

**O'ZBEKISTON RESPUBLIKASI ALOQA
AXBOROTLASHTIRISH VA TELEKOMMUNIKATSIYA
DAVLAT QO'MITASI
TOSHKENT AXBOROT TEXNOLOGIYALARI
UNIVERSITETI FARG'ONA FILIALI**

”Axborot texnologiyalari” kafedrası

”C++ da dasturlash” fanidan

KURS ISHI

Bajardi: 610-13 guruhi talabasi
Bo'stonboyev Doston

Qabul qildi: Bazarboyev M.
Hoshimov A.

Farg'ona - 2015

Mavzu: C++ da Media fayllarni namoyish etish uchun dastur yaratish

Reja:

- I. Kirish
- II. Nazariy qism
 - 2.1. C++ tarixi
 - 2.2 C++ tili va ob'ektlarga mo'ljallangan dasturlash.
 - 2.3 C++ Builder asoslari
- III. Amaliy qism
 - 3.1 Dastur algoritmini tuzish va C++ ga bo'glash
 - 3.2 Dasturda ishlatilgan asosiy kodlar
- IV. Tadbiq qilish qismi
- V. Xulosa
- VI. Foydalanilgan adabiyotlar
- VII. Ilova

I. KIRISH

Prezidentimiz Islom Karimovning 2013-yil 20-sentabrda qabul qilingan “Mamlakatimizning dasturiy ta’minot vositalari ishlab chiquvchilarini rag‘batlantirishni yanada kuchaytirish chora-tadbirlari to‘g‘risida”gi qarorida dasturiy ta’minot vositalarini respublikamizda ishlab chiqarishni yanada rivojlantirish, ularni ichki va tashqi bozorlarda taklif etishga ko‘maklashishga doir muhim vazifalar belgilangan.

O‘zbekiston Respublikasi Aloqa, axborotlashtirish va telekommunikatsiya texnologiyalari davlat qo‘mitasining bo‘lim boshlig‘i Xurshid Komilov O‘zbekiston Milliy axborot agentligi – O‘zA muxbiriga davlatimiz rahbari imzolagan qarorda belgilangan vazifalar va ularning ijrosi haqida so‘zlab berdi.

– Prezidentimizning qaroriga muvofiq, Dasturiy ta’minot vositalari ishlab chiquvchilari milliy reyestrining tashkil etilishi muhim yangiliklardan biri bo‘ldi, – deydi X.Komilov. – Dasturiy ta’minot vositalari ishlab chiquvchilarining milliy reyestri software.uz internet-portalida joylashtirilgan dasturiy mahsulotlar va ishlab chiquvchilarning milliy katalogi negizida yaratiladi. Mamlakatimizning dasturiy ta’minot vositalari ishlab chiquvchilari to‘g‘risidagi axborotning yagona elektron bankini yaratish, milliy dasturiy mahsulotlarning ma’lumotlar bazasini shakllantirish hamda mamlakatimizda dasturiy mahsulotlar ishlab chiqarish bozorining holati va salohiyati to‘g‘risida axborot berish Dasturiy ta’minot vositalari ishlab chiquvchilari milliy reyestrini yuritishning asosiy vazifa va yo‘nalishlari hisoblanadi. Milliy reyestrda o‘z dasturiy ta’minot vositalarini ishlab chiqish, tatbiq etish va sotishdan olingan daromadlari ular sotgan tovarlar (ishlar, xizmatlar) umumiy hajmining kamida 50 foizini tashkil etadigan yuridik shaxslar kiritiladi. Ma’lumki, mehnatga haq to‘lash va u bilan bog‘liq sarf-xarajatlar dasturiy ta’minot ishlab chiquvchi korxonalar xarajatining asosiy qismini tashkil etadi. Davlatimiz rahbarining 2006-yil 5-yanvarda qabul qilingan “Yirik sanoat korxonalari bilan kasanachilikni rivojlantirish asosidagi ishlab chiqarish va xizmatlar o‘rtasida kooperatsiyani kengaytirishni rag‘batlantirish chora-tadbirlari

to'g'risida"gi farmoniga amal qilish xonadonlarda ishlovchi (kasanachi) jismoniy shaxslar bilan mehnat shartnomalari tuzadigan dasturiy ta'minot vositalari ishlab chiquv--chilar – yuridik shaxslarga tatbiq etish ko'zda tutilgan. Ya'ni, kasanachilarga buyurtma beradigan dasturiy ta'minot ishlab chiquvchilar mehnat shartnomasi asosida kasanachilarga to'lanadigan mablag'ga teng miqdorda mehnatga haq to'lash fondidan yagona ijtimoiy to'lovni to'lashdan ozod etiladi. Ayni paytda yagona ijtimoiy to'lov stavkasi 25 foizni tashkil etadi va bu mablag'ni tejash hamda iqtisod qilingan mablag'ni korxonaning moddiy-texnik bazasini rivojlantirish va mustahkamlash, shuningdek, xodimlarni rag'batlantirishga yo'naltirish imkonini beradi. Shu bilan birga, korxonalar sarf-xarajatlarini kamaytirish maqsadida davlatimiz rahbarining mazkur qarorida dasturiy ta'minot vositalarini ishlab chiquvchilarni 2017-yilning 1-yanvarigacha barcha turdagi soliqlar hamda O'zbekiston Respublikasi Moliya vazirligi huzuridagi Respublika yo'l fondiga, byudjetdan tashqari Ta'lim va tibbiyot muassasalarini rekonstruksiya qilish, mukammal ta'mirlash va jihozlash jamg'armasiga majburiy ajratmalar to'lashdan ozod etish ko'zda tutilgan. Axborot tizimlarini ishlab chiqish va joriy etish uchun boshqa butlovchi buyumlar ham talab etilishi inobatga olinib, dasturiy ta'minot ishlab chiquvchilar 2017-yilning 1-yanvarigacha mamlakatimizda ishlab chiqarilmaydigan, O'zbekiston Respublikasi Vazirlar Mahkamasi tomonidan tasdiqlanadigan ro'yxatlar bo'yicha o'z ehtiyojlari uchun olib kelinadigan uskunalar, butlovchi qismlar, detallar, uzellar, texnologiya hujjatlari, dasturiy ta'minot vositalariga boj to'lovlari (bojxona rasmiylashtiruv yig'imlaridan tashqari) to'lashdan ozod etiladi. Shuni ta'kidlash joizki, mamlakatimiz dasturiy mahsulot ishlab chiquvchilarini rag'batlantirish maqsadida me'yoriy-huquqiy hujjatlar muntazam takomillashtirilayotir. Masalan, ilgari dasturiy mahsulot va ma'lumotlar bazasi uchun arizalarni ro'yxatdan o'tkazish mablag' miqdori eng kam ish haqining 2,4 barobarini tashkil etardi. Vazirlar Mahkamasining qaroriga muvofiq, bu stavka nol koeffitsiyent darajasiga kamaytirildi.

II. Nazariy qism

2.1 C++ dasturlash tili tarixi

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

Ob'ektga mo'ljallangan dasturlash

Ob'yektga mo'ljallangan yondashuv bir kunda o'ylab topilgan emas. Uning paydo bo'lishi dasturiy ta'minotning tabiiy rivojidagi navbatdagi pog'ona, xolos. Vaqt o'tishi bilan qanday uslublar ishlash uchun qulay, qaysinisi noqulayekanini aniqlash oson bo'lib bordi. OMY eng muvaffaqiyatli, vaqt sinovidan o'tgan uslublarni o'zida mujassam etadi. Dastlab dasturlash anchayin boshqotirma ixtiro bo'lib, u dasturchilarga dasturlarni kommutatsiya bloki orqali kompyuterning asosiy xotirasiga to'g'ridan-to'g'ri kiritish imkonini berdi. Dasturlar mashina tillarida ikkilik tasavvurda yozilar edi. Dasturlarni mashina tilida yozishda tez-tez xatolarga yo'l qo'yilar, kodni kuzatib borish amalda deyarli mumkin emas edi. Bundan tashqari, mashina kodlaridagi dastur tushunish uchun g'oyat murakkab edi. Vaqt o'tishi bilan kompyuterlar tobora kengroq qo'llana boshlandi hamda yuqoriroq darajadagi protsedura tillari paydo bo'ldi. Bularning dastlabkisi FORTRAN tili edi. Biroq obyektga mo'ljallangan yondashuv rivojiga asosiy ta'sir keyinroq paydo bo'lgan. Protsedura tillari dasturchiga axborotga ishlov berish dasturini pastroq darajadagi bir nechta protseduraga bo'lib tashlash imkonini beradi. Pastroq darajadagi bunday protseduralar dasturning umumiy tuzilmasini belgilab beradi. Ushbu protseduralarga izchil murojaatlar protseduralardan tashkil topgan dasturlarning bajarilishini boshqaradi. Dasturlashning bu yangi paradigmasi mashina tilida dasturlash paradigmasiga nisbatan ancha ilg'or bo'lib, unga tuzilmalashtirishning asosiy vositasi bo'lgan protseduralar qo'shilgan edi. Maydaroq funksiyalarni nafaqat tushunish, balki sozlash ham osonroq kechadi. Biroq boshqa tomondan protsedurali dasturlash

koddan takroran foydalanish imkonini cheklab qo'yadi. Har bir protsedura ma'lumotlarga kirish usullarini dasturlashi lozim bo'lganligi tufayli ma'lumotlar taqdimotining o'zgarishi dasturning ushbu kirish amalga oshirilayotgan barcha o'rinlarining o'zgarishiga olib kelar edi. Shunday qilib, hatto eng kichik to'g'rilash ham butun dasturda qator o'zgarishlar sodir bo'lishiga olib kelar edi. Modulli dasturlashda, masalan, Modula ikki kabi tilda protsedurali dasturlashda topilgan ayrim kamchiliklarni bartaraf etishga urinib ko'rildi. Modulli dasturlash dasturni bir necha tarkibiy bo'laklarga yoki boshqacha qilib aytganda modullarga bo'lib tashlaydi. Agar protsedurali dasturlash ma'lumotlar va protseduralarni bo'lib tashlasa, modulli dasturlash, undan farqli o'laroq ularni birlashtiradi. Modul ma'lumotlarning o'zidan hamda ma'lumotlarga ishlov beradigan protseduralardan iborat. Dasturning boshqa qismlariga moduldan foydalanish kerak bo'lib qolsa, ular modul interfeysiga murojaat etib qo'ya qoladi. Modullar barcha ichki axborotni dasturning boshqa qismlarida yashiradi. Obyektga mo'ljallangan dasturlash (OMD) modulli dasturlashdan keyingi mantiqiy pog'onani egallaydi, u modulga nasldan naslga o'tishni va polimorfizmni qo'shadi. Dasturchi OMD dan foydalanar ekan, dasturni bir qator oliy darajali obyektlarga bo'lish yo'li bilan tizimlashtiradi. Har bir obyekt hal qilinayotgan muammoning ma'lum bir tomonini modellashtiradi. OMD endilikda dasturni bajarish jarayonini boshqarish uchun dasturchi diqqatini protseduralarni ketma-ketlikda chaqirib olish ro'yxatini tuzib o'tirishga qaratmaydi. Buning o'rniga obyektlar o'zaro aloqada bo'ladi. OMY yordamida ishlab chiqilgan dastur hal qilinayotgan muammoning amaldagi modelibo'lib xizmat qiladi. Dasturga obyektlar atamalari bilan ta'rif berish dasturiy ta'minotni ishlab chiqishning eng tushunarli usulidir. Obyektlar hamma narsani obyekt nima qilayotgani nuqtayi nazaridan idrok etishga, ya'ni uning xatti-harakatlarini xayolan modellashtirishga majbur qiladi. Shu tufayli obyektga yondashishda u dasturning bajarilishi jarayonida qanday ishlatiladi degan nuqtayi nazardan biroz e'tiborni chalg'itishi mumkin. Shunday qilib, dasturni yozish jarayonida haqiqiy dunyoning tabiiy atamalaridan foydalanish mumkin. Dasturni alohida protseduralar va ma'lumotlar shaklida (kompyuter dunyosi atamalarida)

qurish o'rniga obyektlardan iborat dastur qurish mumkin. 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'ekтли 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. 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'lchamga ega. Butun son ko'rsatkichni saqlash uchun yetarli ekani hamma mashinalar uchun ham birdek to'g'ri kelavermaydi. Asosiy turga nisbatan const sifatini qo'llash mumkin. Bunda dastlabki turga xos bo'lgan xususiyatlarga ega turni olish imkonini beradi. Bunda faqat bitta istisno bor. U shundaki, const turidagi o'zgaruvchilarning qiymati initsiallashtirish (nomlashtirish) dan so'ng o'zgara olmaydi. Bitta tishli qo'shtirnoqqa olingan belgi (simvol) belgi konstantasi hisoblanadi. E'tibor bering, shunday yo'l bilan aniqlangan konstanta xotirani egallamaydi. Shunchaki, konstanta qiymati kerak bo'lib qolgan o'rinda u bevosita qo'llanishi mumkin bo'ladi. O'zgaruvchilar uchun nomlantirish shart bo'lmasa-da, biroq qat'iy tavsiya etiladi. Lokal o'zgaruvchini nomlantirmay turib kiritishga deyarli asos yo'q. Ushbu turlarning har qanday kombinatsiyasiga quyidagi arifmetik operatsiyalarni qo'llash mumkin: (unar va binar plus) (unar va binar minus) * (ko'paytirish) / (bo'lish). Shuningdek ularga quyidagi qiyoslash operatsiyalarini ham qo'llash mumkin:

== (teng)

!=(teng emas)

<(kamroq)

>(ko'proq)

<=(kamroq yoki teng)

>=(ko'proq yoki teng)

C++ tili ob'ektga mo'ljallangan dasturlash prinsiplarini qo'llab quvvatlaydi. Bu prinsiplar quyidagilardir:

- Inkapsulyasiya
- Merosxo'rlik
- Polimorfizm

Inkapsulyasiya.

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

Merosxo'rlik

C++ tili merosxo'rlikni himoya qiladi. Bu yangi berilganlar tipi (sinf), oldindan bo'lgan sinfni kengaytirishdan hosil bo'ladi. Bunda yangi sinf oldingi sinfning merosxo'ri deb ataladi.

Polimorfizm.

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

2.2 C++BUILDER ASOSLARI

Ishlab chiqishning integratsiyalashgan muhiti komponentalar palitrasini birlashtiradi. Shakllar muharriri, kod muharriri, obyektlar noziri, obyektlar xazinasini bular hammasi kod va zaxiralar ustidan to'liq nazoratni ta'minlovchi dasturiy ilovalarni tez ishlab chiqish instrumentlari. Komponentalar palitrasi ilovalarni qurishda taklif qilinadigan 100 dan ortiq takroran qo'llanadigan komponentalardan iborat. Shakllar muharriri dasturning foydalanuvchi bilan interfeysini yaratish uchun mo'ljallangan. Kod muharriri dastur matnini, xususan, voqealarga ishlov berish funksiyalarini yozish uchun mo'ljallangan. Obyektlar noziri qotib qolgan chigal dasturlash zaruratisiz obyektlar xususiyatlarini vizual o'rnatish imkonini beradi hamda shunday voqealarni o'z ichiga oladiki, bu voqealarni ularning paydo bo'lishiga nisbatan obyektlar reaksiyasi kodlari bilan bog'lash mumkin bo'ladi. Obyektlar Xazinasini ma'lumotlarning shakl va modullari kabi obyektlarga ega bo'lib, ular ishlab chiqishda muvaqqat sarflarni kamaytirish maqsadida ko'plab ilovalar bilan bo'linadi. C++ Builder ilovalarni qurishning vizual metodikasini Komponentalar palitrasidan kerakli boshqarish elementlarini tanlab olish vositasida joriy etadi. Har bir komponenta bilan ushbu komponenta turini va xulq-atvorini o'zgartiradigan xususiyatlar bog'liq bo'ladi. C++ Builder 32 razryadli takomillashtirilgan Vizual komponentalar kutubxonasi VCL (Visual Component Library) bilan birgalikda yetkazib beriladi. Bu kutubxona eng murakkab ilovalarni qurish uchun mo'ljallangan 100 dan ortiq takroran qo'llanadigan komponentalardan iborat. Kutubxonaning asosiy komponentalari Palitralar komponentalarining instrumental panelida berilgan. Komponentalar belgilari dasturingiz shakliga olib o'tiladi. Kutubxona Windows va Windows 95 operatsiya tizimlaridagi Foydalanuvchi Grafik Interfeysi standart interfeys obyektlarining to'liq inkapsulatsiyalanishini o'z ichiga oladi. Ular orasida, ixtisoslashgan komponentalar bilan bir qatorda, relyatsion ma'lumotlar bazasini boshqarish uchun mo'ljallangan komponentalar alohida o'rin egallaydi. Ishonchli va samarali dasturlarni yaratishda C++Builder obyektga mo'ljallangan dasturlash (OMD) imkoniyatlaridan to'liq foydalanadi.

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

Standart komponentalar

TMainVlenu

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

TPopUpMenu

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

TLabel

TLabel shaklda tahrir qilib bo'lmaydigan statik matnning to'rtburchak sohasini aks ettiradi. Odatda matn boshqa komponenta nomidan iborat bo'ladi. Nom matni Caption xususiyatining qiymatidir. Alignment xususiyati matnni tekislash usulini aniqlaydi. Shrift o'lchami avtomatik tarzda sohaning maksimal to'ldirilishiga mos kelishi uchun AutoSize xususiyatining true qiymatini o'rnatish. Kalta soha ichida matnning hammasini ko'rish imkoniga ega bo'lish uchun WordWrap xususiyatining true qiymatini bering. Transparent xususiyatining true qiymatini lo'rnatish, boshqa komponentaning bir qismini to'g'ri uning ustidajoylashtirilgan nom orasidan ko'rinib turadigan qilishingiz mumkin.

TEdit

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

TMemo

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

TButton

TButton yozuvli to'rtburchak tugmani yaratadi. Tugmacha bosil-ganda, dasturda biror bir xatti-harakat nomlanadi (initsiallashtiriladi). Tugmachalar ko'proq dialogli darchalarda qo'llanadi. Default xususiyatining true qiymati tomonidan tanlab olingan yashirin tugmacha, dialog darchasida har gal Enter klavishi bosilganda, OnClick voqea qayta ishlatgichini ishga tushiradi. Cancel

xususiyatining true qiymati tanlab olgan uzish tugmachasi, dialog darchasida har gal Escape klavishi bosilganda, OnClick voqea qayta ishlatgichini ishga tushiradi. TVutton komponentasi TButtonControl sinfining hosilasi hisoblanadi.

TCheckBox

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

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 OnClick voqeasi yuzaga keladi. Odatda radio-tugmalar avvaldan shaklda o'rnatilgan konteyner ichigajoylashtiriladi. Agar bitta tugma tanlangan bo'lsa, ushbu guruhga mansub barcha boshqa tugmalarning tanlovlari avtomatik tarzda olib tashlanadi.

TListBox

TListBox tanlash, qo'shish yoki o'chirish uchun mo'ljallangan matn variantlari ro'yxatining to'rtburchak sohasini aks ettiradi. Agar ro'yxatdagi barcha elementlar ajratilgan sohaga sig'masa, ro'yxatni aylantirish lineykasi yordamida ko'rib chiqish mumkin. Ro'yxat elementlari Items xususiyatining ichida, dastur

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

`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, 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

`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

TGroupBox to'g'ri burchakli ramka ko'rinishidagi konteyner bo'lib, u qandaydir bir interfeys elementlarining mantiqan bog'langan guruhini shaklda vizual birlashtiradi. Bu komponenta Windowsning bir nomdagi obyektning inkapsulatsiyalanishidan iborat. TRadioGroup TRadioGroup to'g'ri burchakli ramka ko'rinishidagi konteyner bo'lib, u bir-birini mantiqan istisno qiladigan radio-tugmalar guruhini shaklda vizual birlashtiradi. Radio-tugmalar bitta konteynerga joylashtirilganda «guruhlanadi». Bu guruhdan faqat bitta tugmacha tanlab olinishi mumkin. TRadioGroup komponentasiga tugmalarni qo'shish uchun Items xususiyatining 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, siz radiotugmalarni bir necha ustunga guruhlashingiz mumkin.

TPanel

TPanel boshqa komponentalarni o'z ichiga olishi mumkin bo'lgan bo'sh panelni yaratadi. Siz TPanel dan o'z shaklingizda instrumentlar paneli yoki holatlar satrlarini yaratish uchun foydalanishingiz mumkin. TPanel panel komponentasi TCustomPanel sinfining hosilasi bo'lib, uning barcha xususiyatlari, metodlari va voqealari to'liq vorislik qiladi. Windows komponentalari Windows komponentalari sizning dasturingizga Windows ning 12 ta interfeys elementlarining ulanishini amalga oshiradi.

TTabControl

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

TPageControl

TPageControl ko'p varaqli dialogni tashkil etish uchun mo'ljallangan qisman bir-birining ustini yopuvchi qo'shimcha kartoteka ilovalari ko'rinishidagi

maydonlar to'plamini aks ettiradi. Tegishli qo'shimcha ilovaga ega bo'lgan yangi dialogli sahifani yaratish uchun berilgan komponentaning kontekst menusidan New Page opsiyasini tanlab oling. Siz konkret sahifani quyidagi 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.

TTabSheet

qurilma boshqarish komponentasi amalga oshiradi. Agar hamma qo'shimcha ilovalar bir qatorga sig'masa, komponenta aylan- tirish tugmalarini chiqaradi. Qo'shimcha ilovalarni bir necha qatordaaks ettirish uchun MultiLine xususiyatining true qiymatini bering.

TTreeView

TTreeView elementlar tartibi tabaqaviy (shajaraviy) ko'rinishga ega bo'lgan maydonni aks ettiradi: hujjatlar sarlavhasi, ko'rsatkichdagi yozuvlar, disklardagi fayllar yoki kataloglar. Bu komponentaning ishini ko'plab Windows ilovalarida ko'rish mumkin. Items xususiyati shajara elementlarining tahrir qilinayotganro'yxatiga ega bo'lgan TTreeNode obyektiga iqtibos (murojaat) qiladi.

TListView

TListView tabaqaviy (shajaraviy) elementlar ro'yxatiga ega bo'lgan maydonni aks ettiradi. ViewStyle xususiyati ro'yxat elementlarini quyidagi tartibda aks etishini belgilaydi: sarlavhali ustunlar bo'yicha, vertikal, gorizontal, katta yoki kichik piktogrammalar bilan. Items xususiyati ost yozuvlarini qo'shish, o'chirish va modifikatsiyalash, shuningdek, ro'yxat elementlariga piktogrammalarni tanlash imkonini beradi. Ro'yxat muharriri ushbu xususiyat qiymatlari grafasidan tugmacha yordamida chaqiriladi.

TImageList

TImageList bir xil o'lchamdagi grafik tasvirlardan iborat kolleksiya uchun konteyner yaratadi. Bu tasvirlarning har bittasini ularning 0 dan n-1 gacha qiymatlar intervalidagi indeks bo'yicha tanlab olish mumkin. Grafik kolleksiyalar bitli obrazlar yoki piktogrammalarning katta to'plamlariga samarali xizmat ko'rsatish uchun qo'llanadi. Ushbu bitli obrazlar va piktogrammalar eni k*n bo'lgan yagona bitli obraz sifatida saqlanadi. Niqobli monoxrom 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

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

TStatusBar

TStatusBar dastur ishlayotgan paytda chiqarib beriladigan maqomli (statusli) axborotni aks ettirish uchun holatlar panellari satrini (bu satr odatda shaklning pastki chegarasi bo'ylab tekislanadi) yaratadi. Har bir panel Panels xususiyatlari ro'yxatida o'z o'rniga ega. Panellar 0 indeksidan boshlab, chapdan o'ngga qarab 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

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

TProgressBar

TProgressBar sizning dasturingizdagi qandaydir protseduraning bajarilish jarayonini kuzatib boradi. Protsedura bajarilgan sayin, to'g'ri burchakli indikator chapdan o'ngga qarab asta-sekin berilgan rangga bo'yalib boradi. Min va Max xususiyatlari indikator qiymatlari indikatorini o'rnatadi. Step xususiyati indikator pozitsiyasi o'zgarishi bilan har gal Position xususiyati qiymatining o'zgarish qadamini belgilaydi.

TBitBtn

TBitBtn bit obrazining tasviri tushirilgan tugmachani 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

TSpeedButton odatda ma'lum menu buyruqlarini tez chaqirish yoki rejimlarni o'rnatish paneli (TPanel) da joylashtiriladigan grafik tugmani yaratadi. Tezkor tugmaning turli holatlariga (masalan, «bosilgan», «qo'yib yuborilgan», «taqiqlangan» va h.k.) turli bit obrazlari mos kelishi mumkin. Bir-birining o'rnini bosadigan tasvirlar va yozuv matnini tanlash uchun xususiyatlar mavjud. Kengayishli tasvirlar fayllari muharririning darchasi Glyph xususiyati qiymatlarining grafasidagi tugma bilan ochiladi. Tez tugmalarning boshqa xususiyatlari ularning biror bir guruhdagi ishini tashkil etadi.

TMaskEdit

TMaskEdit o'ziga xos formatdagi ma'lumotlarning tahrir qilinadigan nazoratdagi to'rtburchak sohasini yaratadi. Kiritilayotgan matnning to'g'riligi ruxsat etilgan formatlarni kodlovchi niqob vositasida tekshiriladi. Bu formatlarga matn kiritilgan va foydalanuvchiga taqdim etilgan bo'lishi mumkin (sana, vaqt, telefon raqami va h.k.). EditMask xususiyati joriy niqob kodini saqlaydi. Niqoblar muharriri darchasi ushbu xususiyat qiymatlari grafasida tugma bilan ochiladi. TMaskEdit komponentasi TCustomMaskEdit sinfining to'g'ridanto'g'ri hosilasidir. U satrlar yoki ustunlar bo'yicha belgili ketmaketliklarni aks ettirish uchun mo'ljallangan muntazam (regulyar) to'rni yaratadi.

TStringGrid

TStringGrid komponentaga tegishli barcha xususiyatlarning nomlari va vazifalari bo'lib, siz ulardan dasturni loyihalash bosqichida to'la foydalanishingiz mumkin. Ular keyingi paragrafda tavsifi berilgan TDrawGrid komponentasi xususiyatlariga to'liq to'g'ri keladi. Simvolli ketma-ketliklar bilan bog'liq barcha obyektlar kerakli obyektga murojaat qilish imkonini beradigan Objects xususiyatida mujassam bo'lgan. Dastur bajarilish paytida simvolli ketma-ketliklar va setka ustunining ular bilan bog'liq obyektlari Cols xususiyati bilan adreslanadi. Rows xususiyati setka satrlari bilan xuddi shunday ish tutish imkonini beradi. Setkaning barcha simvolli ketma-ketliklari setkaning kerakli uyasini adreslaydigan (manzillaydigan) Cells xususiyatida mujassamdirlar.

TDrawGrid

TDrawGrid tuzilma holiga keltirilgan grafik ma'lumotlarni satrlar yoki ustunlar bo'yicha aks ettirish uchun muntazam setka yaratadi. RowCount va ColCount xususiyatlari vertikal va gorizontal bo'yicha setka uyalarining sonini belgilaydi. Options xususiyatining qiymatlari setkaning turi (masalan, ustunlar orasida ajratuvchi chiziqlarga ega bo'lgan setka turi) va uning xulq-atvorini (masalan, ustundan ustunga Tab 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. FixedCol va FixedRows xususiyatlari ustunlar va satrlarning aylantirilishini taqiqlab qo'yish imkonini beradi, FixedColor xususiyati esa barcha ustun va satrlarga ma'lum rang beradi. DefaultDrawing xususiyatining true qiymati setka uyalarining ichidagilarini avtomatik tarzda chizib ko'rsatadi, bunda uning foni, asosi va rangi yashirin tanlanadi. DefaultDrawing xususiyatining falseqiymatini o'rnatish hamda setka uyalarini «qo'lda» to'ldirish uchunmo'ljallangan OnDrawCell voqeasi qayta ishlatgichining yozilishinitalab qiladi. DefaultColWidths va DefaultRowHeights xususiyatlari yordamida yashirin tanlanayotgan barcha ustunlar va satrlarning enini o'rnatish mumkin. ColWidth va RowHeight xususiyatlari konkretustun enini va konkret satr bo'yini spetsifikatsiyalaydi. Dastur ishlash paytida siz CellRect metodi yordamida biror bir uyaning rasmini chizish uchun ma'lum sohani o'z ixtiyoringizga olishingiz mumkin. MouseToCell metodi ustun raqami va sichqoncha kursori o'rnatilgan satr uyasining koordinatalarini qaytarib beradi. Setkaning tanlab olingan uyasi Selection xususiyatining qiymati bo'lib qoladi

TImage

TImage shaklda grafik tasvir konteynerini yaratadi (bu bit obrazi, piktogramma yoki meta fayl bo'lishi mumkin). Tasvirlar fayllari muharririning darchasi Picture xususiyati qiymatlari grafasidagi tugma bilan ochiladi. Konteyner o'z o'lchamlarini tasvirni to'liq sig'diradigan qilib o'zgartirishi uchun AutoSize xususiyatining true qiymatini o'rnatish. Kichikroq o'lchamdagi dastlabki tasvir

butun konteynerga choʻzilib ketishi uchun Stretch xususiyatining true qiymatini oʻrnating. 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>»);

TShape

TShape aylana va ellips, kvadrat va toʻgʻri toʻrtburchak (bur-chaklarini yumaloqlash mumkin) kabi oddiy geometrik shakllarning rasmini chizadi. Tanlab olingan geometrik shaklning turini Shape xususiyati, rang va boʻyash usulini Brush komponentasiga joylangan ikkita Color va Style xususiyatlari aniqlaydi. Shakllarning oʻlchamlarini ham tegishli xususiyatlar aniqlaydi.

TBevel

TBevel xuddi iskana bilan oʻyilgandek hajmli koʻrinadigan chiziqlar, bokslar yoki ramkalarni yaratadi. Komponenta chizayotgan obyektни Shape xususiyati aniqlaydi, Style xususiyatining qiymati esa obyekt koʻrinishini oʻzgartirib, uni boʻrtiq yoki botiq holga keltiradi. Foydalanuvchi shakl oʻlchamlarini oʻzgartirganda ham obyektning nisbiy holatini oʻzgarmas qoldirish uchun Align xususiyatining true qiymatini oʻrnating.

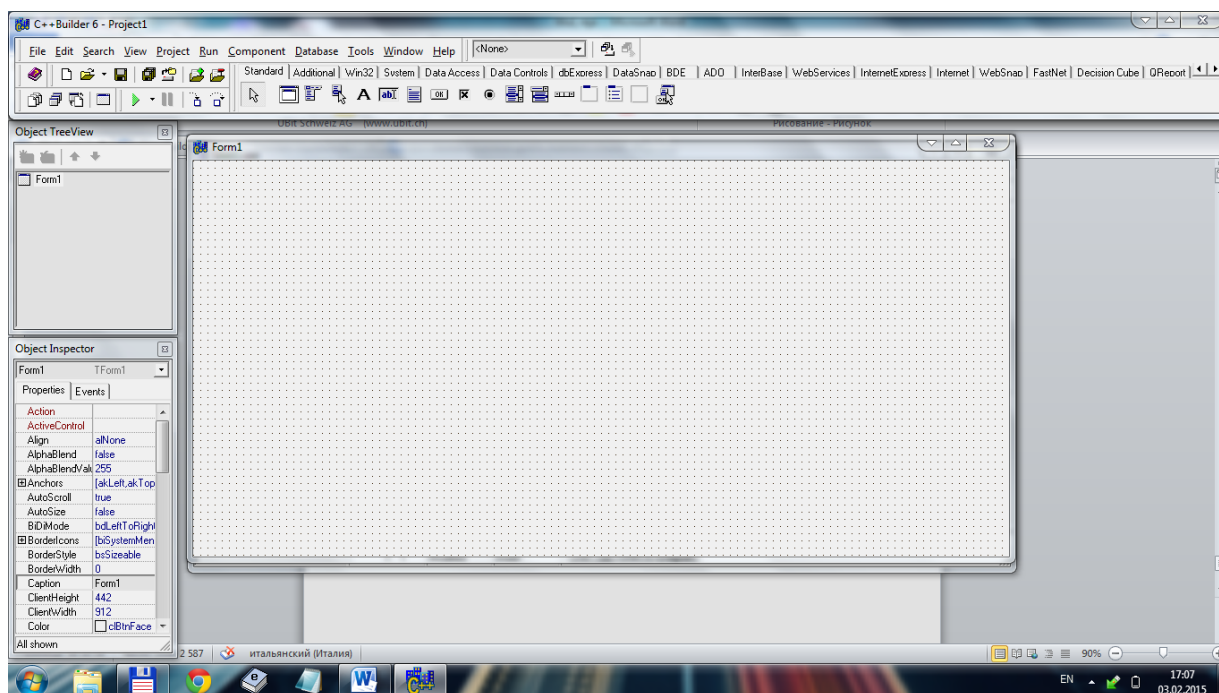
TScrollBar

TScrollBar darchada oʻlchamlari oʻzgaruvchan boksni yaratadi, bu boks shu topdayoq avtomatik tarzda zaruratga koʻra aylantirish lineykalari bilan taʼminlanadi. Aylantirib koʻrish boksi yordamida darchaning ayrim sohalarini aylantirib koʻrishdan himoyalash mumkin. Masalan, instrumentlar paneli va holat panelini himoyalash uchun avval darchani aylantirish lineykasini berkitib qoʻying, keyin esa aylantirish boksini mijoz sohasida instrumentlar paneli va holat paneli oʻrtasida joylashtiring. Boksni aylantirib koʻrish lineykasi darchaga tegishli boʻlib koʻrinadi, biroq aylantirish faqat boks ichida amalga oshiriladi. Aylantirib koʻrish bokslaridan yana boshqachasiga ham foydalanish mumkin: ular biror bir darchada koʻplab aylantirib koʻrilayotgan sohalar (turlar) yaratish imkonini beradi.

III. Amaliy qism.

3.1 Dastur algoritmini tuzish va C++ ga bo'glash

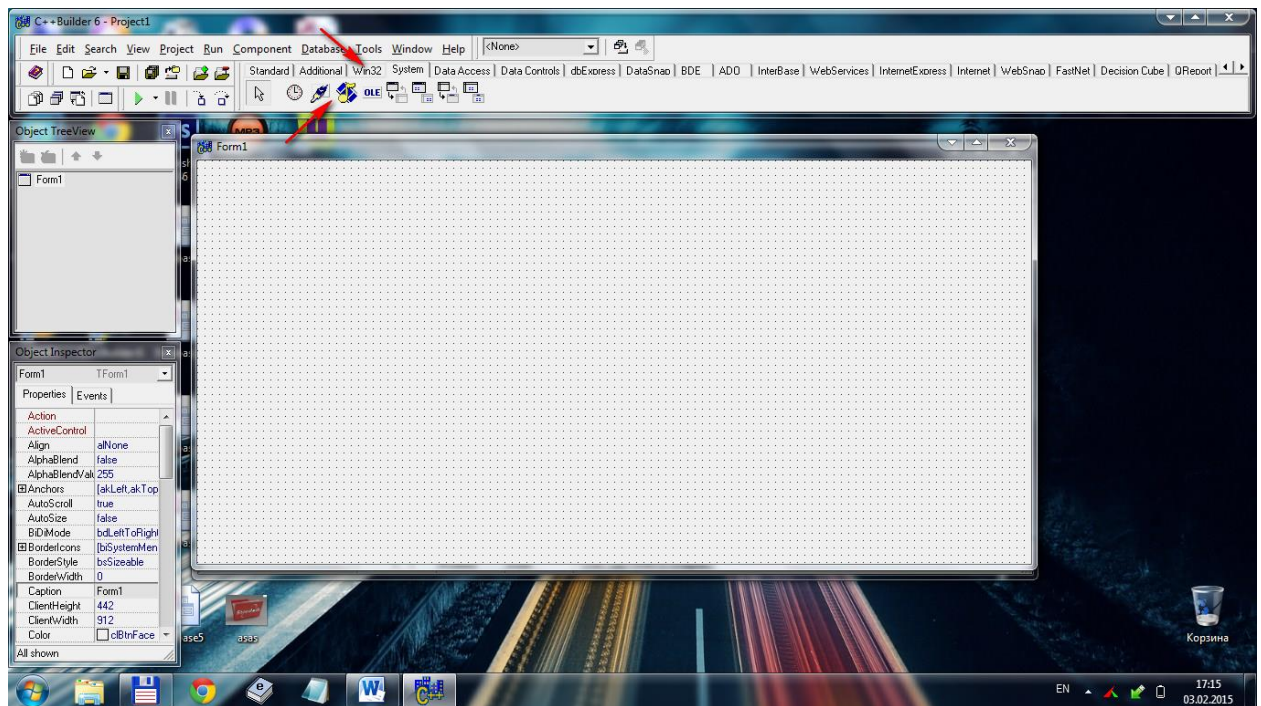
Biz avvalambor C++ Builder 6 dasturi ishga tushiramiz va File menyusidan New bo'limiga o'tib yangi application yaratamiz.



3.1 rasm. C++ builder formaning umumiy korinishi

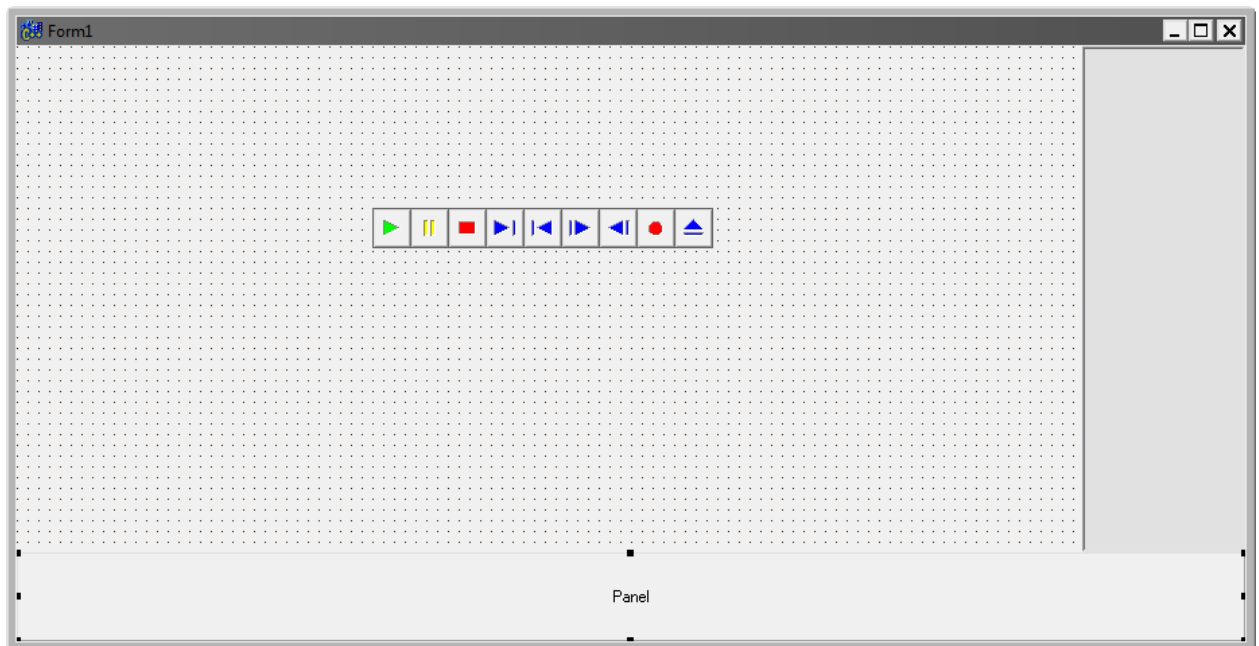
Endi biz dasturni kerakli joyga saqlab qo'yamiz, va dasturga kerakli komponentalarni joylashtiramiz.

Avvalombor biz, komponentalar palitrasidan Systemni tanlaymiz va u yerdan MediaPlayer komponentasini tanlab, dasturga joylashtiramiz.



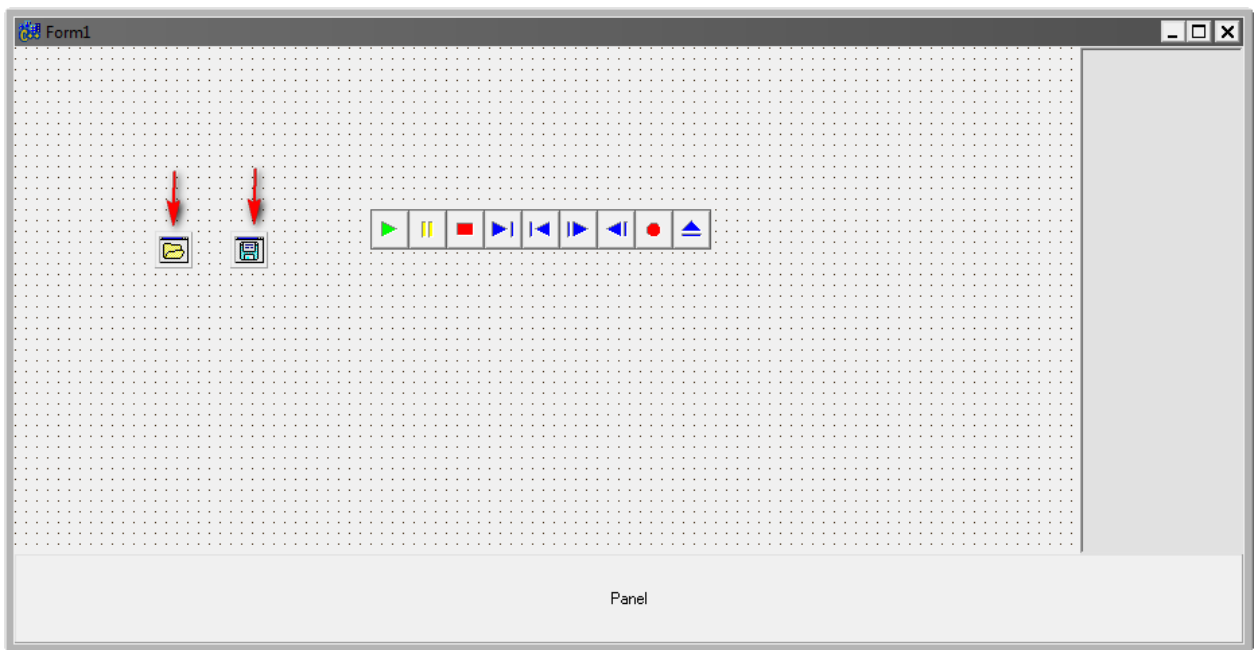
3.2-rasm. Win 32 dagi MediaPlayer komponentasi

Dasturni dizayn qismini ishlash uchun bir nechta panel va Listbox tashlaymiz.



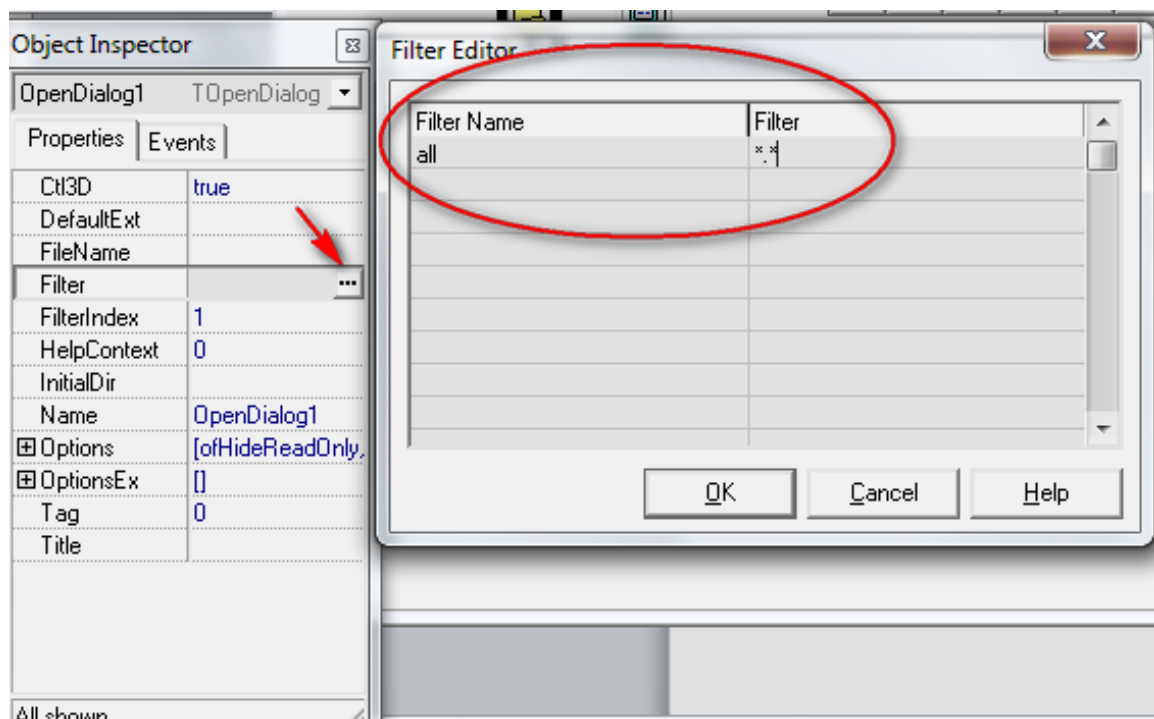
3.3-rasm. Formaga ListBox va Panel tashlangan holat

Shundang so'ng, komponentalar politasidan Dialogsga kirib OpenFileDialog va SaveDialog komponentalarini dasturga tashlaymiz.



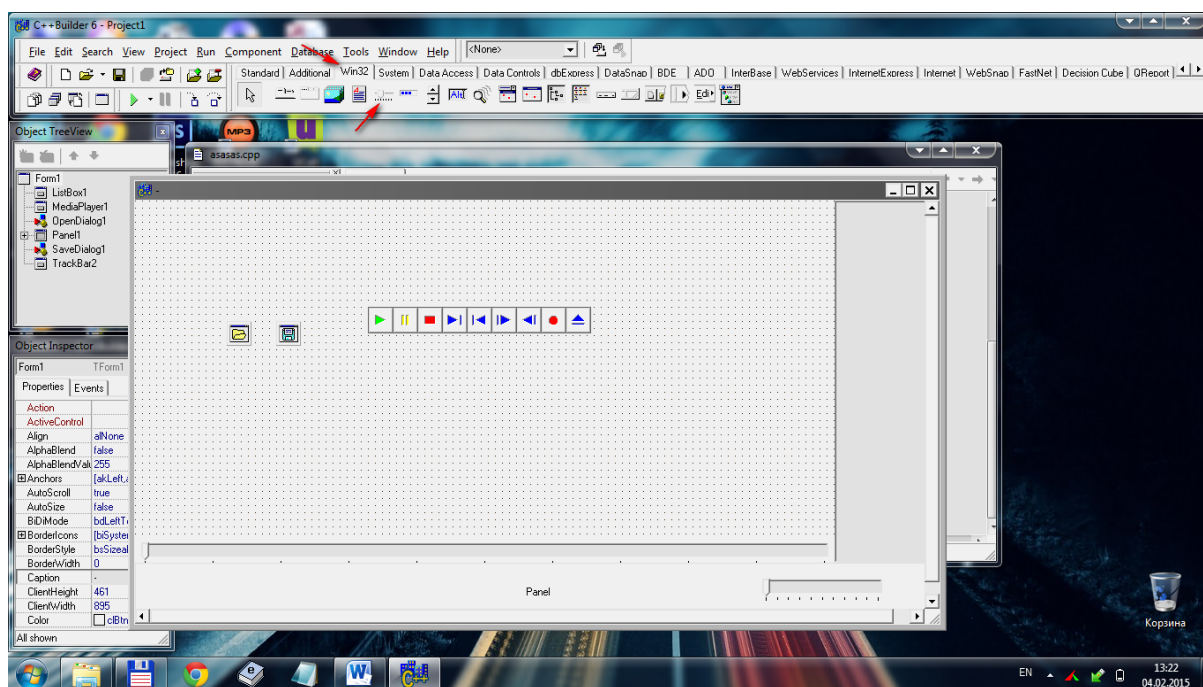
3.3-rasm. Formaga SaveDialog va OpenFileDialog tashlangan holat

OpenDialogni va SaveDialogni Filter xususiyatini sozlaymiz. Buning uchun biz OpenFileDialogni Filteriga Filter Name = All ni Filter = *.* ni qo'shamiz



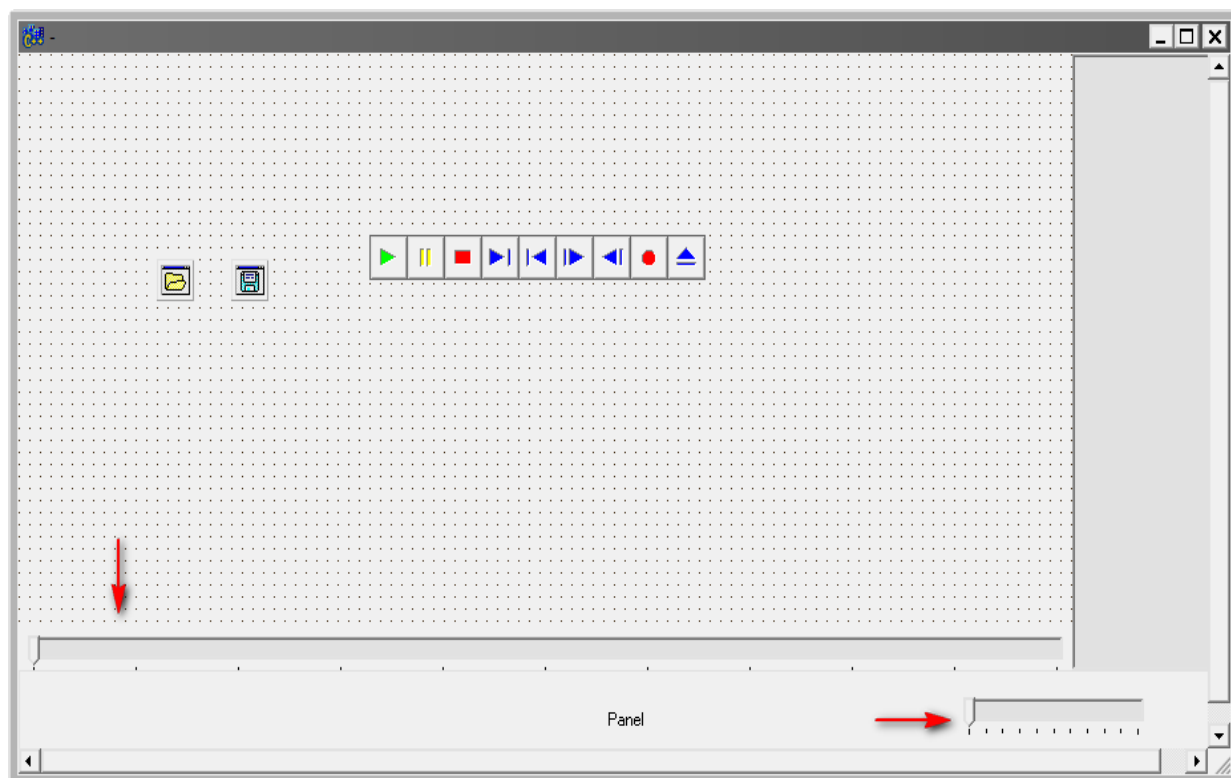
3.4-rasm. OpenFileDialogning Filter xususiyatini sozlash

Biz MediaPlayer dasturimizga kerak bo'ladigan ovoz sozlash va position yurgichlarni o'rnatamiz. Buning uchun bizga Win32 komponentalar politrasiidagi Trackbar ni formamizga tashlaymiz.



3.5-rasm. Win 32 palitrasidagi TrackBar komponentasi

Shundang so'ng formamizda tashlangan Trackbarlarni kerakli ko'rinishga keltiramiz. Birinchi trackbar pozitsiya uchun ikkinchisi ovoz uchun kerak bo'ladi.



3.6-rasm. TrackBarlarni sozlash

Ovoz uchun atalagan Trackbar quyidagi kodni kiritamiz.

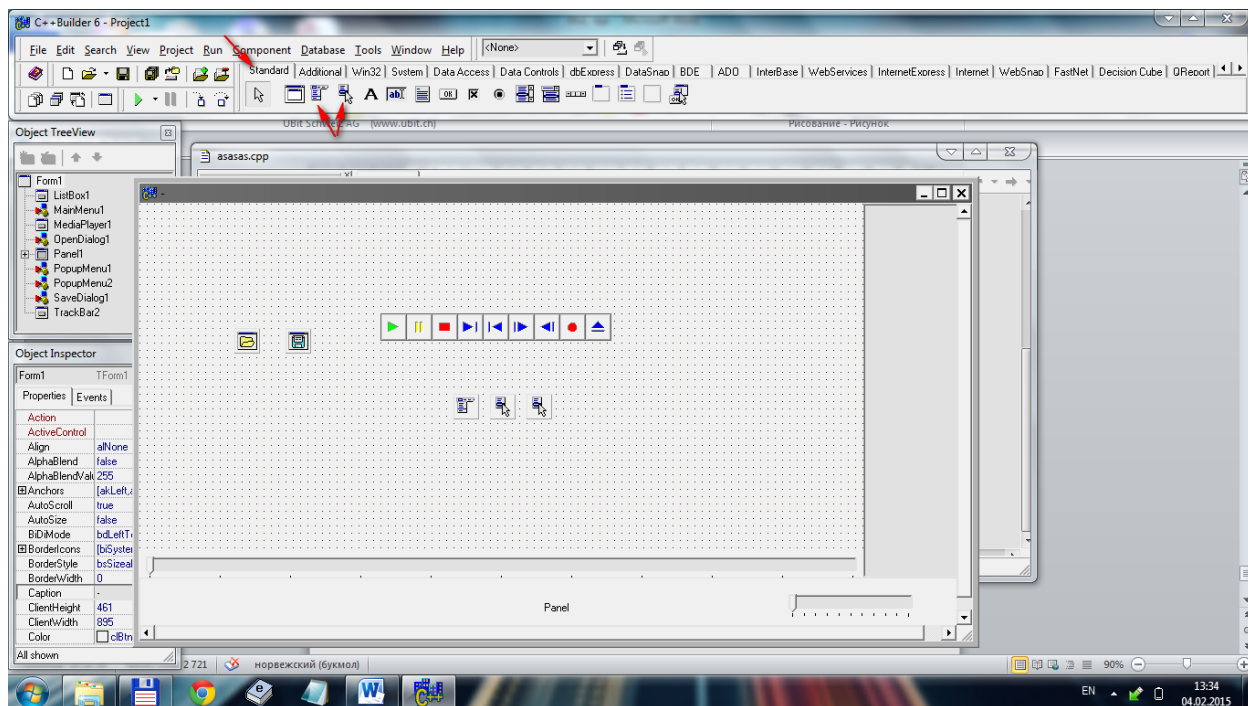
```
void __fastcall TForm1::TrackBar1Change(TObject *Sender)
{
    TrackBar1->Max=65350;
    waveOutSetVolume(0,MAKELONG(TrackBar1->Position, TrackBar1-
    >Position));
}
```

Pozitsiya uchun atalgan Trackbar uchun quyidagicha :

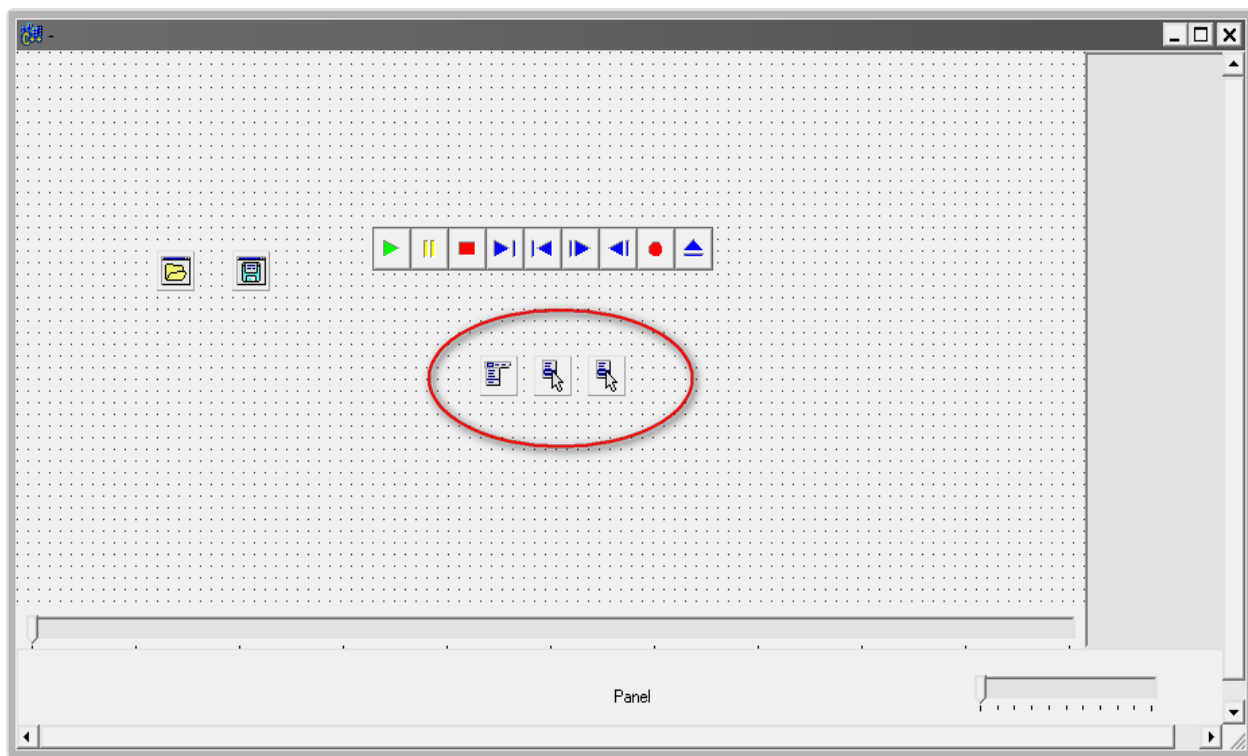
```
void __fastcall TForm1::ScrollBar1Change(TObject *Sender)
{
    ScrollBar1->Position=MediaPlayer1->Position;
}

void __fastcall TForm1::ScrollBar1Scroll(TObject *Sender,
    TScrollCode ScrollCode, int &ScrollPos)
{
    MediaPlayer1->Position=ScrollBar1->Position;
    MediaPlayer1->Play();
}
```

Formamizga Standart komponentlar politrasiidan TMainMenu va TPopupMenu larni tashlaymiz. TmainMenu dasturimizning asosiy menyusi hisoblanadi. Qolgan menyular esa yordamchi menyu sifatida ishlatiladi.

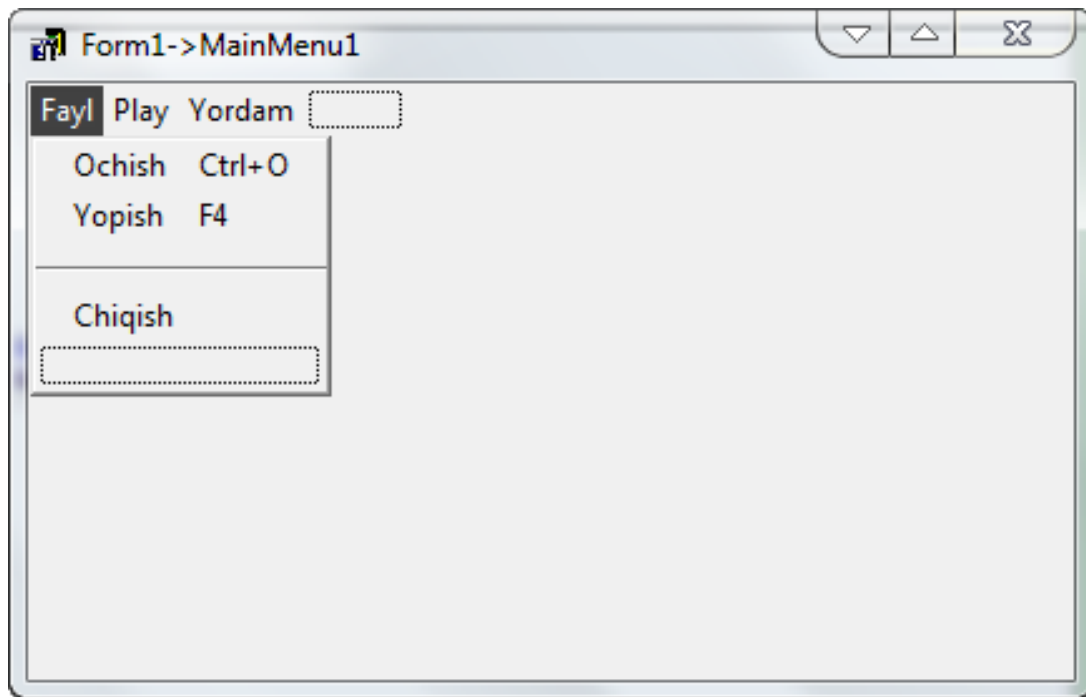


3.7-rasm. Menyularning Palitradagi ko'rinishlari



3.8-rasm. Formadagi menyular joylashuvi

TmainMenu menyusiga kerakli bo'limlarni kiritamiz va har bir bo'limga kodlar yozamiz.



3.9-rasm. Asosiy menyuning ko'rinishi

```
void __fastcall TForm1::Ochish1Click(TObject *Sender)
{
    if(OpenDialog1->Execute())
    {
        MediaPlayer1->FileName=OpenDialog1->FileName;
        MediaPlayer1->Open();
        if(MediaPlayer1->Enabled==false)
        {
            Timer1->Enabled=true;
        }
        ScrollBar1->Max=MediaPlayer1->Length;
        MediaPlayer1->Play();
        ListBox1->Items->Add(ExtractFileName(OpenDialog1->FileName));
    }
}

void __fastcall TForm1::Yopish1Click(TObject *Sender)
{
    MediaPlayer1->Close();
}
```

```

}

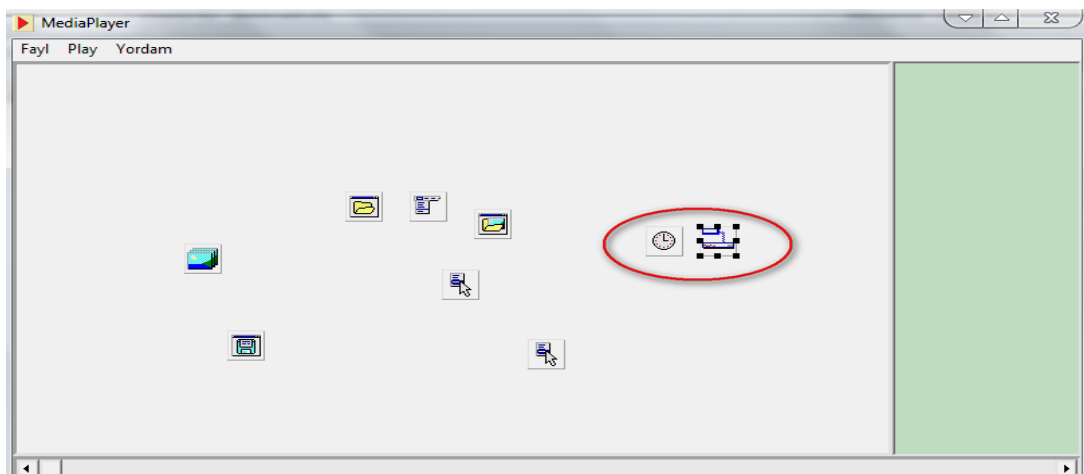
void __fastcall TForm1::Chiqish1Click(TObject *Sender)
{
    Form1->Close();
}

void __fastcall TForm1::Qollanma1Click(TObject *Sender)
{
    Form3->ShowModal();
}

void __fastcall TForm1::Dasturhaqida1Click(TObject *Sender)
{
    Form2->ShowModal();
}

```

Asosiy menyumizni yaratib bo'lganimizdan keyin formamizga Timer va TrayIcon komponentalarini tashlaymiz va Timerimizga kerakli kodlarni kiritamiz.



3.10-rasm. TrayIcon va Timerlarning formdag ijoylashivu

Timerimiz joriy faylni o'tayotgan vaqtini va fayl positsiyasini ko'rsatishi uchun quyidagi kodlarni Timerning ichiga kiritamiz.

```

void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
    ScrollBar1->Position=MediaPlayer1->Position;
}

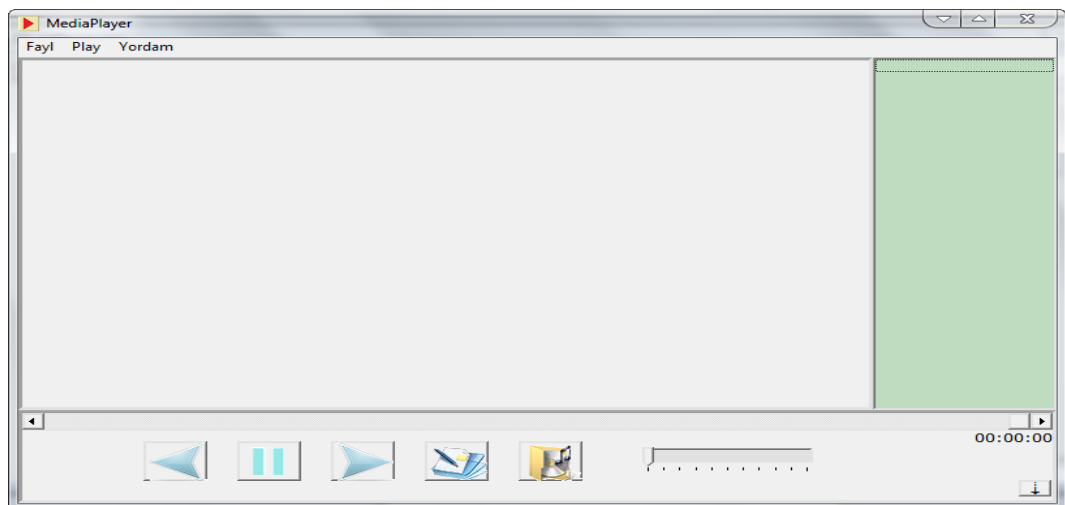
```

```

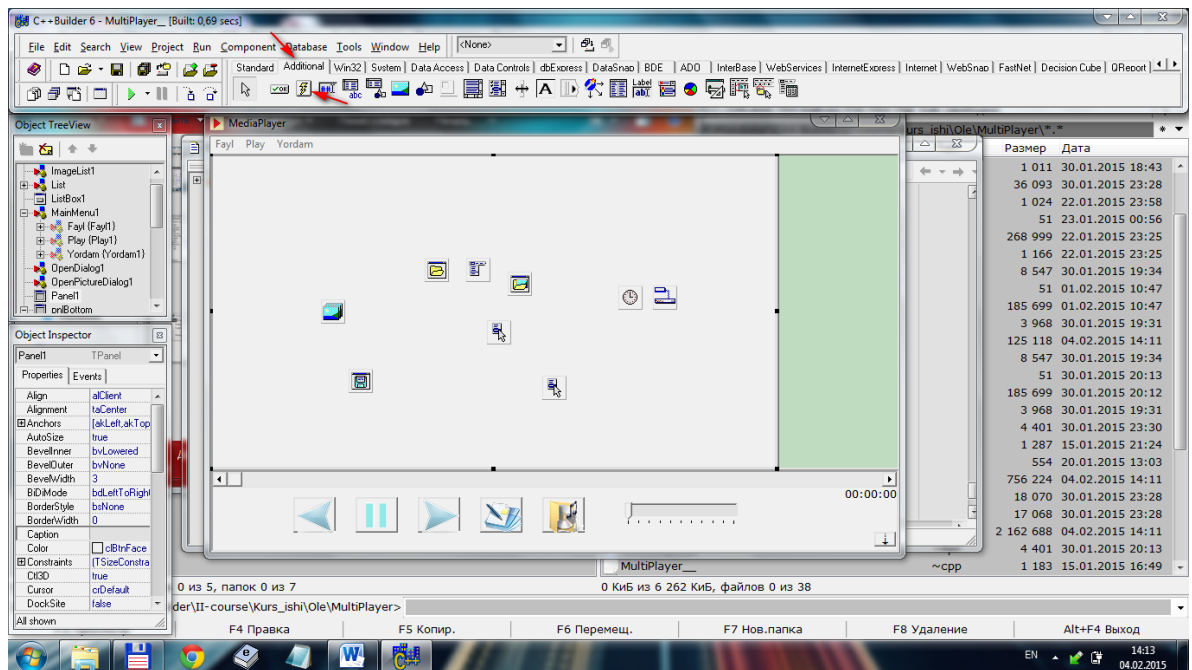
if(MediaPlayer1->Mode==mpPlaying)
{ long time = MediaPlayer1->Position;
  long mm = time / 60000;
  time -= mm * 60000;
  long ss = time / 1000;
  long ms10 = (time - ss*1000)/10;
  char buff[20];
  sprintf( buff, "%0.2d:%0.2d.%0.2d", mm, ss, ms10 );
  Label1->Caption = buff;
}
}

```

Fayllarni boshqarish uchun formamizga bir nechta SpeedButton larni tashlaymiz va ularga rasm qo'yamiz, panellar rangini ham kerakli rangga o'zgartiramiz.

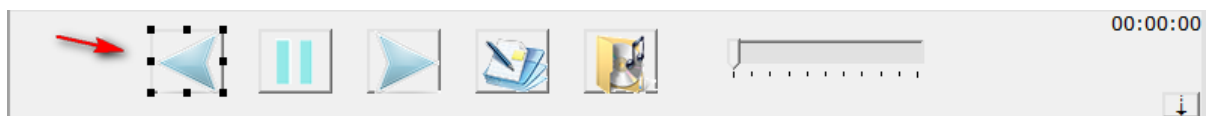


3.11-rasm. Formadagi asosiy ko'rinishi



3.12-rasm. Additional palitrasidagi SpeedButton tugmasini ko'rinishi

Formamizga tashlangan SpeedButtonlarni kerakli kodlarni kiritamiz. Birinchi navbatda 'oldingi' ma'nosini anglatuvchi tugmamizga kod yozamiz.



3.13-rasm. Oldingi tugmasini ko'rinishi

```
void __fastcall TForm1::oldingiClick(TObject *Sender)
{
    int oldingi = ListBox1->ItemIndex;
    if(oldingi==0)
    {
        oldingi=ListBox1->Count;
    }
    oldingi--;
    MediaPlayer1-> FileName=ListBox1-> Items-> Strings[oldingi] ;
    MediaPlayer1->Open();
}
```

```
MediaPlayer1->Play();
ListBox1->ItemIndex=oldingi; }
```



3.14-rasm. Pausa tugmasini umumiy ko'rinishi

```
void __fastcall TForm1::boshlaClick(TObject *Sender)
{
MediaPlayer1->Pause();
}
```



3.15-rasm. Keyingi tugamaning formadagi ko'rinishi

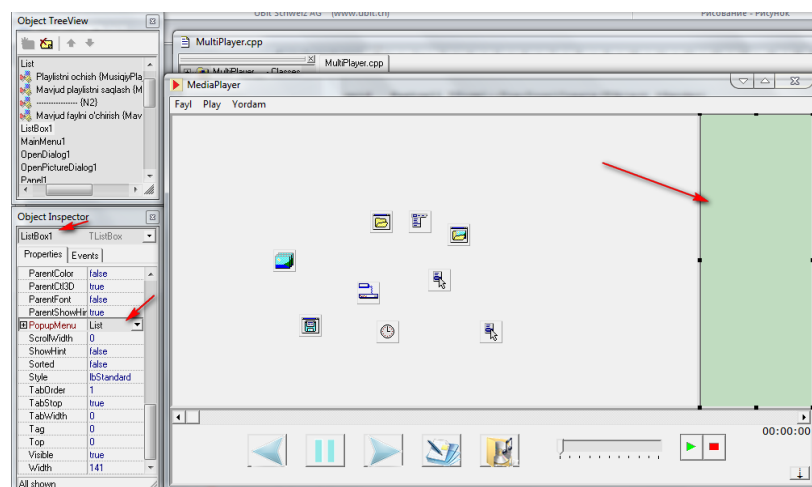
```
void __fastcall TForm1::keyingiClick(TObject *Sender)
{
Timer1->Enabled=True;
int keyingi = ListBox1->ItemIndex;
if(keyingi==ListBox1->Count-1)
{
keyingi=-1;
}
keyingi++;
MediaPlayer1-> FileName=ListBox1-> Items-> Strings[keyingi] ;
MediaPlayer1->Open();
MediaPlayer1->Play();
ListBox1->ItemIndex=keyingi;
}
```



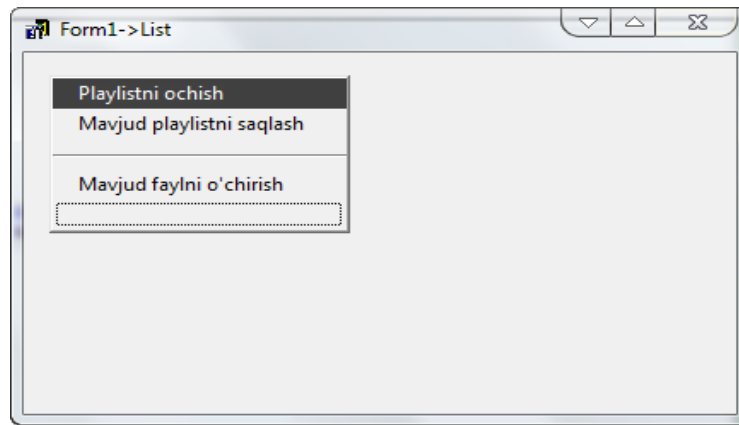

3.16-rasm. Ochish tugmasini formadagi ko'rinishi

```
void __fastcall TForm1::ochishClick(TObject *Sender)
{
    if(OpenDialog1->Execute())
    {
        OpenFileDialog1->Title="Medaifaylni tanlang";
        MediaPlayer1->FileName=OpenDialog1->FileName;
        MediaPlayer1->Open();
        Timer1->Enabled=true;
        ScrollBar1->Max=MediaPlayer1->Length;
        MediaPlayer1->Play();
        ListBox1->Items->Add(ExtractFileName(OpenDialog1->FileName));
    }
}
```

Dasturimizda tayyor playlistlar uchun maxsus menyu yaratamiz. Bu menyu joriy ro'yhatdagi qo'shiqlarni playlist qilib saqlash va ularni o'qish uchun xizmat qiladi. Bu playlistni Listboxga biriktirib qo'yamiz.



3.17-rasm. ListBox ga menyuni biriktirish holoti



3.18-rasm. ListBoxning shaxsiy menyusi

Maxsus playlist kodlari :

```
void __fastcall TForm1::MusiqiyPlaylist1Click(TObject *Sender)
{
    if(OpenDialog1->Execute())
    {
        ListBox1->Items->LoadFromFile(OpenDialog1->FileName);
    }
}

void __fastcall TForm1::Mavjudplaylistnisaqlash1Click(TObject *Sender)
{
    if(SaveDialog1->Execute())
    {
        ListBox1->Items->SaveToFile(SaveDialog1->FileName);
    }
}
```

Dasturimiz videofayllarni namoyish eta olishi uchun forma markazidagi panelni ekran sifatida ishlatamiz. Buning uchun MediaPlayermizni Display xususiyatiga markaziy panel nomini belgilaymiz. Panel ekran sifatida ajralib turishi uchun BevelWidth xususiyatiga biror son yozib qo'yamiz. Masalan, BevelWidth = 10.

3.2. Dasturda ishlatilgan asosiy kodlar

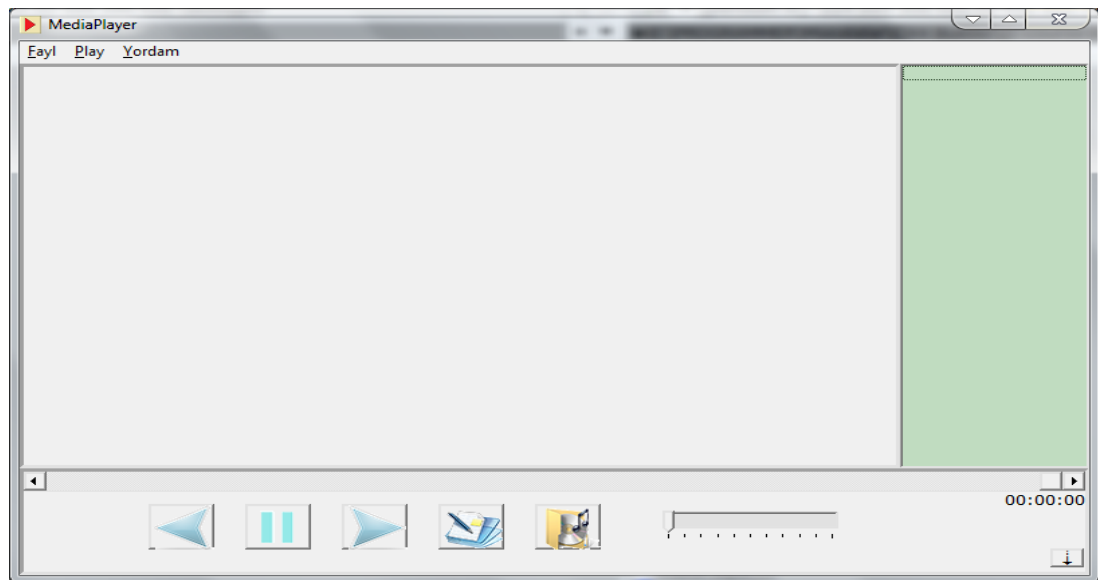
Ushbu dasturda men asosan MediaPlayer komponentasini ishini yengillashtiruvchi tugmalar keyingi va oldingi kodlarini yozdim. Chunki aynan shu kodlar dasturning asosiy qora ishchisi bo'lib xizmat qildi. Quyida ushbu tugma kodlari keltirilgan:

```
void __fastcall TForm1::oldingiClick(TObject *Sender)
{
    int oldingi = ListBox1->ItemIndex;
    if(oldingi==0)
    {
        oldingi=ListBox1->Count;
    }
    oldingi--;
    MediaPlayer1-> FileName=ListBox1-> Items-> Strings[oldingi] ;
    MediaPlayer1->Open();
    MediaPlayer1->Play();
    ListBox1->ItemIndex=oldingi; }

void __fastcall TForm1::keyingiClick(TObject *Sender)
{
    Timer1->Enabled=True;
    int keyingi = ListBox1->ItemIndex;
    if(keyingi==ListBox1->Count-1)
    {
        keyingi=-1;
    }
    keyingi++;
    MediaPlayer1-> FileName=ListBox1-> Items-> Strings[keyingi] ;
    MediaPlayer1->Open();
    MediaPlayer1->Play();
    ListBox1->ItemIndex=keyingi}
```

IV. Tadbiq qilish qismi

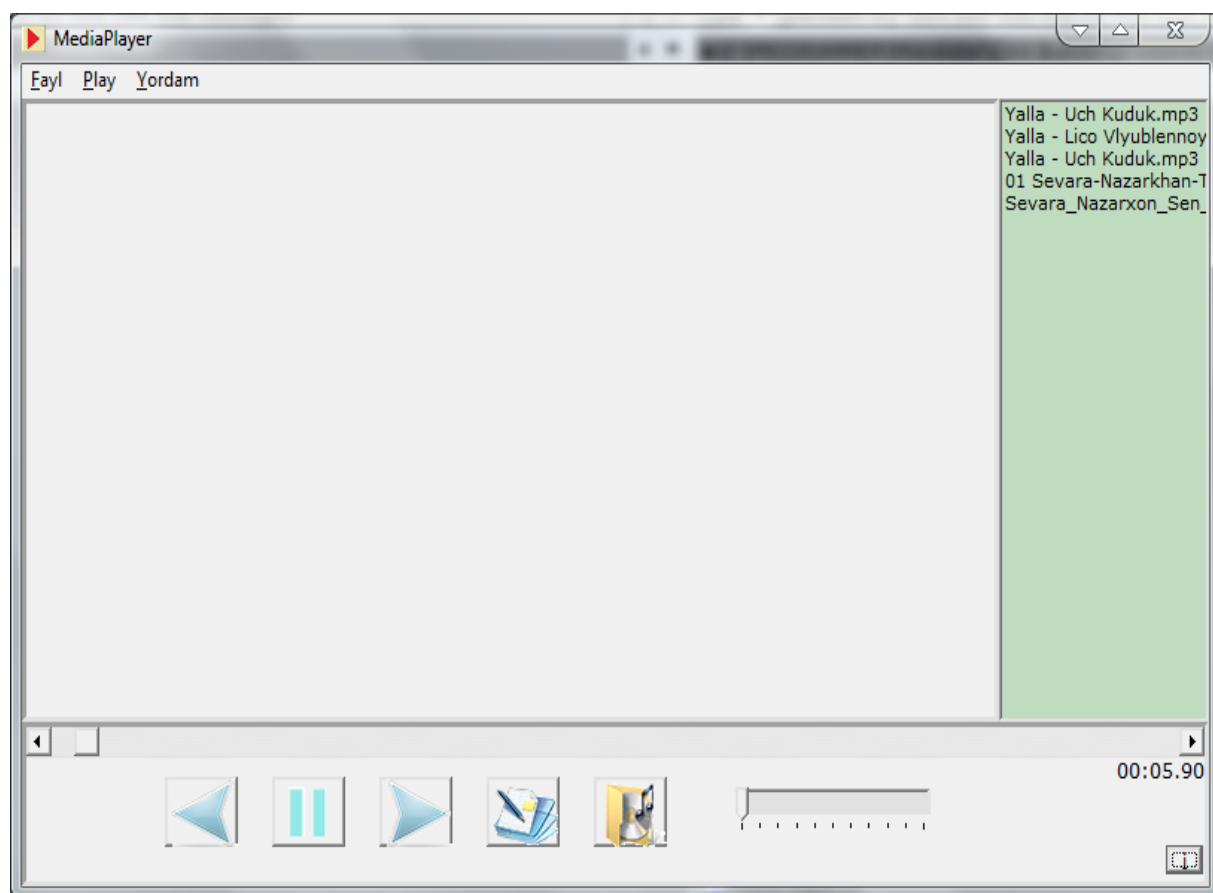
Ushbu dastur Mediafayllarni o'qish va ko'rish uchun mo'ljallangan. Dasturda videofayllarni ko'rish uchun maxsus display yaratilgan. Ushbu display orqali turli xil videoroliklarni ko'rish mumkin. Dasturda qo'shiqlar tinglashdagi imkoniyatlarga ham katta e'tibor berilgan. Siz dasturda biror qo'shiqlarni eshitish bilan birga, bu qo'shiqlarni musiqiy playlist sifatida saqlash imkoniyatga egasiz. Saqlangan yoki mavjud musiqiy playlistlarni ochish ham dasturga qiyinchilik tug'dirmaydi. Quyida dasturni ishga tushish jarayoni va ishlash muhitining limiti ko'rsatilgan:



4.1-rasm. Dastur ochilgandagi holat



4.2-rasm. Dasturning videofayllar bilan ishlash jarayoni



4.3-rasm. Dasturning audio fayllar bilan ishlash jarayoni

Xulosa

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++ Builderda mavjud bo'lgan ko'plab komponentalar haqida ma'lumotga ega bo'ldim. Komponentalarning turlari, ularning vazifalari, dasturda yaratadigan qulayliklari haqida tushunchaga ega bo'ldim va ko'plab komponentalarni o'z dasturimni tuzish davomida ishlatdim. Kurs ishini bajarish davomida dasturdagi ko'plab komponentalar palitrasidan, komponentalardan foydalanishni o'rgandim. O'z dasturimni tayyorlash jarayonida dasturda yuzaga kelishi mumkin bo'lgan hatoliklar, hamda turli xildagi istisno holatlarni oldindan bilib, ularni bartaraf qilish bo'yicha kerakli chora tadbirlarni amalga oshirdim. Dasturda hatolikka olib keladigan ma'lumotlar kiritilganda foydalanuvchini ogoxlanitirish xabarlarini chiqaruvchi dastur kodlaridan foydalandim.

O'z dasturimni tayyorlash jarayonida shunga amin bo'ldimki, har bir dasturchi, o'z dasturini tuzar ekan, avvalambor uning foydalanuvchiga tushunarli, boshqarishga anson, hamma qismlari puxta hisob-kitob qilingan, dasturda yuz berishi mumkin bo'lgan xatoliklarni oldi olingan, dasturning yashash sikli hisbga olingan bo'lishi lozim ekan. Shundagina yaratilgan dastur foydalanuvchiga uzoq muddat, samarali hizmat qilishi mumkin.

Foydalanilgan adabiyotlar

1. “Yuksak ma’naviyat yengilmas kuch”. Islom Abdudg’anievich Karimov
Toshkent 2008.
2. “O’zbekiston XXI asr bo’sag’asida”. Islom Abdudg’anievich Karimov.
Toshkent 2000.
3. “Havfsizlik va barqaror taraqqiyot yo’lida”. Islom Abdudg’anievich
Karimov. Toshkent 2004.
4. Культин Н. Б. Самоучитель C++ Builder
5. О.Д.Вальпа - Borland C++ Builder. Экспресс-курс – 2006
6. Самоучитель программирования C++
7. Самоучитель программирования на языке C++ в системе C++Build
8. Федоренко Ю.П. - Алгоритмы и программы на C++ Builder
(Профессиональная серия для программистов) – 2010

Internet manbalari:

www.cyber-form.ru

www.wikipedia.org

www.programmer.com

ILOVA

```
void __fastcall TForm1::Ochish1Click(TObject *Sender)
{
    if(OpenDialog1->Execute())
    {
        MediaPlayer1->FileName=OpenDialog1->FileName;

        MediaPlayer1->Open();
        if(MediaPlayer1->Enabled==false)
        {
            Timer1->Enabled=true;
        }
        ScrollBar1->Max=MediaPlayer1->Length;

        MediaPlayer1->Play();
        ListBox1->Items->Add(ExtractFileName(OpenDialog1->FileName));

    }
}
```