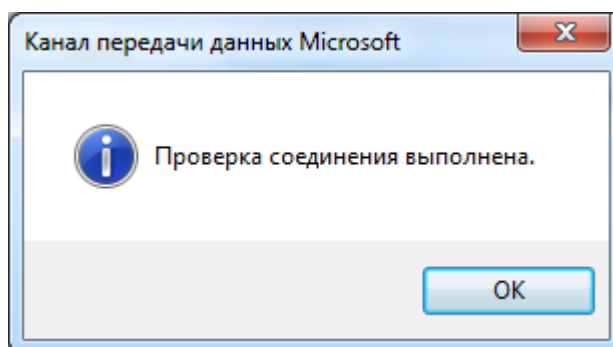
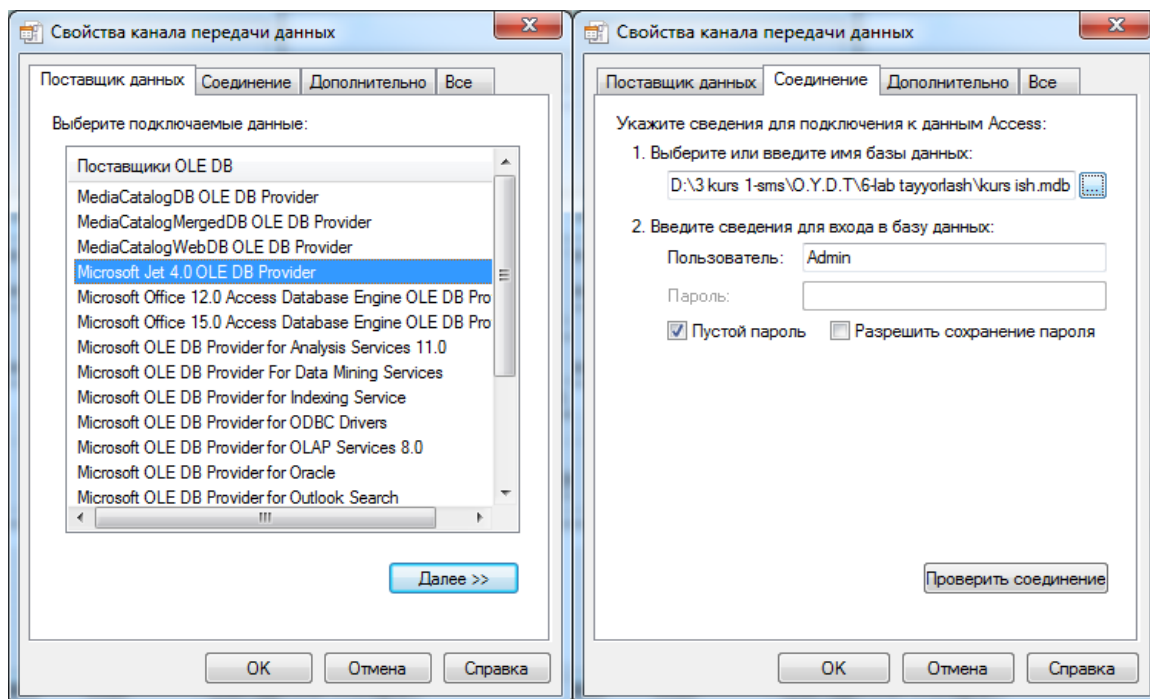
Ma'lumotlar bazasida 1 ta asosiy jadval mavjud bo'lib, ulardan birinchi savol nomli jadval dasturdagi savollarni saqlash uchun kerak. Jadvalda qiyinlik deb nomlanganu ustun mavjud bo'lib, u savollarning qiyinlik darajasini belgilaydi

Ma'lumotlar bazasini to'liq tuzib olganimizdan so'ng dasturni yaratishni boshlaymiz. Dasturni yaratishda Embarcadero 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, qidirishni amalga oshirish uchun ADOQuery, ma'lumotlarni dasturga bo'g'lash uchun DataSource komponentasidan foydalanamiz



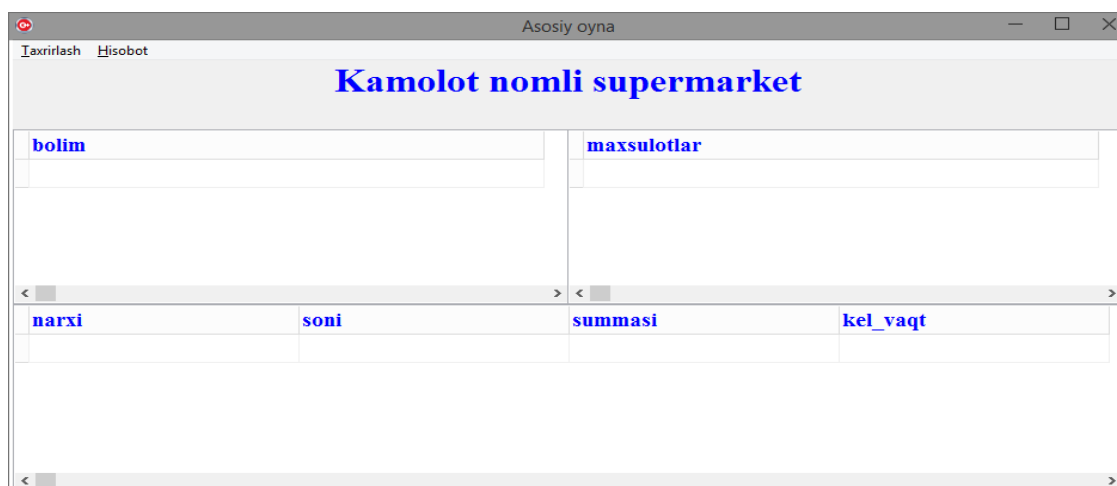
2.3-rasm

ADOconnectiondan accesdagi bazani bog'lab olamiz



2.4-rasm

ADOTable1 active hususiyatini true ga ozgartirib DBGrid1 ga jadvalni chiqaramiz

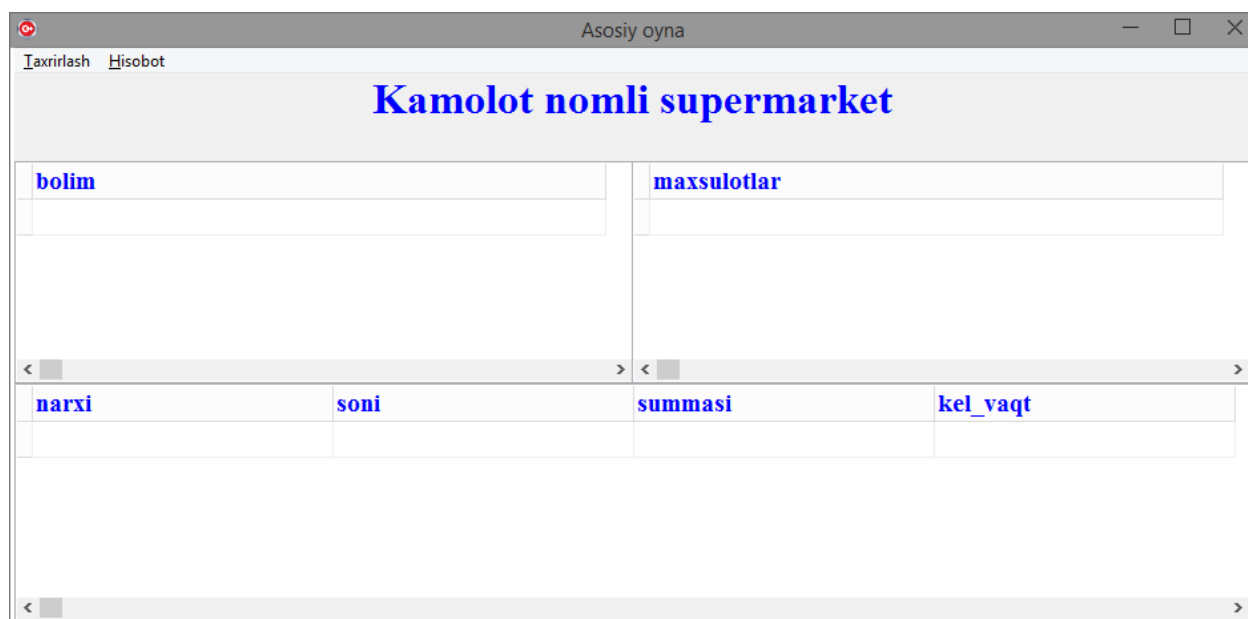


2.5 rasm

Jadvalni o'zgartirish qo'shish o'chirish uchun tugmalar joylashtiramiz

2.2 Test ishlash jarayonini hosil qilish

Dasturning eng asosiy vazifasi komlot nomli supermarketning kirim chiqim xisobotini hisob kitob qiluvchi dasturdan iborat



2.7 rasm. Test tekshirish oynasi

Bu oynada Label, RadioGroup, BitBtn komponentlaridan foydalaniladi. Asosiy masala bazadagi savollarni tartibini o'zgartirgan holda tanlab olish kerak. Buning uchun quyidagi funksiya yaratildi:

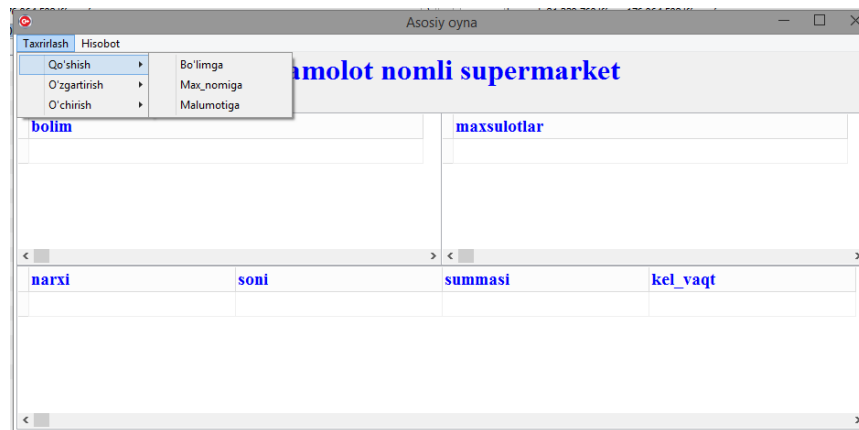
```
class TForm1 : public TForm
{
__published:      // IDE-managed Components
    TEdit *Edit1;
    TBitBtn *BitBtn1;
    TLabel *Label1;
    TBitBtn *BitBtn2;
    TLabel *Label2;
    TEdit *Edit2;
    void __fastcall BitBtn1Click(TObject *Sender);
    void __fastcall BitBtn2Click(TObject *Sender);
private:          // User declarations
public:           // User declarations
```

```
__fastcall TForm1(TComponent* Owner);
```

```
};
```

Bu funksiya ma'lumot bazasidagi savollar orasidan ko'rsatilgan qiyinlikdagi savollarni bazadan tanlab oladi hamda formaga bu savolni chiqarib beradi.

Dasturning asosiy oynasi quyidagi ko'rinishga ega



2.8 rasm. Asosiy oyna

Asosiy oynadan turib hisobotni ko'rish uchun menyular qatoridan hisobot menyusiga kirib bor maxsulotlar xisoboti bo'limiga kirsak bizga supermarketga kelgan maxsulotlar va sotilgan maxsulotlar xisobotini chiqarishga mo'ljallangan.

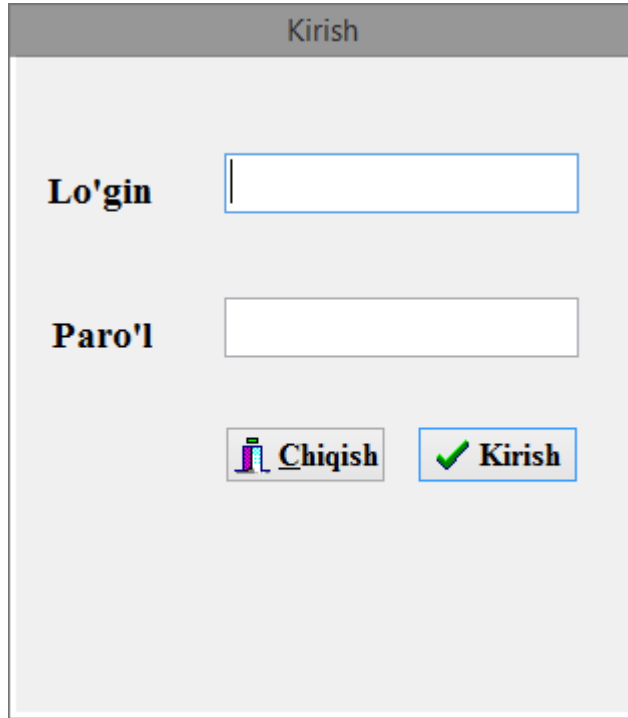
```
WINAPI _tWinMain(HINSTANCE, HINSTANCE, LPTSTR, int)
```

```
{
    try
    {
        Application->Initialize();
        Application->MainFormOnTaskBar = true;
        Application->CreateForm(__classid(TForm1), &Form1);
        Application->CreateForm(__classid(TForm2), &Form2);
        Application->CreateForm(__classid(TBolim), &Bolim);
        Application->CreateForm(__classid(TForm4), &Form4);
        Application->CreateForm(__classid(TForm5), &Form5);
        Application->Run();
    }
}
```

```
    catch (Exception &exception)
    {
        Application->ShowException(&exception);
    }
    catch (...)
    {
        try
        {
            throw Exception("");
        }
        catch (Exception &exception)
        {
            Application->ShowException(&exception);
        }
    }
    return 0;
}
```

2.3 Dastur imkoniyatlari

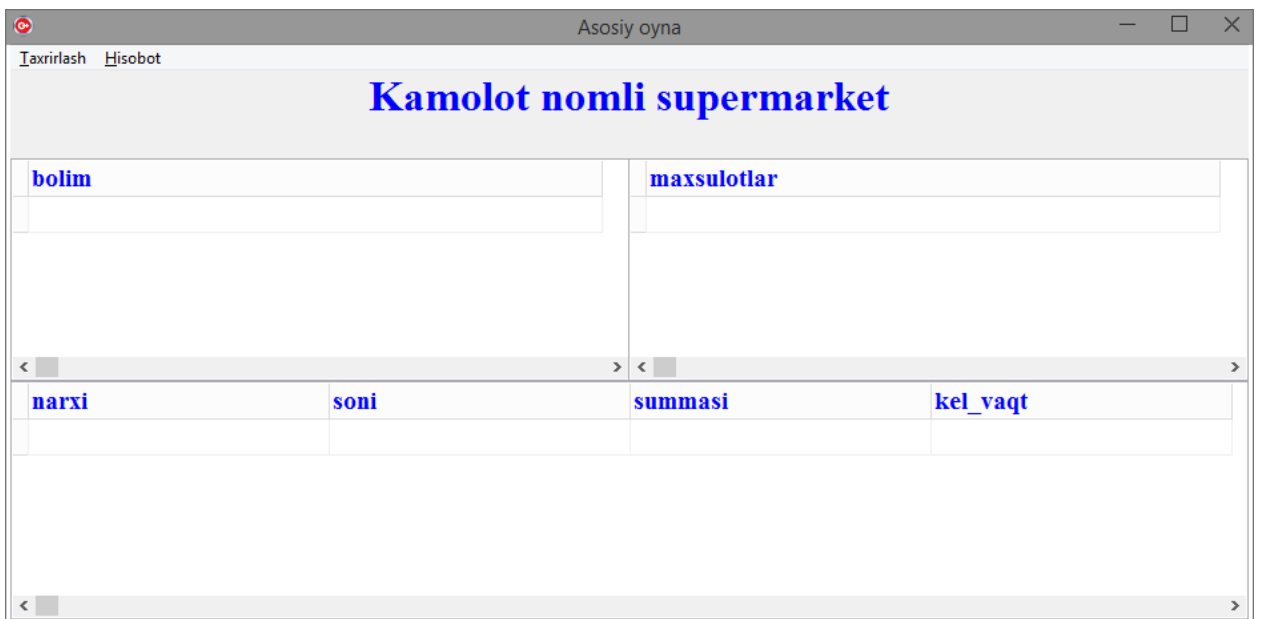
Dastur ishga tushgandan so'ng quyidagi oyna ishga tushadi. Bu oynadan xavfsizlikni taminlash uchun login va parol so'raydi. Bu oynaga loginni va parolni terib asosiy ishchi oynaga o'tamiz login = 1 parol = 2.



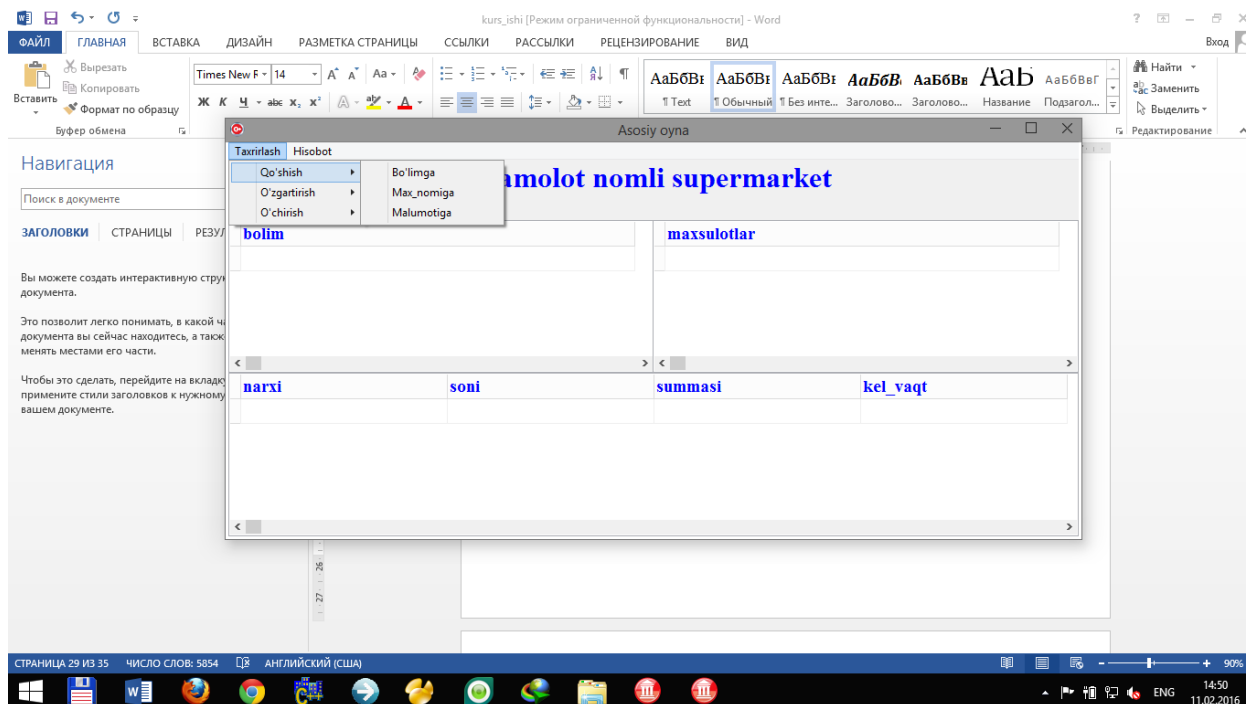
The image shows a login window titled "Kirish". It contains two text input fields: "Lo'gin" and "Paro'l". Below the fields are two buttons: "Chiqish" (Exit) with a door icon and "Kirish" (Login) with a green checkmark icon.

2.9 rasm

Asosiy oyna 2 ta asosiy oyna va 3 ta bo'limdan iborat.



The image shows the main application window titled "Asosiy oyna". The window has a menu bar with "Taxrirlash" and "Hisobot". The main title is "Kamolot nomli supermarket". The window is divided into two main sections: "bolim" (Department) and "maxsulotlar" (Products). The "bolim" section has a table with columns "narxi" (Price) and "soni" (Quantity). The "maxsulotlar" section has a table with columns "summasi" (Total) and "kel_yaqt" (Received). There are scroll bars for both sections.



III. Xulosa.

Yuqorida yaratilgan dastur ham foydalanuvchiga turli qulayliklar yaratadi. 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 foydalanishni, dastur tuzishdan oldin uning algoritmini to'liq tuzib chiqish lozimligini o'rgandim.

IV. Foydalanilgan adabiyotlar

1. C++ Builder в задачах и примерах. Никита Культин. Borland C++ Builder -краткий справочник. Komponentar haqida ma'lumotlar.
2. C++ Builder. Qudrat Abduraximov o'quv dasturi.
3. TATU Farg'ona filiali AT kafedrası. C++ da dasturlash fanidan ma'ruza matnlari.
4. www.ziyonet.uz. Maruza matnlari C++ da dasturlash.
5. dastur.uz, dasturchi.uz. Qudrat Abduraximov sayti. C++ Builder dasturini videolari.

V.Ilovalar.

Dasturdagi kodlardan misollar.

Asosiy forma kodlari:

```
//-----  
  
#include <vcl.h>  
#pragma hdrstop  
  
#include "Unit2.h"  
#include "Unit1.h"  
#include "Unit3.h"  
#include "Unit4.h"  
#include "Unit5.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm2 *Form2;  
//-----  
__fastcall TForm2::TForm2(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
void __fastcall TForm2::FormClose(TObject *Sender, TCloseAction &Action)  
{  
    Form1->Edit2->Text = "";  
    Form1->Edit1->Text = "";  
  
    Form1->Show();  
    Form1->Edit1-> SetFocus();  
}  
//-----  
void __fastcall TForm2::Bolinga1Click(TObject *Sender)  
{  
    ADOTable1->Insert();  
    Bolim->Show();  
}  
//-----  
void __fastcall TForm2::Bolimini1Click(TObject *Sender)  
{  
    ADOTable1->Edit();  
    Bolim->Show();  
}  
//-----
```

```

void __fastcall TForm2::Bolimdan1Click(TObject *Sender)
{
    Form2->ADOTable1->Delete();
}
//-----

void __fastcall TForm2::Maxnomiga1Click(TObject *Sender)
{
    ADOTable2->Insert();
    Form4->Show();
}
//-----

void __fastcall TForm2::Maxsulotni1Click(TObject *Sender)
{
    ADOTable2->Edit();
    Form4->Show();
}
//-----

void __fastcall TForm2::Maxnomidan1Click(TObject *Sender)
{
    Form2->ADOTable1->Delete();
}
//-----

void __fastcall TForm2::Malumotini1Click(TObject *Sender)
{
    ADOTable3->Edit();
    Form5->Show();
}
//-----

void __fastcall TForm2::Malumotiga1Click(TObject *Sender)
{
    ADOTable3->Insert();
    Form5->Show();
}
//-----

void __fastcall TForm2::Malumotidan1Click(TObject *Sender)
{
    ADOTable3->Delete();
}
//-----

```

```
//-----  
  
#include <vcl.h>  
#pragma hdrstop  
  
#include "Unit3.h"  
#include "Unit2.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TBolim *Bolim;  
//-----  
__fastcall TBolim::TBolim(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
void __fastcall TBolim::BitBtn2Click(TObject *Sender)  
{  
    Form2->ADOTable1->Post();  
    Bolim->Close();  
  
}  
//-----  
void __fastcall TBolim::BitBtn1Click(TObject *Sender)  
{  
    Bolim->Close();  
}  
//-----
```