

O'zbekiston Respublikasi Aloqa, Axborotlashtirish
va Telekommunikatsiya Texnologiyalari Davlat
Qo'mitasi Toshkent Axborot Texnologiyalari
Universiteti Farg'ona Filiali

"Axborot texnologiyalari" kafedrası

C++ da dasturlash fanidan

KURS ISHI

Bajardi:

610-13 guruh talabasi

Nematov Ibrohimjon

Qabul qildi

Umarov. Sh

Mulaydinov. F

Tohirov. R

Farg'ona 2015

Mavzu: “Ingliz tilidagi so’zlarni yodda saqlash uchun yordamchi dastur” yaratish.

Reja:

I. Kirish.

II. Nazariy qism.

2.1. C++tili tarixi

2.2. C++Builder muhitida ishlash

2.3. C++ Builder dasturlash tilida ma’lumotlar bazasi bilan ishlash.

III. Amaliy qism.

3.1. Dasturni algoritmi

3.2. Dastur yaratish.

IV. Tatbiq qilish qismi.

V. Foydalanilgan adabiyotlar.

VI. Xulosa.

VII. Ilova.

I.Kirish qismi

O'zbekiston Prezidenti Islom Karimovning 4 fevraldagi Farmoniga asosan, Axborot texnologiyalari va kommunikastiyalarini rivojlantirish vazirligi tashkil etildi. Farmonning O'zAda keltirilgan matniga ko'ra, vazirlik O'zbekiston Aloqa, axborotlashtirish va telekommunikastiya texnologiyalari davlat qo'mitasi negizida tashkil etilgan. Yangi vazirlikning asosiy vazifalari va faoliyat yo'nalishlari sifatida quyidagilar belgilangan: axborot texnologiyalari va kommunikastiyalar sohasida, "elektron xukumat"ni joriy etishda yagona davlat siyosati amalga oshirilishini ta'minlash; aloqa va telekommunikastiyalar sohasidagi faoliyatni, shuningdek, radiochastotali spektrdan foydalanishni davlat yo'li bilan boshqarish, listenziyalash va nazorat qilish borasidagi funkstiyalarni amalga oshirish; Internet tarmog'ining milliy segmenti yanada shakllantirilishini ta'minlash; raqobatdosh dasturiy maxsulotlarning mamlakatimizda ishlab chiqarilishini va ichki bozorini xamda ularga ko'rsatiladigan xizmatlarni rivojlantirishga ko'maklashish; axborot xavfsizligini ta'minlash; zamonaviy kommunikastiya vositalari sohasida ilmiy tadqiqotlar va ishlanmalarni, kadrlarni tayyorlash, qayta tayyorlash va malakasini oshirishni tashkil qilish; vazirlik faoliyati doirasiga kiruvchi boshqa yo'nalishlar bo'yicha xorijiy investistiyalarni jalb etish va boshqalar. "Gazeta.uz"ning qayd etishicha, Aloqa, axborotlashtirish va telekommunikastiya texnologiyalari davlat qo'mitasi 2012 yil 16 oktyabrida Aloqa va axborotlashtirish agentligi negizida tashkil etilgan edi. O'zAAA esa, o'z navbatida, 2002 yili Pochta va telekommunikastiyalar agentligi negizida, ushbu agentlik esa 1997 yili Aloqa vazirligi negizida tashkil etilgan edi. Aloqa vazirligining o'zi 1992 yilda tashkil etilgan.

Biz dasturchilar uchun bu qarorlar juda muhim qarorlardan biri bo'ldi. 2013-yildan 2020-yilgacha Elektron hukumat tizimiga o'tilish talab qilinmoqda. Shuning uchun axborot tizimlariga o'tilishga kata ahamiyat berilmoqda. Yangi o'quv yurtlari va yangi fakultetlar ochilmoqda. Dunyo globallashib borgani sari

axborot texnologiyalarga bo'lgan talab ortib bormoqda. O'zbekiston ham rivojlangan davlatlar qatoriga kirib bormoqda. Bunda esa yoshlarni ongini saviyasini oshirish va zamon talablariga mos ravishda kadrlarni tayyorlash davlatimiz taraqqiyotini rivojlanishiga sabab bo'ladi.

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. 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.

II. Nazariy qism

2.1.C++ tili tarixi.

Birinchi elektron hisoblash mashinalari paydo bo'lishi bilan dasturlash tillari evolyusiyasi boshlanadi. Dastlabki kompyuterlar ikkinchi jahon urushi vaqtida artilneriya 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 saqllovchi (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'ekli 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 `int` 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)

2.2.C++BUILDER MUHITIDA ISHLASH.

Ishlab chiqishning integratsiyalashgan muhiti komponentalar palitrasini birlashtiradi. Shakllar muharriri, kod muharriri, obyektlar noziri, obyektlar xazinasini bular hammasi kod va zaxiralar ustidan to'liq nazoratni ta'minlovchi dasturiy ilovalarni tez ishlab chiqish instrumentlari. Komponentalar palitrasi ilovalarni qurishda taklif qilinadigan 100 dan ortiq takroran qo'llanadigan komponentalardan iborat. Shakllar muharriri dasturning foydalanuvchi bilan interfeysini yaratish uchun mo'ljallangan. Kod muharriri dastur matnini, xususan, voqealarga ishlov berish funksiyalarini yozish uchun mo'ljallangan. Obyektlar noziri qotib qolgan chigal dasturlash zaruratisiz obyektlar xususiyatlarini vizual o'rnatish imkonini beradi hamda shunday voqealarni o'z ichiga oladiki, bu voqealarni ularning paydo bo'lishiga nisbatan obyektlar reaksiyasi kodlari bilan bog'lash mumkin bo'ladi. Obyektlar Xazinasini ma'lumotlarning shakl va modullari kabi obyektlarga ega bo'lib, ular ishlab chiqishda muvaqqat sarflarni kamaytirish maqsadida ko'plab ilovalar bilan bo'linadi. C++ Builder ilovalarni qurishning vizual metodikasini Komponentalar palitrasidan kerakli boshqarish elementlarini tanlab olish vositasida joriy etadi. Har bir komponenta bilan ushbu komponenta turini va xulq-atvorini o'zgartiradigan xususiyatlar bog'liq bo'ladi. C++ Builder 32 razryadli takomillashtirilgan Vizual komponentalar kutubxonasi VCL (Visual Component Library) bilan birgalikda yetkazib beriladi. Bu kutubxona eng murakkab ilovalarni

qurish uchun mo'ljallangan 100 dan ortiq takroran qo'llanadigan komponentalardan iborat. Kutubxonaning asosiy komponentalari. Palitralar komponentalarining instrumental panelida berilgan. Komponentalar belgilari dasturingiz shakliga olib o'tiladi. Kutubxona Windows va Windows 95 operatsiya tizimlaridagi Foydalanuvchi Grafik Interfeysi standart interfeys obyektlarining to'liq inkapsulatsiyalanishini o'z ichiga oladi. Ular orasida, ixtisoslashgan komponentalar bilan bir qatorda, relyatsion ma'lumotlar bazasini boshqarish uchun mo'ljallangan komponentalar alohida o'rin egallaydi. Ishonchli va samarali dasturlarni yaratishda C++Builder obyektga mo'ljallangan dasturlash (OMD) imkoniyatlaridan to'liq foydalanadi.

C++Builder bosh xususiyati avvalambor uning dasturni vizual ishlash jarayonida nafaqat tayyor komponentalardan foydalanish, balki yangi komponentalarni yaratish qobiliyatida ham namoyon bo'ladi. Yangi komponentalar, dastlabki komponentalar kabi, sodda bo'lishi mumkin, bunda ularning funksional imkoniyatlari biroz kengaytirilgan yoki o'zining mutlaqo o'ziga xos ko'rinishi, xulq-atvori va kodining mazmuni bilan farqlanadigan bo'ladi. Komponentalarning yaratilishi OMD ning vorislik mexanizmiga tayanadi, cheklanishlarga deyarli ega bo'lmaydi hamda quyidagi bosqichlardan o'tadi: mavjud komponenta turiga vorislik yangi xususiyatlar, metodlar va voqealarni aniqlash yaratilgan komponentani qayd etish. Qidirish oson bo'lishi uchun Palitra funksional jihatdan o'xshash komponentalarni birlashtiradigan qo'shimcha ilovalar bilan bo'lingan. Tanlab olingan komponentaning kontekst menusini unga sichqonchaning o'ng tugmasini bosib ochish mumkin.

Standart komponentalar

TMainMenu - 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 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 sichqonchani 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 o'rnatsangiz, boshqa komponentaning bir qismini to'g'ri uning ustida joylashtirilgan nom orasidan ko'rinib turadigan qilishingiz mumkin.

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

TMemo - axborot ko'plab satrining tahrir qilinayotgan kiritishidagi to'rtburchak sohani shaklda aks ettiradi. Tahrir sohasining ichidagi boshlang'ich narsalarni Lines xususiyatining qiymati bo'lgan satrlar massivi aniqlaydi. Ushbu

xususiyat qiymati ustunida tugmachani bossangiz, ro'yxat elementlari muharririning darchasi ochiladi. TMemo komponentasi TCustomMemo sinfining to'g'ridan-to'g'ri hosilasi bo'lib, uning barcha xususiyatlari, metodlari va voqealariga vorislik qiladi.

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

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

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 usullarning biri yordamida faollashtirishingiz mumkin: ActivPage xususiyatining tushib qolayotgan ro'yxatidan tanlab olingan sichqoncha yordamida shuningdek, kontekst menuning NextPage va Previous Page opsiyalari yordamida qo'shimcha ilovalarni varaqlash vositasida. PageIndex xususiyati faol sahifa raqamiga ega. Tab Visible xususiyatining false qiymatini o'rnatib, shu sahifani ko'rinmas qilish mumkin.

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

TBitBtn - 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 - 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.

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'rnating. Kichikroq o'lchamdagi dastlabki tasvir butun konteynerga cho'zilib ketishi uchun Stretch xususiyatining true qiymatini o'rnating. Tasvirlar fayllarining dinamik yuklanishi va saqlanishi uchun Picture obyekt xususiyatining LoadFromFile va SaveToFile metodlaridan quyidagi turlar yordamida foydalaning:

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

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

TScrollBox - darchada o'lchamlari o'zgaruvchan boksni yaratadi, bu boks shu topdayoq avtomatik tarzda zaruratga ko'ra aylantirish lineykalari bilan ta'minlanadi. Aylantirib ko'rish boksi yordamida darchaning ayrim sohalarini aylantirib ko'rishdan himoyalash mumkin. Masalan, instrumentlar paneli va holat panelini himoyalash uchun avval darchani aylantirish lineykasini berkitib qo'ying, keyin esa aylantirish boksini mijoz sohasida instrumentlar paneli va holat paneli o'rtasida joylashtiring. Boksni aylantirib ko'rish lineykasi darchaga tegishli bo'lib ko'rinadi, biroq aylantirish faqat boks ichida amalga oshiriladi. Aylantirib ko'rish bokslaridan yana boshqachasiga ham foydalanish mumkin: ular biror bir darchada ko'plab aylantirib ko'rilayotgan sohalar (turlar) yaratish imkonini beradi. Turlar ko'pincha tijoriy matn protsessorlarida, buxgalteriya dasturlarida va loyihalarni rejalashtirish dasturlarida qo'llanadi. Aylantirib ko'rish boksi boshqa komponentlarga, masalan, Tbutton va TCheckBox ga ham ega bo'lishi mumkin.

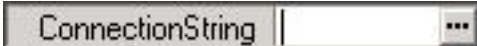
2.3. C++ Builder dasturlash tilida ma'lumotlar bazasi bilan ishlash.

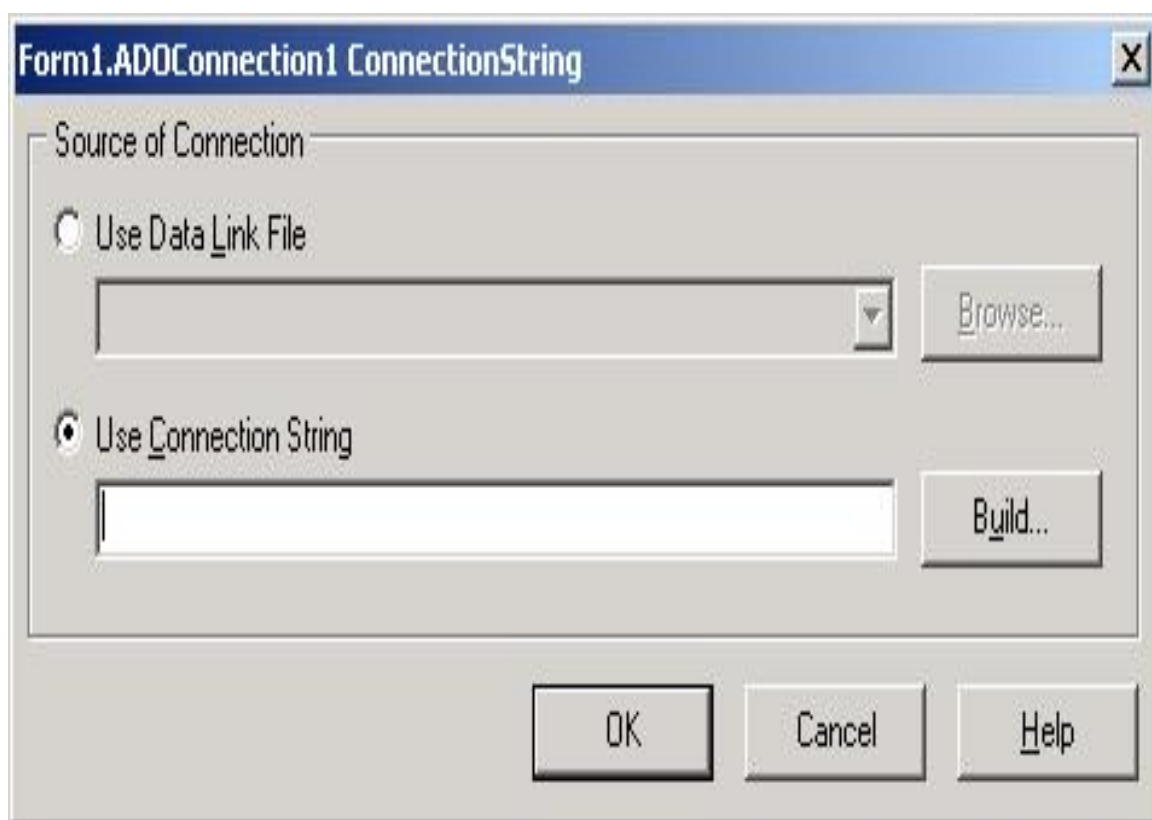
C++ Builder dasturlash tilini ma'lumotlar ombori bilan bog'lash turlai ko'p bo'lib ulardan biz ADO komponentalar to'plamiga tegishli bo'lgan ADOConnection, ADOTable va DataAcess komponentalar to'plamiga tegishli bo'lgan DataSource komponentalardan foydalanamiz.

Data Source bo'limi

Komponentaning ko'rinishi	Komponentaning Nomi	Komponentaning vazifasi
	ADOConnection	bu component ma'lumotlar bazasini Builderdasturlash tili bilan bir-biriga bog'laydi
	ADOTable	Bu component ma'lumotlar bazasidagi jadvalni Builder dasturlash tili bilan bir-biriga

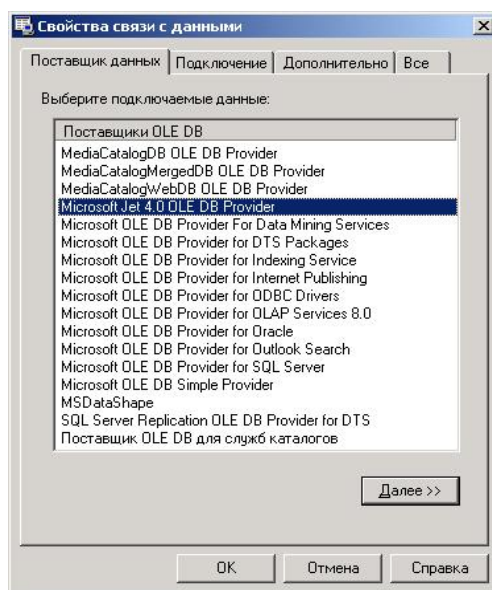
		bog'laydi
	DataSource	Bu komponent malumotlar bazasidagi jadvaldagi ma'lumotni inson ko'ra oladigan ko'rinishga keltiradi va aksincha.

Bu komponentlarni C++Builder dasturlash tilining proect oynasiga ya'ni formaga o'rnatamiz. Komponentlardan ADOConnectionni tanlab Object Inspector oynasidagi Properties(Xususiyat) bo'limidagi ConnectionString  bandini tanlaymiz. Bu banddagi  tugmani chertamiz. Shunda quyidagi oyna hosil bo'ladi. (2.3.1 – rasm)



(2.3.1 – rasm).Bazani C++Builder dasturlash tili bilan bog'lash oynasi.

Bu oyndagi **Build** tugmasini chertamiz. Shunda quyidagi oyna hosil bo'ladi.



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

Bu oynadagi **Подключение** oynasiga o'tib bazani qayerda turganini ko'rsatamiz va **OK** yugmasini bosamiz. Shundan so'ng ADOTable komponentini aktivlashtiramiz. Uning xususiyatlar oynasidan Connection xususiyatini tanlab komponentani ADOConnection komponentasi bilan bog'laymiz. DataSource komponentasi xususiyatlar oynasidan DataSet xususiyatini tanlaymiz va ADOTable komponentasini bir biriga bog'laymiz. Ma'lumotlar ombori bilan ishlaydigan komponentlar ro'yhati.

Data Access bo'limi

	DataSource –ma'lumotlarni yoki komponentlarni bir-biri bilan bog'lash.
	Table –ma'lumotlar omborini (faylni) bog'lash.
	Query –ma'lumotlar omboridagi yozuvlarni boshqarish.
	StoredProc – serverdan ma'lumotlar omborini yuklash.

	Database – yagona ma’lumotlar omborini bog’lash
---	---

Data Controls bo’limi

	DBGrid –ma’lumotlar omboridagi yozuvlarni jadval ko’rinishida chiqarish
	DBNavigator – ma’lumotlar omboridagi yozuvlarni taxrirllovchi component. Undagi yangi so’z qo’shish , o’zgartirish,o’chirish va x.k. ishlarni qilish mumkin.
	DBText –ma’lumotlar omboridagi matnli maydon ma’lumotlarini chiqarish..
	DBMemo – ma’lumotlar omboridagi memo tipidagi ma’lumotlarni taxrirlash.
	DBListBox –ma’lumotlar omboridagi ma’lumotlarni listga chiqarish
	ComboBox –ma’lumotlarni kombinatsiyali tanlash.
	DBRichEdit – ma’lumotlarni taxrirlashning memo ga nisbatan kengroq imkoniyati.

Ma’lumotlar omboridagi kiritilgan ma’lumotlarni taxrirlashimiz yangi ma’lumot qo’shish imkoniyatiga ega bo’lamiz C++Builder bilan ishlanganda.

C++Builder bosh xususiyati avvalam bor uning dasturni vizual ishlash jarayonida nafaqat tayyor komponentalardan foydalanish, balki yangi komponentalarni yaratish qobiliyatida ham namoyon bo’ladi. Yangi komponentalar, dastlabki komponentalar kabi, sodda bo’lishi mumkin, bunda ularning funktsional imkoniyatlari sal-pal kengaytirilgan yoki o’zining mutlaqo o’ziga xos ko’rinishi, xulq-atvori va kodining mazmuni bilan farqlanadigan bo’ladi.

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.

TLevel - xuddi iskana bilan o'yilgandek xajmli ko'rinadigan chiziqlar, bokslar yoki ramkalarniyaratadi. Komponenta chizayotgan ob'ektni Shape xususiyati aniqlaydi, Style xususiyatining qiymati esa ob'ekt ko'rinishini o'zgartirib, uni bo'rtiq yoki botiq holga keltiradi. Foydalanuvchi shakl o'lchamlarini o'zgartirganda ham ob'ektning nisbiy holatini o'zgarmas qoldirish uchun, Align xususiyatining true qiymatini o'rnating.

TImage - shaklda grafik tasvir konteynerini yaratadi (bu bit obrazi, piktogramm 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'rnating. Kichikroq o'lchamdagi dastlabki tasvir butun konteynerga cho'zilib ketishi uchun, Stretch xususiyatining true qiymatini o'rnating. Tasvirlar fayllarining dinamik yuklanishi va saqlanishi uchun, Picture ob'ekt xususiyatining LoadFromFile va SaveToFile metodlaridan quyidagi turlar yordamida foydalaning:

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

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

Vizual komponentalarning umumiy xususiyatlari

Barcha vizual komponentalar uchun TControl sinfi asos hisoblanadi va u elementning o'lchami va joylashuvi, uning sarlavhasi, rangi va shunga uxshash parametrlaridan iborat asosiy funksional atributlarni ta'minlaydi. TControl sinfi

vizual komponentalar uchun umumiy bo'lgan xossalar, hodisalar va metodlarni o'z ichiga oladi. Vizual komponentalarni ikkita katta guruhga ajratish mumkin:

1. To'g'ri to'rtburchakli boshqaruv elementlari;
2. To'g'ri to'rtburchakli bo'lmagan boshqaruv elementlari.

To'g'ri to'rtburchakli boshqaruv elementi o'zida ma'lum bir maqsad uchun aniqlangan maxsus to'g'ri to'rtburchakni ifodalaydi. Bu elementlarga misol tariqasida boshqaruv tugmalarini, tahrir maydonlari, harakatlanish yo'laklarini ko'rsatishimiz mumkin. Ular uchun asos sinf TWinControl hisoblanadi.

To'g'ri to'rtburchak elementlari qiymar kiritish fokuslarini ilishi mumkin . Elementni fokus olganligi ikki xil usulda kursatiladi :

1. Tahrir kursori yordamida;
2. To'g'ri to'rtburchak orqali.

Matn tahrirlari bo'lgan Edit va Memo komponentalari o'z sohasida tahrir kursori (matn kursori) paydo bo'lishi orqali fokus (boshqaruvni) olganligini bildiradi.

Mantlarni tahrirlash bilan bog'liq bo'lmagan komponentalarda qora punktir chiziqli to'g'ri to'rtburchak paydo bo'lishi uning fokus olganligini anglatadi.

Masalan, Button tugmasi fokus olganda sarlavha atrofida to'g'ri to'rtburchak paydo bo'ladi, ListBox komponentada esa ro'yxatdagi ayni paytda tanlangan satrni ajratilgan holda (aksariyat hollarda ko'k fonda) ko'rsatishi boshqaruvni olganligini bildiradi. Bulardan tashqari, to'g'ri to'rtburchak boshqaruv elementlari konteyner sifatida o'z ichida boshqa boshqaruv elementlarini olishi mumkin. Bu holda boshqaruv elementi o'z ichidagilarga ota hisoblanadi.

To'g'ri to'rtburchak bo'lmagan boshqaruv elementlari TGraphicControl sinfining avlodlari hisoblanadi. Bu guruh elementlarini qiymat kiritish fokusini olmaydi va interfeys elementlari uchun "ota" bo'la olmaydi. To'g'ri to'rtburchak bo'lmagan boshqaruv komponentalarining afzalligi - ularni nisbatan kam resurs talab qilishida.

C++ Builder dasturlashda vizual komponentalarning xossalari

Xossalar ilovalar yaratilishi va boshqarilishida komponentalar tashqi ko'rinishi va amal qilishini boshqarish imkoniyatini beradi. Odatda komponenta xossasining qiymatlari ilovalarni yaratish vaqtida Ob'ektlar Inspektori yordamida amalga oshiriladi. Keltiriladigan misollarda tushunarli bo'lishi uchun xossalar qiymatlari qiymat berish operatori yordamida amalga oshiriladi. Shuni ta'kidlab o'tish kerakki, komponentalar barcha xossaga ega bo'lmasligi mumkin. Masalan, Edit tahriri Caption xossasiga ega emas, Label yozuvi ReadOnly xossasiga ega emas va hokazo.

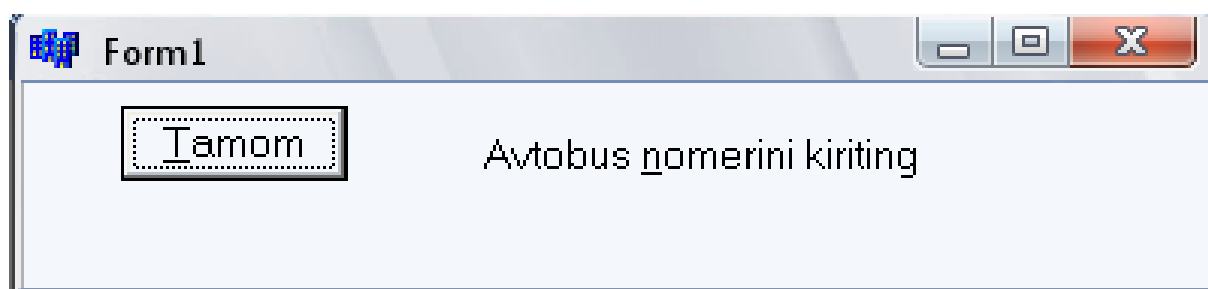
Quyida komponentalarda uchraydigan asosiy xossalarga izox beramiz.

Caption - xossasi komponenta sarlavhasini yozish satrini o'z ichiga oladi. Sarlavha satridagi ayrim belgilar tagiga chizilgan bo'lishi mumkin, ular tezkor murojaat tugmalarining kombinatsiyasini bildiradi. Ko'rsatilgan belgini <Alt> tugmasi bilan bir vaqtda bosilishi shu sarlavhadagi komponentaga sichqonchanning tugmasini bosish bilan bir xil amalni yuzaga keltiradi. Tezkor murojaat belgisini, shu belgi oldiga '&' belgisini quyish orqali belgilanadi, masalan:

Label1->Caption="Avtobus &nomerini kiriting";

Button1->Caption="&Tamom";

Shuni qayd etish kerakki, Caption xossasining qiymatlarini Object inspector darchasi Properties varagidagi mos qatoridagi satr maydonida kiritish ham mumkin. Ilova ko'rinishi quyidagicha bo'ladi:



3.3-rasm

Align - xossasi komponentani u joylashgan konteyner ichidagi joylashuv variantlarini aniqlaydi. Aksariyat hollarda konteyner sifatida Form formasi yoki Panel paneli keladi.

Align xossasi quyidagi qiymatlarning birini qabul qilishi mumkin:

alNone - to'g'rilash amalga oshirilmaydi. Komponenta ilovani yaratish paytida qaerga joylashtirilgan bo'lsa, shu joyda qoladi;

alTop - komponenta konteynerning yuqori qismiga ko'chiriladi, komponenta balandligi o'zgarmaydi, eni esa konteyner eniga teng bo'ladi;

alBottom - AlTop ga o'xshash, faqat komponenta konteyner pastiga joylashadi;

alLeft - komponenta konteynerning chap tomoniga ko'chadi, eni o'zgarmaydi, buyi esa konteyner buyiga tenglashadi.

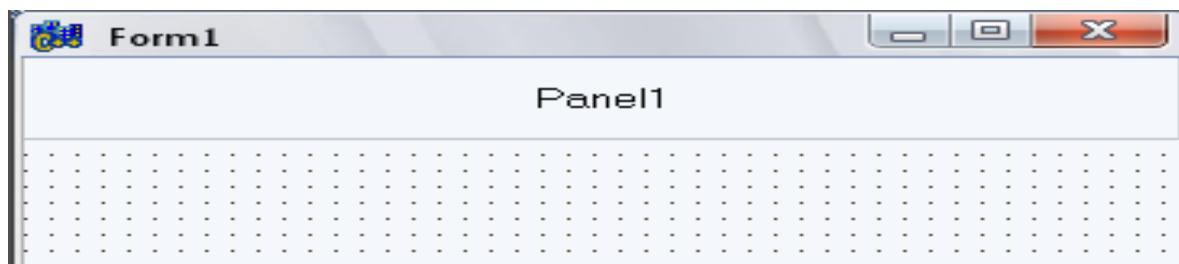
alRight - alLeft ga o'xshash, faqat komponenta konteynerning o'ng tomoniga ko'chadi;

alClient - komponenta konteynerni to'la egallaydi.

Misol. Panelni formaga nisbatan tog'rilash.

Panel1->Align=AlTop;

Natijada Panel1 paneli Form1 formaning yuqori qismiga joylashadi.



3.3-rasm

Color - komponenta fonining rangini aniqlaydi. Color xossasining qiymati 4 baytli 16 sanoq sistemasidagi son bo'lib, uning katta bayti rang palitrasini aniqlaydi (odatda \$00), kichik uchta bayti qizil, yashil va ko'k ranglarning RGB

intensivligini aniqlaydi. Bu ranglarni turli nisbatdagi aralashmasidan natijaviy rang xosil bo'ladi. Masalan, \$000000 - qora, \$FFFFFF - oq, \$0000FF - qizil, \$00FF00 - yashil, \$FF0000 - ko'k rangni beradi. Ranglar qiymatini konstantalar orqali berish ancha qulay. Masalan, clAqua - Och ko'k (\$FFFF00), clBlack – Qora (\$000000) va hakoza.

Ctr3D - xossasi boolean turida bo'lib, vizual komponenta ko'rinishini aniqlaydi. Agar Ctr3D qiymati false bo'lsa komponenta ikki o'lchamli tasvir kurinishida, agar true bo'lsa - uch ulchamli tasvirlanadi (kelishilgan holdagi ko'rinish).

Cursor - xossasi sichqoncha ko'rsatgichining ko'rinishini aniqlab beradi. Bu ko'rinishlar rang barang, ular ichida eng asosiylari quyidagi konstantalar bilan beriladi :

crDefault - sichqoncha ko'rsatgichi kelishuv bo'yicha (odatda strelka) ;

crNone - ko'rsatgich ko'rinmaydi ;

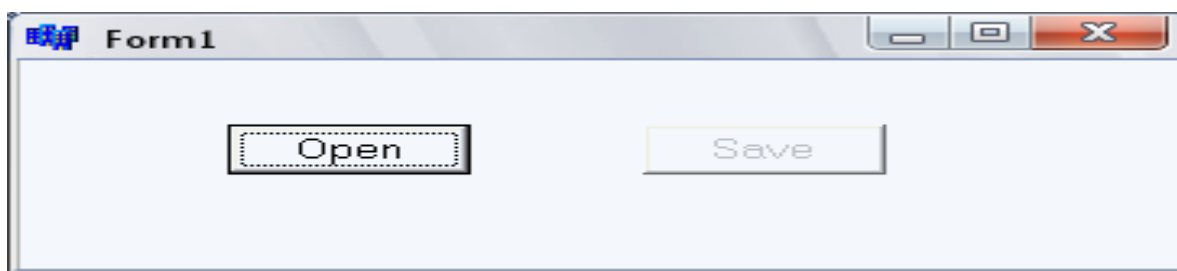
crArrow - ko'rsatgich strelka ko'rinishida;

crCross - ko'rsatgich xoch ko'rinishida;

crHourGlas - ko'rsatgich qum soati ko'rinishida.

Enabled - xossasi boolean turida bo'lib komponentani faolligini, ya'ni sichqonchadan yoki klaviaturadan kelayotgan signallarga aks ta'sir bera olishini aniqlaydi. Agar xossa true (kelishilgan holat qiymati) qiymatga ega bo'lsa, komponenta faol hisoblanadi. Komponentaning faol bo'lmagan holatini sarlavhaning yoki matnning oqargan rangda ekanligi bildiradi. Ayrim hollarda qandaydir komponenta o'chirilgan (blokirovka qilingan) bo'ladi, agar u bilan bog'liq qandaydir amalni bajarishni iloji bo'lmasa. Masalan, Save tugmasi xujjatni saqlash amalini bajaradigan bo'lsin, agar saqlanadigan xujjatning o'zi bo'lmasa, saqlash amalining ma'nosi yo'q. Bunday hollarda tugma faol bo'lmagan holatga o'tkaziladi:

Save->Enabled=false;



3.4-rasm

Font - xossasi vizual komponenta akslanuvchi matn shriftini aniqlaydi. O'z navbatida TFont sinfi shrift parametrlarini boshqarishga imkon beruvchi xossalariga ega. Bularning ichida asosiylari quyidagilar:

Name - Shrift nomini aniqlaydi;

Size - Punktlarda shrift o'lchamini beradi;

Style - Shrift Stilini beradi;

Color - Matn rangini boshqaradi.

Masalan, Label1 yozuvining rangini berish:

Label1->Font->Color=clGreen;

Label1->Color=clWhite;

Label1 sarlavhasi yashil fonddagi oq rangdagi matn ko'rinishida bo'ladi.

Height va **Width** xossalari komponentaning mos ravishda vertikal va gorizontal o'lchamlarini piksillarda beradi.

Left va **Top** xossasi komponentaning u joylashgan konteynerga (forma yoki panelga) nisbatan koordinatalarini aniqlaydi. O'z o'rnida forma ham komponenta va uning koordinatalari ekranning chap yuqori burchagiga nisbatan aniqlanadi.

Hint xossasi kursor komponenta sohasida va bir necha soniya harakatsiz holatda bo'lganda ekranga yordamchi matnni chiqaradi. Yordamchi matn sariq fonda chiqadi va komponentaning ishlatilish maqsadini qisqacha yoritadi. Yordamchi matn ekranda paydo bo'lishi uchun boolean turidagi **ShowHint** xossasining qiymatini true deb aniqlash zarur.

PopupMenu xossasi suzib chiquvchi lokal menuga ko'rsatadi. Bu menyu sichqoncha ko'rsatgichi komponenta maydonida (sohasida) turgan holda sichqonchaning o'ng tugmasi bosilganda paydo bo'ladi va bu holat ro'y berishi uchun **AutoPopup** xossasiga (boolean turidagi) true qiymatini berish kerak. Kelishuv buyicha uning qiymati false bo'ladi.

TabOrder xossasi konteynerdagi komponentalarning fokus olish tartibini aniqlaydi ("Tab" tugmasi bosilganda), ya'ni komponentalar "aylanib" chiqish ketma - ketligini aniqlaydi. Kelishuv bo'yicha bu ketma - ketlik formani ko'rinishini yaratishda komponentalarni konteynerga joylashtirish tartibiga mos keladi: birinchi komponentaning TabOrder xossasining qiymati 0, ikkinchisining - 1 va hakoza. Tartibni o'zgartirish uchun komponentaning TabOrder xossasiga zarur qiymatni berish kerak. Har bir konteyner boshqalariga bog'liq bo'lmagan tabulyasiya tartibiga ega bo'ladi. Ikkita komponenta bir xil tabulyatsiya tartibiga ega bo'lishi mumkin emas.

III. Amaliy qism

3.1. Dasturni algoritmi.

Dasturni yaratishda avvalo ma'lumotlar bazasini, undagi jadvallarni, jadvaldagi ustunlarni to'g'ri kiritish lozim. Jadval, ustun nomlarini yozishda probel, apostrof(') belgilaridan foydalanmaslik zarur. Chunki dasturda SQL so'rovlardan foydalanganda xatoliklar sodir berishi mumkin. Ingliz tilidagi sozlarni yodda saqlash uchun yordamchi dastur uchun so'zlar tarjimasi ma'lumotlar bazasi hosil qilib olamiz. Ingliz tilidagi so'zlar bazasida inglizcha – o'zbekcha so'zlar lug'at joylashgan bo'ladi. Bazada Inglizcha - Uzbekcha nomli jadval mavjud. Bu jadvalda ingliz tilidagi mavjud so'zlar haqida ma'lumotlar bor. Inglizcha-Uzbekcha jadvali "Inglizcha sozlar", "Uzbekcha sozlar" ustunlaridan tashkil topgan. Yana bir jadval yaratamiz uni nomini "Uzbekcha-Inglizcha sozlar" deb nomlaymiz. Undagi ikkita ustunni "Uzbekcha sozlar", "Inglizcha sozlar" deb nomlab olamiz. Shu bilan ma'lumotlar bazasini ham tuzib oldik. Ma'lumotlar bazasini tuzib olganimizdan so'ng dasturning dizayn qismini yaratib olamiz. Dasturning dizayn qismi va dastur kodlarini kiritish uchun C++ Builder 6 dasturidan foydalandim. C++ Builder 6 dasturini ishga tushiramiz. File menyusidan New->Application buyruqlar ketma-ketligini kiritamiz. Shundan so'ng dastur yaratishimiz uchun yangi forma hosil bo'ladi. Bu yerga o'zimizga kerakli bo'lgan komponentalardan foydalanib dasturning dizayn qismini yaratib olamiz. Dasturni bazaga bog'lash uchun ADOConnection, ADOTable, DataSource komponentalaridan foydalanamiz. Bizda yana bir ma'lumotlar bazasi bo'ladi unda o'yinga kirgan foydalanuvchilar ro'yhatga olinadi. Bu bazani "Ma'lumotlar" deb atadim unda ustunlar quyidagicha bo'ladi: "FISH", "guruh", "to'g'ri so'zlar", "notog'ri so'zlar", "sana" bu bilan biz o'yin oynovchilarni tekshirib turishimiz mumkin.

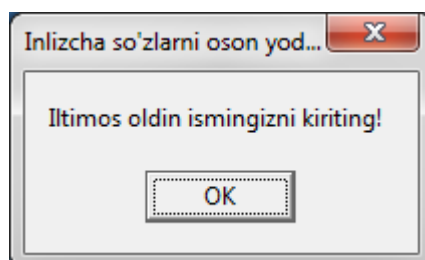
3.2. Dastur yaratish.

Bizning asosiy formamiz quyidagicha. Dastur ishlatilganda quyidagi forma hosil bo'ladi. Yuqoridagi oyna dasturning titul qismi bo'lib, unda dasturga kirishda ismini va guruhini so'raydi.



3.2.1-rasm Dasturni kirish qismi

Agar ismini kiritmay Kirish tugmasini bosak:



(3.2.2-rasm) Ismini kiritmay dasturga kirish

tugmasi hosil bo'ladi. Lug'atda so'zlar bo'lib uni tarjimasini kiritilishi kerak har bir urinishda bizning hisoblovchimiz orqali to'g'risini sonini noto'g'risini sonini aytib boradi. Bizda ikkita oyna mavju bo'lib birinchi oynada "Inglizcha – O'zbekcha" lug'at oynasi hosil bo'lib unda so'zning "O'zbekcha" nomi kiritiladi.



3.2.3-rasm O'zbekcha kiritish qismi

Ikkinchi oynada esa “Inhizcha – O’zbekcha” lug’atning “Inglizcha” so’zlarni kiritish oynasi hosil bo’ladi.



3.2.4-rasm Inglizcha kiritish qismi

“Kirish” qismidagi komponentalarga birma – bir izox berib o’tsak.

Tarix – komponentasida foydalanuvchi dasturdan oldin foydalangan shaxslar to’g’risida ma’lumot olishi mumkin ya’ni:

Print Preview

Close

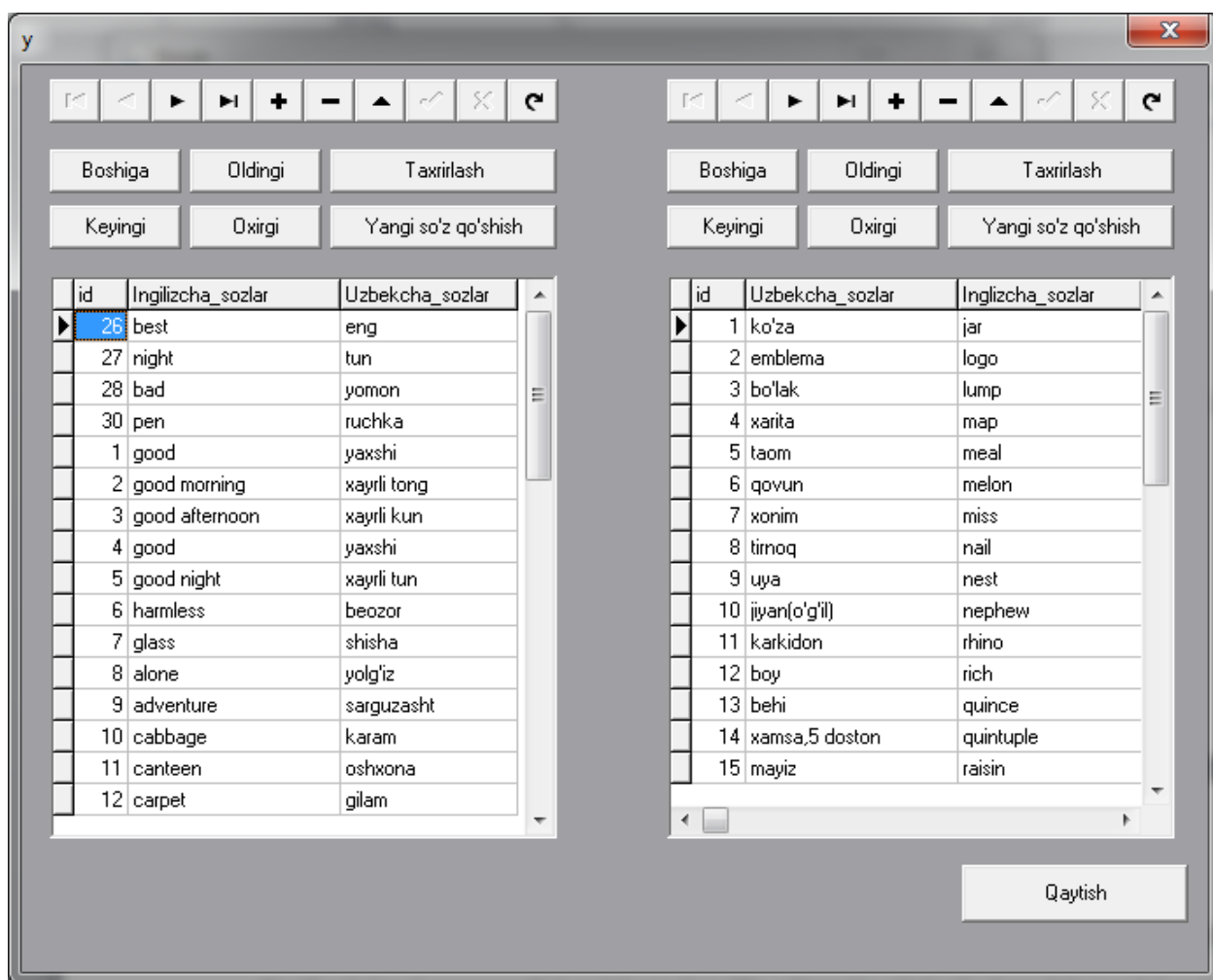
Foydalanganlar ro'yxati

Ism	Guruh	To'g'ri so'zlari	Noto'g'ri so'zlari	Vaqli
545	45546			
KK	TYUJIK			
aaaa	11			
dsf	dsd			
jdha	dds			
sdf	sdf			
dsfj	sdf			
doc	jf			

Page 1 of 2

3.2.5-rasm. Tarix oynasi

Menu – komponentasiorqali lug'atdagi soz'lar haqida ma'lumotlarni olishimiz mumkin va undagi so'zlar ustida bir qancha amallarni bajarishimiz mumkin, misol uchun yangi so'z qo'shishimiz yoki so'zlarni taxrirlashimizni aytishimiz mumkin.



3.2.6-rasm.Dasturdagi so'zlar ro'yxati

Dasturchi haqida – bunda biz dasturni yaratgan shaxs to'g'risida kerakli ma'lumotlarga ega bo'lamiz.

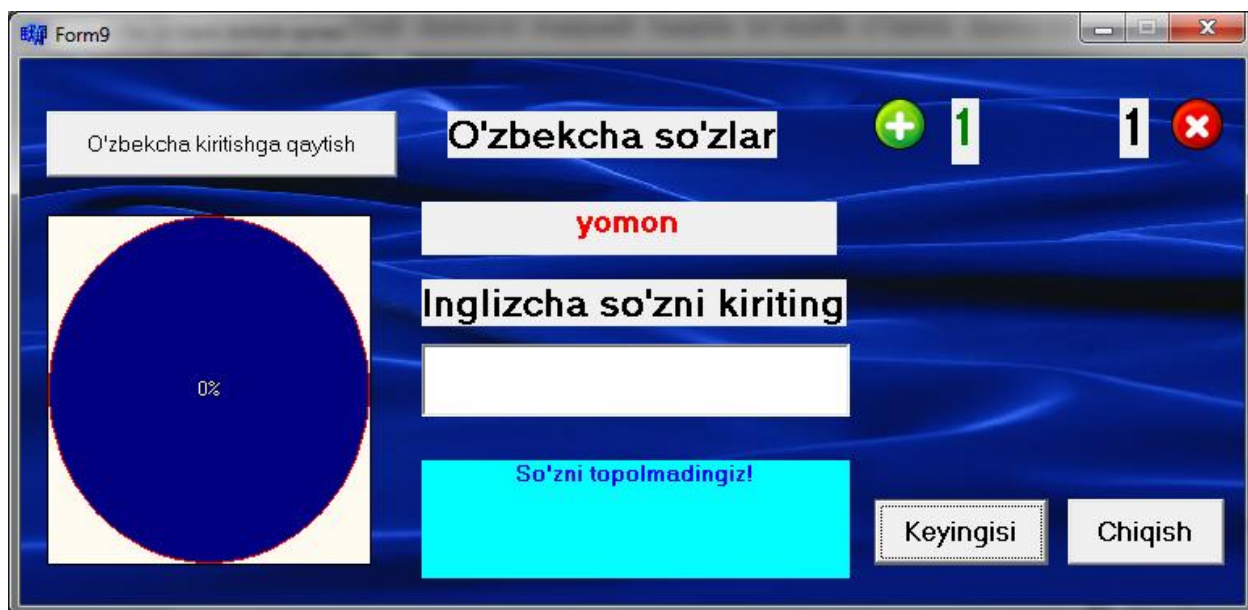


3.2.7-rasm.Dasturchi haqida oyna

Chiqish – bu dasturni yopish uchun xizmat qiladi, ya’ni dasturni to’liq yopadi.

Endi dasturni maqsadi haqida to’xtalib o’tamiz dasturda har bir tog’ri so’zlarni to’g’ri, noto’g’ri so’zni esa noto’g’ri degan yozuv chiqarib hisoblab boradi.





3.2.8-rasm.Dasturda hisoblagich

Endi dasturni kod qismi bilan ishlaymiz va yangi kodlar bilan tanishib chiqamiz:

```
{if(edtEn->Text==dbeEn->Text)
{lblNatija->Caption="So'zni to'g'ri topdingiz!";
edtEn->Text="";
togriSozlar++;
lblTogri->Caption=IntToStr(togriSozlar);}
else{lblNatija->Caption="So'zni topolmadingiz!";
notogriSozlar++;
lblNotogri->Caption=IntToStr(notogriSozlar);}
if(hisoblagich>sozSoni){
Button2->Enabled=false;
lblNatija->Caption="Urinishlar soni tugadi.\nShundan to'g'ri topilgan \nso'zlar
soni:"+IntToStr(togriSozlar)+
"\n Noto'g'ri topilgan \nso'zlar soni:"+IntToStr(notogriSozlar);}
else{
frmYordamchi->atEnUz->Next();
```

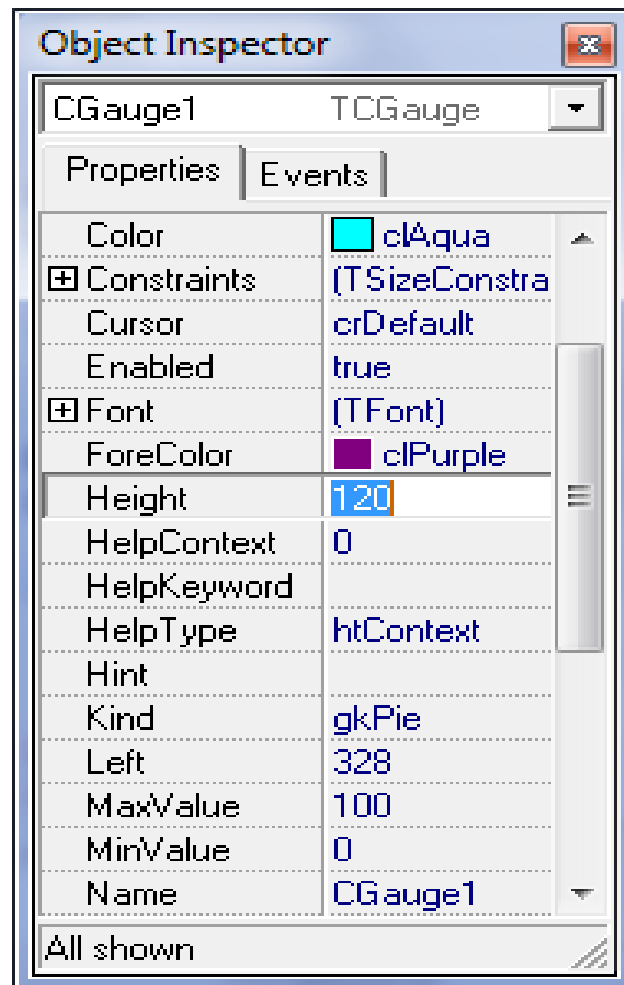
```
edtEn->Text="";  
hisoblagich++;  
CGauge1->Progress=hisoblagich;}
```

Endi bu kod qismida o'yinimizni asosiy qismi bajariladi C++ Builder muhitida ishlashda Yordamchi formalar bilan ishlash uchun va chiqish tugmasi bosilganda quyidagi buyruq beriladi:

```
{ShowMessage("Foydalanuvchi "+frmkirish->ADOTable1->FieldByName("fish")-  
>AsString+  
" siz berilgan "+IntToStr(sozSoni)+" so'zdan "+IntToStr(togriSozlar)+" ta to'g'ri  
"+IntToStr(notogriSozlar)+" ta noto'g'ri so'z topdingiz!");  
frmkirish->ADOTable1->Edit();  
frmkirish->ADOTable1->FieldByName("togri")->AsInteger=togriSozlar;  
frmkirish->ADOTable1->FieldByName("notogri")->AsInteger=notogriSozlar;  
frmkirish->ADOTable1->FieldByName("vaqt")-  
>AsDateTime=Now().CurrentDateTime();  
frmkirish->ADOTable1->Post();  
Close();}
```

Endi dasturda ishlatilgan komponentalardan CGauge bilan ishlash qismiga o'tamiz. Bu komponenta uchun kod qismi:

```
hisoblagich=0;  
CGauge1->MaxValue=sozSoni;  
CGauge1->Progress=hisoblagich;
```



3.2.9-rasm

V .Tatbiq qilish qismi.

Men bu dasturni ishlab chiqish jarayonida juda ko'p ma'lumotlarga ega bo'ldim. Ma'lumotlar bazasiga bir nechta jadvalni ulash va yana bir nechta formalarni bir biriga ulashni o'rgandim, bir formadan turib ikkinchi formani chaqirib, bir tugma orqali keyingi formaga o'tish mumkin ekan kod qismida Show() buyrug'i orqali forma chaqirilib olinar ekan. Ingliz tilidagi so'zlarni yodda saqlash uchun yordamchi dasturni yaratish jarayonida ingliz tilini lug'at qismi bilan ham tanishib ko'p so'zlarni o'rganib oldim. Hozirgi kunda yoshlarimizga ko'p imkoniyatlar yaratilib berilmoqda dunyo rivojlanib borgan sari ingliz tili xalqaro til darajasiga chiqdi. Shuning uchun bizda ham ingliz tiliga bo'lgan e'tibor kuchayib borayapti. Buning natijasida ingliz tilini o'rganish qulay va qiziqarli bo'lishi uchun har xil dasturlar ishlab chiqilmoqda.

Men ham shuni e'tiborga olib uncha murakkab bo'lmasada ingliz tilidagi so'zlarni yodda saqlash uchun biroz bo'lsada esda qolishi uchun manashu mavzuni kurs ishim uchun mavzu sifatida tanladim. Dastur lug'at sifatida ingliz tilini o'rganish juda ham oson ayniqsa maktab o'quvchilari uchun qiziqarli va qulay yaratilgan. Lug'at tarzida yaratilgan dasturi inson miyasida fikrlash qobiliyatini rivojlantiradi , qiziqish orqali so'zlarni o'rganib olinadi. Dasturni yaratish jarayonida bir nechata C++ Builderga oid ma'lumotlariga ega bo'ldim masalan CGauge komponentasi haqida ma'lumot oldim. Hisoblagichlarni ishlatishni o'rgandim o'yindagi so'zlar o'tgani sari hisoblagichda o'zgarish bolyapti va nihoyat oxirida natijada hisoblagich ishga tushadi.

VII.Xulosa

Vaqt o'tishi bilan kompyuterlar tobora kengroq qo'llana boshladi hamda yuqoriroq darajadagi protsedura tillari paydo bo'ldi. Bularning dastlabkisi FORTRAN tili edi. Biroq obyektga mo'ljallangan yondoshuv rivojiga asosiy ta'sirni keyinroq paydo bo'lgan, masalan, ALGOL kabi protsedura tillari ko'rsatdi. 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 funktsiyalarni nafaqat tushunish, balki sozlash ham osonroq kechadi. Biroq, boshqa tomondan, protsedurali dasturlash koddan takroran foydalanish imkonini cheklab qo'yyadi. Buning ustiga dasturchilar tez-tez «makaron» dasturlar ham yozib turishganki, bu dasturlarni bajarish likopdagi spagetti uyumini ajratishga o'xshab ketar edi. Va, nihoyat, shu narsa aniq bo'ldiki, protsedurali dasturlash usullari bilan dasturlarni ishlab chiqishda diqqatni ma'lumotlarga qaratishning o'zi muammolarni keltirib chiqarar ekan.

C++ Builder modulli dasturlash keyingi mantiqiy pog'onani egallaydi, u modulga nasldan-naslga o'tishni va polimorfizmni qo'shadi. C++ Builderdan foydalanr ekan, dasturchi dasturni bir qator oliy darajali ob'ektlarga bo'lish yo'li bilan tizimlashtiradi. Har bir ob'ekt hal qilinayotgan muammoning ma'lum bir tomonini modellashtiradi. C++Builder endilikda dasturni bajarish jarayonini boshqarish uchun dasturchi diqqatini protseduralarni ketma-ketlikda chaqirib olish ro'yxatini tuzib o'tirishga qaratmaydi. Buning o'rniga ob'ektlar o'zaro aloqada bo'ladi. Builder yordamida ishlab chiqilgan dastur hal qilinayotgan muammoning amaldagi modeli bo'lib xizmat qiladi.

Men ushbu kurs ishini bajarish davomida C++ dasturlash tili da dasturlash hamda avtomatlashtirish to'g'risidagi bilimlarimni mustaxkamlab oldim. Dastur tuzish davomida C++ Builder da ishladim. C++ Builder ko'plab komponentalarni ishlatdim. Kurs ishini bajarish davomida dasturdagi ko'plab komponentalar palitrasidan, komponentalardan foydalanishni o'rgandim. Men ushbu kurs ishini bajarish davomida hisobot yartishni va bu hisobot QReport komponemtasini QuickRep xususiyatini ishlatishni o'rganib oldim Bu kurs ishini bajarish orqali ko'p bilim va ko'nikmalarga ega bo'ldim.

VI.Foydalanilgan adabiyotlar

1. Гради Буч. Объектно-ориентированный анализ и проектирование с примерами приложений на С++. Невский диалект. 2001.
2. И. Грехем. Объектно-ориентированные методы. Принципы и практика. Вильямс. 2004.
3. Г.С.Иванова. Объектно-ориентированное программирование. Учебник. МГТУ им. Баумана. 2003
4. Т. Фейсон. Объектно ориентированное программирование на С++ Киев. Диалектика. 1996.
5. WWW.Ziyonet.uz
6. www.google.uz
7. . Джесс Либерти. Освой самостоятельно С++ за 21 день.- СПб.:Питер.2003.815с.
8. . Трехтомник Берна Страуструпа
9. . Қораев Ф ва б. Программалашдан масалалар тўплами (методик қўлланма)-Тошкент -2008
- 10.. Назиров Ш. Объектга мулжалланган дастурлаш тиллари - Тошкент- 2006
11. Nazirov Sh."Obektga mo'ljallangan dasturlash tillari".Toshkent .2006.
12. www.infocom.uz
13. С++ dasturlash tilidan ma'ruzalar matni.

```

//-----

#include <vcl.h>
#pragma hdrstop
//-----
USEFORM("Unit1.cpp", frmAdmin);
USEFORM("Unit2.cpp", frmYordamchi);
USEFORM("Unit3.cpp", frmLugat);
USEFORM("Unit4.cpp", frmKirish);
USEFORM("Unit5.cpp", frmHisobot);
USEFORM("Unit6.cpp", frmDasturchi);
USEFORM("Unit7.cpp", Form7);
USEFORM("Unit8.cpp", Form8);
USEFORM("Unit9.cpp", Form9);
//-----
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
    Try
    {
        Application->Initialize();
        Application->Title = "Inlizcha so'zlarni oson yodlas!";
        Application->CreateForm(__classid(TfrmKirish), &frmKirish);
        Application->CreateForm(__classid(TfrmAdmin), &frmAdmin);
        Application->CreateForm(__classid(TfrmYordamchi), &frmYordamchi);
        Application->CreateForm(__classid(TfrmLugat), &frmLugat);
        Application->CreateForm(__classid(TfrmHisobot), &frmHisobot);
        Application->CreateForm(__classid(TfrmDasturchi), &frmDasturchi);
        Application->CreateForm(__classid(TForm7), &Form7);
        Application->CreateForm(__classid(TForm8), &Form8);
        Application->CreateForm(__classid(TForm9), &Form9);
        Application->Run();
    }
    catch (Exception &exception)
    {
        Application->ShowException(&exception);
    }
}

```

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