

**O‘ZBEKISTON RESPUBLIKASI OLIY VA
O‘RTA MAXSUS TA’LIM VAZIRLIGI**

GULISTON DAVLAT UNIVERSITETI

AXBOROT TEXNOLOGIYALARI KAFEDRASI



DASTURLASH TILLARI

**fanidan
oquv - uslubiy majmua**

Bilim sohasi: 100000 - Gumanitar

Ta’lim sohasi: 110000 - Pedagogika

**Ta’lim yo’nalishi: 5110700 - Informatika o’qitish metodikasi
V semestr**

Guliston – 2020

O'quv–uslubiy majmua Oliy va o'rta maxsus ta'lim vazirligi tomonidan 2018 yil 27.03. dagi 274 - sonli buyrug'ining 2 - ilovasi bilan tasdiqlangan Dasturlash tillari fani dasturi (№ BD -5112100 2.02 2018 – yil 03.03) talablari asosida tayyorlangan.

Tuzuvchi: Abdurahimov D.B. - GulDU “Axborot texnologiyalari” kafedrası dotsenti, pedagogika fanlari nomzodi

Negmatulloev Z.T. - GulDU, “Axborot texnologiyalari” kafedrası kattai o'qituvchisi.

Taqrizchi: D.E.Toshtemirov - GulDU “Axborot texnologiyalari” kafedrası dotsenti, pedagogika fanlari nomzodi

Ushbu o'quv–uslubiy majmua Guliston davlat universiteti O'quv – metodik Kengashi tomonidan (29.08.2020 y. dagi, № 1 - sonli bayonnoma) nashrga tavsiya etilgan.

Mundarija

Bet

1	Kirish	
2	Ma'ruza mashg'ulotlari	
3	Amaliy mashg'ulotlari	
4	Laboratoriya mashg'ulotlari	
5	Mustaqil ta'lim mashg'ulotlari	
6	Adabiyotlar ro'yxati	
7	Glossariy	
8	Ilovalar:	
	Fan dasturi	
	Ishchi fan dasturi	
	Tarqatma materiallar	
	Testlar	

Kirish

O'quv fanining dolzarbligi va oliy kasbiy ta'limdagi o'rni

Mustaqil Respublikamizda yuz berayotgan siyosiy, iqtisodiy, ilmiy-texnikaviy va madaniy o'zgarishlar Oliy ta'lim tizimida ham o'z aksini topmoqda. O'zbekistonda uzluksiz ta'lim-tarbiya tizimini yaratish, shu asosida ta'lim sifatini jahon andozalari darajasiga etkazish ta'lim sistemasining eng dolzarb vazifasiga aylandi. Bu esa barcha mutaxassisliklar qatori Informatika va dasturlash bo'yicha kadrlar tayyorlash sifatini oshirishni ham taqozo etadi. Bu maqsad vazifalar ushbu fan dasturi mazmunini ham belgilaydi. Algoritm konsepsiyasining vujudga kelishi bilan algebra, sonlar nazariyasi, geometriya va matematikaning boshqa sohalariga tegishli bir qator muammolarning echimli yoki echimli emasligini aniqlashtirish imkonini berdi. Algoritm nazariyasi faoliyat sohasi EHMlar vujudga kelishi bilan yanada kengaydi. Yuqoridagi fikrlar "Dasturlash tillari" fanining asosiy mazmunini belgilashga yordam beradi.

"Dasturlash tillari" fani umumkasbiy fanlar blokiga kiritilgan kurs hisoblanib, 2- va 3-kurslarda o'qitilishi maqsadga muvofiq. "Dasturlash tillari" fani "Informatika o'qitish metodikasi" ta'lim yo'nalishida o'qitiladi. Mazkur fan Algoritm nazariyasi fanining nazariy va uslubiy asosini tashkil qilib, o'z rivojida aniq va tabiiy fanlar uchun zamin bo'lib xizmat qiladi.

O'quv fanining maqsadi va vazifasi

"Dasturlash tillari" fanini o'qitishdan maqsad – talabalarga dasturlashning ilmiy-nazariy asoslarini, informatika o'qituvchisining kasbiy sohasida egallashi lozim bo'lgan bilimlar, amalda qo'llash uchun ko'nikma va makalalarni shakllantirish hamda rivojlantirishdan iborat.

Ushbu maqsadga erishish uchun fan talabalarni ob'ektga yo'naltirilgan dasturlash tillarida ishlash, amaliy masalalarga dasturlar tuzishga oid nazariy bilimlar, amaliy ko'nikma va malakalarini shakllantirish vazifalarini bajaradi.

Fan bo'yicha talabalarining bilim, ko'nikma va malakalariga quyidagi talablar qo'yiladi.

Talaba:

–ob'yektga yo'naltirilgan dasturlash tillarining nazariy asoslari, ob'yektlarni loyihalash, matematik va interfeys ob'yektlari, voqealar va xabarlar, ob'yektga yo'naltirilgan muhitlarda xabarlarni uzatish, ularga ishlov berish mexanizmlari, ob'yektlar iyerarxiyasi asosida dasturlarni loyihalash, muayyan ob'yektga yo'naltirilgan dasturlash tillari to'g'risida **tasavvurga ega bo'lishi**;

–ob'yektga yo'naltirilgan dasturlash tillarida chiziqli, tarmoqlanuvchi va takrorlanuvchi va modulli dasturlar tuza olishni, dasturlashning ob'yektga yo'naltirilgan paradigmasini, ob'yektga yo'naltirilgan muhitlarda dasturlarni loyihalashni **bilishi va ulardan foydalana olishi**;

–ob'yektga yo'naltirilgan dasturlash tillari muhitida ishlash, masalalarni tahlil qila olish, muayyan dasturlash tillari yordamida masalalarning dasturini tuzish va natijalarni taqqoslay olish **ko'nikmalariga ega bo'lishi lozim**.

Ma'ruza matnlari

Mavzu: Borland C++ Builder dasturlash muhiti

Reja:

1. Borland C++ Builder dasturlash muhitining interfeysi.
2. Borland C++ Builder dasturlash muhitining menyular satri.
3. Borland C++ Builder dasturlash muhitining tezkor tugmalari

1. Ixtiyoriy dasturlashdan minimal bilimga (dasturlashdan maktab kursidan) ega foydalanuvchi ham Borland C++ Builder 6 muhitni tezda o'rgana oladi. Bu muhitning soddaligi professional programmist bo'lishga xalaqit qilmaydi.

Dasturlash muhitlari ichida eng sodda muhitlardan biri hisoblanadi, bir nechta mashqlarni mustaqil bajargandan keyin, muhitdan kishining ajralgisi kelmaydi.

Umuman har qanday yangi texnologiyani o'zlashtirayotganda bilgan o'rgangan bilimlarni amalda tez-tez takrorlab turmasa xotirada saqlab qolishning iloji bo'lmaydi. Yangi tushunchani takrorlab xotiraga joylash va uni tadbiq qilishni bilgandan keyingina xotirada mustahkam joylashadi. Har bir dasturlash tili, muhiti o'zidan oldingi tillarning mukammal tomonlarini o'zida mujassamlashtiradi, uncha muvaffaqiyat qozonmagan tomonlarini mukammallashtirish hisobiga rivojlantiriladi. Shunga ko'ra bitta zamonaviy tilni mukammal bilgan dasturchi ikkinchi tilni katta kuch va vaqt sarflamasdan o'rgana oladi. Dasturchilar tillarni o'rganayotganda birini ikkinchisi bilan solishtirib o'rganishadi. Masalan, Obyektga Yo'naltirilgan Dasturlash (OYD) zamonaviy dasturlash tillarining asosini tashkil qiladi. Demak barcha zamonaviy dasturlash tillari umumiy qoidalarga bo'ysunadi.

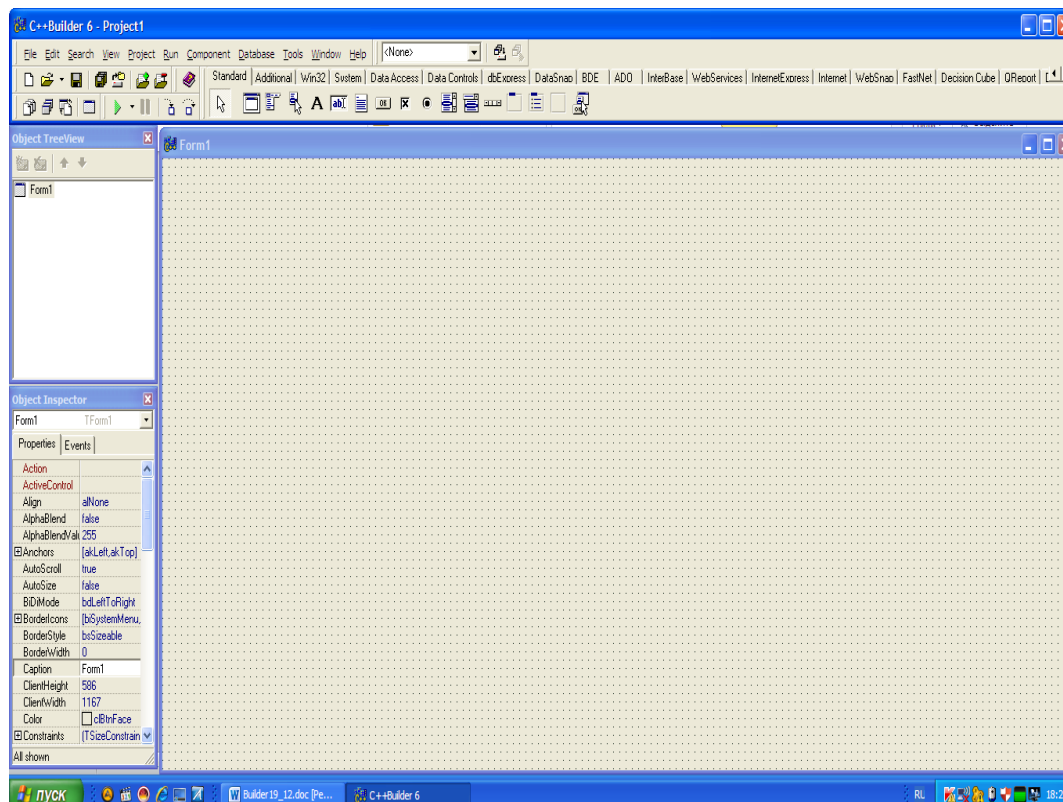
OYDda har bir dasturlash elementi obyekt sifatida qaraladi. Bunda obyekt, barcha hollarda ma'lum umumiy qoidalarga va xususiy qoidalarga ega bo'ladi. Bu obyektlar oynalar, tugmalar, konteynerlar va Canvas lar ko'rinishida bo'lishi mumkin.

Bunda har bir ilova kichik qismlarga ajratiladi va yakunida bu qismlar birlashtiriladi.

Borland C++ Builder 6 muhitida Windows osti ilovalar yaratish juda oson. Borland C++ Builder 6 interfeysi 1-rasmda keltirilgan.

OYD da qadamba qadam alohida uncha katta bo'lmagan dasturlarni funktsiya metodlari amalga oshirsa, harakatlanuvchi jarayonlarni qayta ishlash uchun xodisa amalidan, chaqiriluvchi obyektlarni tugmalar va oynalar bilan ifodalaydi.

Dasturchilar tilida bu interfeysga tez qayta ishlovchi muhit RAD (Rapid Application Development) deb atashadi. Bunga sabab bu muhitda dastur, ilova tuzish va uning dizaynini qurishda tayyor obyektlar va kutubxonada mavjud metodlardan foydalanish mumkin.



1-rasm. Borland C++ Builder 6 interfeysi

Masalan, kompyuter avtomatik ravishda xodisani qayta ishlovchi funktsiya dasturi matnini hosil qiladi.

Interfeysning tepasi 2-rasmda C++ Builder 6 – Project1 asosiy oynasi – joylashgan



2-rasm. Interfeysning asosiy oynasi

Ilova bu – tayyor bajariluvchi fayl hosil qilish uchun kerak bo'ladigan barcha fayllar to'plamidir. Masalan, ilova tarkibiga dastur matni, tovush fayllari, ikonka rasmlari va shu kabi ilovaga, kerakli fayllar kirishi mumkin. Bunda har bir ilova uchun alohida papka hosil qilish maqsadga muvofiq sanaladi. Chunki ilovani boshqa kompyuterga o'tkazmoqchi bo'lsak va papkada saqlamagan bo'lsak, ularni yig'ishimizga to'g'ri keladi. Bunda interfeysning o'zi ilovani saqlashni talab qiladi, ammo papka hosil qilishni o'zimiz bilishimiz kerak.

Interfeysning asosiy oynasi sarlavha oynasida ilova nomi, ilovani tiklash, berkitish tugmalari joylashgan. Sarlavha nomi tagida asosiy menyu joylashgan. Bu menyu orqali muhitning barcha funktsiya va komandalarini ishga tushirish mumkin. Asosiy menyuning tagida tez tugmalar joylashgan. Bu tugmalar ma'nolariga ko'ra guruhlariga ajratilgan. Bu tugmalar orqali tez-tez ishlatiladigan komandalarni ishga tushirishimiz mumkin. Bu tugmalarning o'ng tomonida vizual komponentalar VCL

(Visual Component Library - vizual komponentlar kutubxonasi) palitrasi joylashgan. Bu shunday obyektlarki yoki shunday dasturlash komponentlariki, bular yordamida Windows uchun vizual dasturlarni tezda yaratish mumkin. Komponentalar yordamida har xil tugmalar, rasmlar, yozuvlar, taymerlar, kalendar va hokazolarni ilovaga kiritishimiz mumkin. Vizual komponentalar palitrasi ma'nosiga va vazifasiga ko'ra guruhlariga ajratilgan.

Bu muhitning barcha oynalarini berkitish mumkin faqat asosiy oynani berkitsa ilovadan chiqib ketiladi, boshqa barcha oynalar o'lchamlarini kattalashtirish va kichiklashtirish imkoniyatiga egamiz.

Vizual komponentalar palitrasi. Vizual komponentalar palitrasini bir qismi monitorda ko'rinib turadi, qolganlarini o'ng va chapga siljituvchi tugmachalar vositasida ko'rishimiz mumkin.

Ekran (monitor) markazida forma (shakl) dizayneri joylashgan (3-rasm).

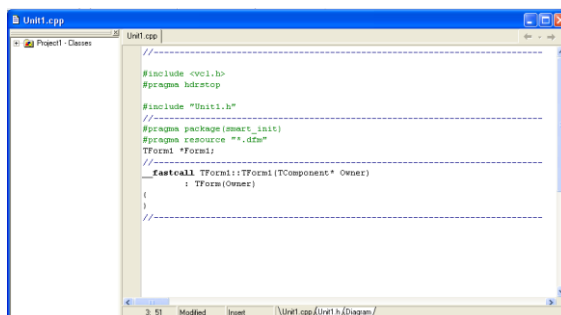


3-rasm. Forma (shakl) dizayneri

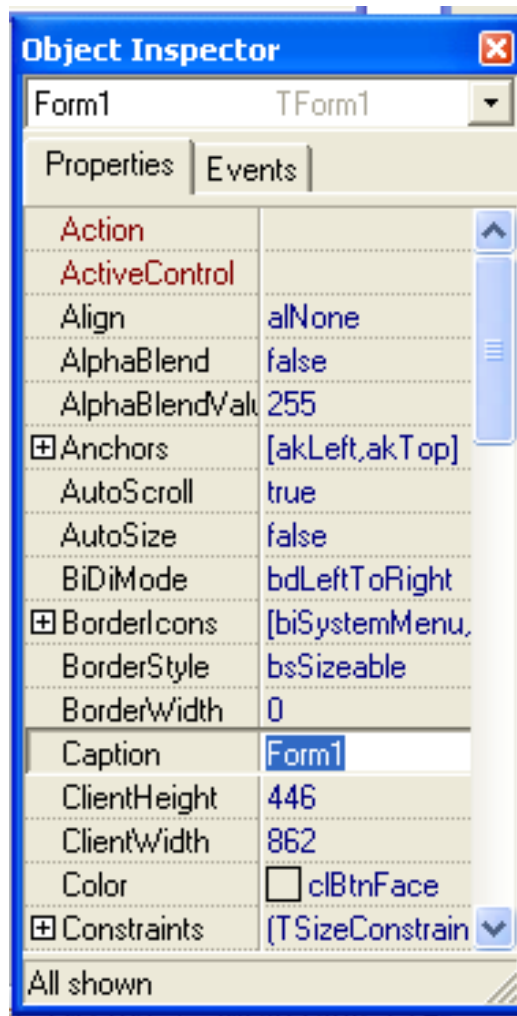
Bu bo'lg'usi dasturning interfeysini hosil qiluvchi oynadir. Oynaning nomi va sarlavhasi uning tepasida yozilgan bo'ladi. Odatda, Form 1 (Form 2, Form 3, Form 4) va shuningdek oynani berkituvchi va kichraytiruvchi tugmachalar ekranning o'ng tepasida joylashgan bo'ladi. Oynaning sathiga vizual dastur uchun zarur bo'ladigan VCL komponentalar joylashtiriladi.

Forma (shakl) dizayneri ostida dastur kodi muharriri joylashgan, odatda u **Unit1.cpp** sarlavha bilan beriladi. Bu oyna dastur kodini kiritish va tahrir qilish uchun mo'ljallangan (4- rasm).

Ekraning chap past qismida Obyektlar inspektori oynasi joylashgan. Bu oyna Object Inspector kabi sarlavxa bilan beriladi (5-rasm). Bu oynada vizual komponentalarga xususiyatlar o'rnatiladi. Bu oynani dasturchi o'zi xoxlagan ekran sathiga joylashtirishi mumkin.

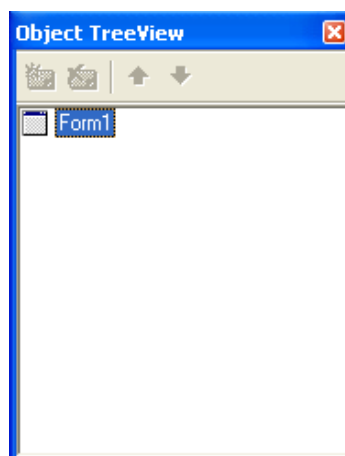


4-rasm. Dastur kodi muharriri



5-rasm. Obyektlar inspektori oynasi

Obyektlar inspektori oynasi ostida Obyektlarni daraxtsimon ko'rinishi oynasi (6-rasm) joylashgan. Bu oynada ilovadagi barcha obyektlar daraxt strukturasi shaklida ifodalangan bo'ladi. Formalar, dastur kodi va boshqa dastur tarkiblari fayllari berilgan bo'ladi.



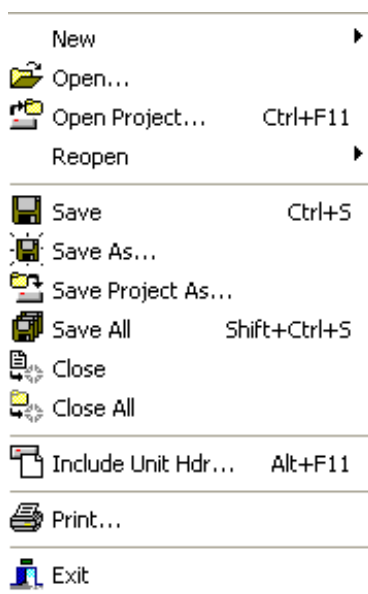
6-rasm. Obyektlarni daraxtsimon ko'rish oynasi.

Bu interfeys ilova tuzatayotganda eng asosiy foydalanayotgan qurolimiz bo'ladi.

Borland C++ Builder 6 muhiti asosiy menyu buyruqlari

File buyruqlari guruhi

Menyuning birinchi File (Fayl) buyrug'i guruhi 7-rasmda berilgan

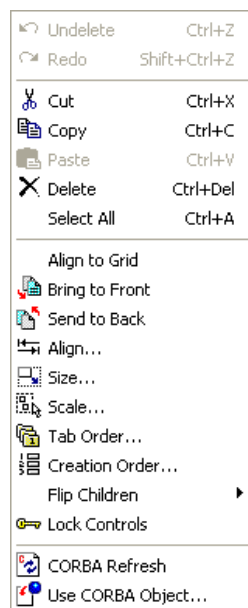


7-rasm. File buyrug'i guruhi

Bu menyu buyruqlari fayllar bilan ishlashga mo'ljallangan va qo'yidagi amallarni bajaradi: (**New**) yangi fayl hosil qilish, forma (oyna) ochish, mavjud fayllarni (**Open**) ochish, fayllarni saqlash (**Save**) va (**Close**) yopish, ilova dastur kodi matnini (**Print**) chop qilish mumkin.

Edit buyruqlari guruhi

Edit (Muharrir), 8-rasmda ifodalangan.

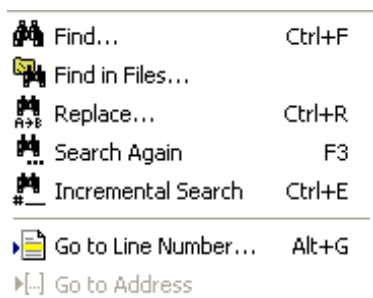


8-rasm. Edit buyruqlari guruhi

Menyuning bu buyruqlari guruhi tahrir qilish uchun zarur bo'lgan buyruqlardir. Masalan inkor qilish (**Undo**) va (**Redo**) takrorlash, qirqib olish(**Cut**), nusxa olish (**Copy**), nusxa olinganni qo'yish (**Paste**) va o'chirish (**Delete**) kabi amallarni o'z ichiga olgan.

Search buyruqlari guruhi

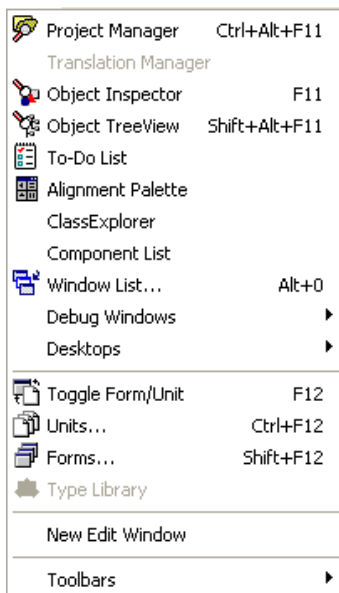
Search (Izlash)



9-rasm. Search (Izlash)

Menyuning bu buyruqlari guruhi fayllardan biror matnni izlash va avtomatik almashtirish, kerakli satrga o'tish va uni o'zgartish imkonini beradi.

View buyruqlari guruhi.

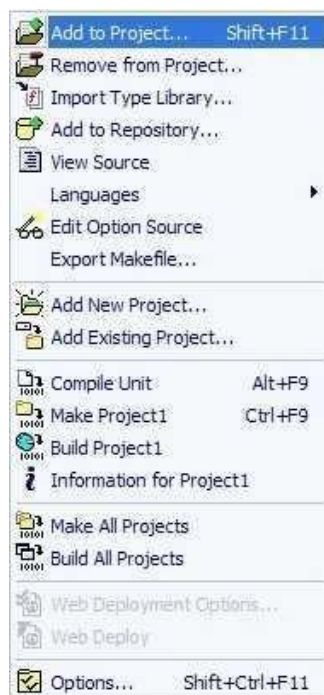


10-rasm. View(Ko'rish) buyruqlar guruhi

Menyuning bu buyruqlari guruhidan ilova va komponentalarni boshqarishni asosiy muloqot oynalari chaqiriladi. Masalan, ilova menedjeri (**Project Manager**), komponentlar ro'yxati (**Component List**) va oynalar ro'yxati (**Window List**).

Project buyruqlari guruhi

Project (Ilova) 11-rasmda keltirilgan.

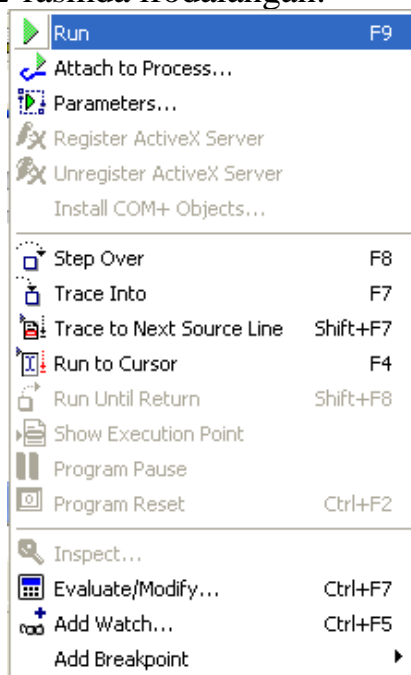


11-rasm. Project buyruq guruhlari

Menyuning bu buyruqlari guruhida ilovani boshqarish buyruqlari yig'ilgan. Bu buyruqlar yordamida fayllarni qo'shish, o'chirish, VCL komponentalar kutubxonasiga komponenta qo'shish, kompilyatsiya qilish va shunga o'xshash amallarni bajarish mumkin.

Run buyruqlari guruhi

Run (Bajarish) 12-rasmda ifodalangan.

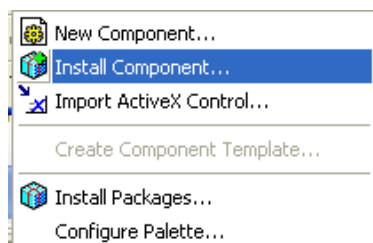


12-rasm. Run buyruqlar guruhi

Menyuning bu buyruqlari guruhi yordamida ilovani ishga tushirish va bekor qilish, ilovani butunlay va qadamba-qadam rejimda ishga tushirish, ko'rish uchun qo'shimcha o'zgaruvchilar kiritish, ilova bajarilishini to'xtatuvchi belgilar kiritish mumkin.

Component buyruqlari guruhi

Component(Komponent) buyruqlar guruhi 13-rasmda berilgan.

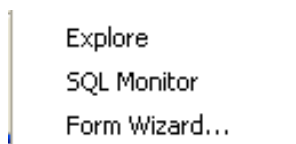


13-rasm. Component (Komponent) buyruqlar guruhi

Menyuning bu buyruqlari guruhi yordamida yangi komponentalarni qo'shish va konfiguratsiyalarini aniqlash ishlari bajariladi.

Database buyruqlari guruhi

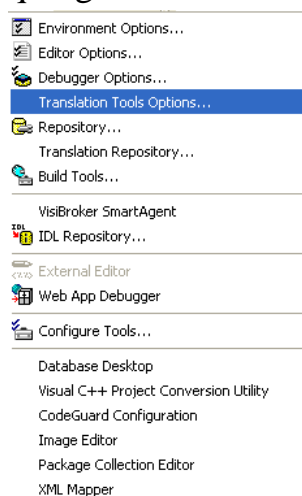
Database (Ma'lumotlar bazasi) buyruqlar guruhi (14-rasm) ma'lumotlar bazasi bilan ishlaydigan buyruqlarni o'z ichiga olgan.



14-rasm. Database (Ma'lumotlar bazasi) buyruqlar guruhi

Tools buyruqlari guruhi

Tools (Instrumentlar) buyruqlar guruhi 15-rasmda berilgan



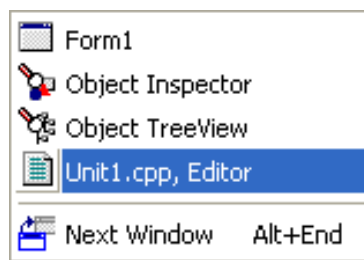
15-rasm. Tools buyruqlar guruhi

Menyuning bu buyruqlari guruhi yordamida ilova parametrlarini o'rnatish va

yordamchi dasturlar buyruqlarini chaqirish mumkin.

Windows buyruqlari guruhi

Windows (oyna) buyruqlari guruhi

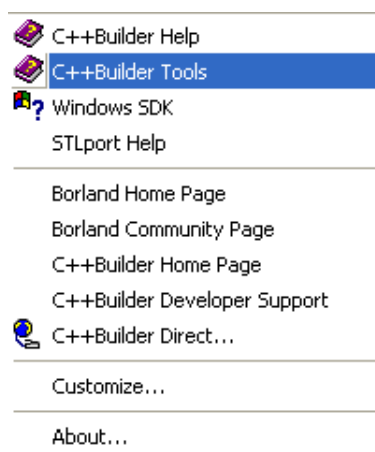


16-rasm. Windows(oyna) buyruqlari guruhi

Menyuning bu buyruqlari guruhi yordamida interfeysning oynalarini boshqarish mumkin.

Help buyruqlari guruhi

Help (Yordam) buyruqlari guruhi



17-rasm. Help (Yordam) buyruqlari guruhi

Menyuning bu buyruqlari guruhi yordamida muhit, til, komponentalar va kompyuter haqidagi ma'lumotlarni olishimiz mumkin. Masalan, quyidagicha:



18-rasm. About buyrug'i oynasi

Tezkor tugmalar

Tezkor tugmalar asosiy menyu bilan yonma-yon joylashgan.



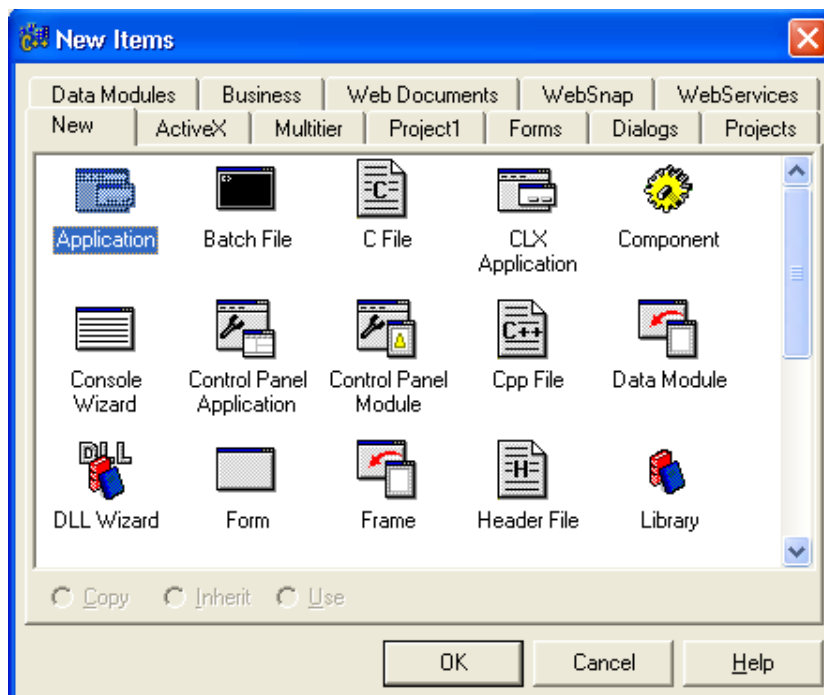
19-rasm. Tezkor tugmalar

Bu tugmalar ko'p ishlatiladigan, tez-tez murojaat qilinadigan bo'lganligi uchun alohida guruhlariga ajratilgan holda kichik piktogrammalar yordamida berilgan. Bu tugmalar asosiy menyuda ham mavjud, ammo vaqtdan yutish uchun ajratib ham qo'yilgan.

Standard (Standart) tugmalar.

Dasturning obyektlarini hosil qilish uchun Standart tugmalardan foydalaniladi. Bu obyektlarga dasturlar, yangi formalar, fayllar, biblioteka lar va hakoza kirishi mumkin. Yangi obyektни hosil qilish uchun **New** tugmasini bosamiz. Bunda **New Items** (20-rasm) muloqot oynasi paydo bo'ladi, bundan ilova tuzish uchun kerakli obyektни tanlaymiz. Bu oynada ko'plab obyektlar ma'lum guruhlariga ajratilgan holda joylashgan.

Bu guruhning boshqa tugmalari faylni saqlash (Save) va ochish (Open) uchun xizmat qiladi. AddFileToProject tugmalari orqali ilovaga fayl qo'shish va RemoveFileFromProject fayl o'chirish imkonini beradi.



20-rasm. New Items muloqot oynasi

View (Ko'rish) tugmasi.

Bu guruhga quyidagi tugmalar jamlangan (New Form) yangi forma hosil qiladigan, (ViewUnit) modullarni qurish, (ViewForm) formani qurish, (**Toggle Form/Unit**) formadan modulga, moduldan formaga o'tishni ta'minlaydigan tugmachalar.

Toggle Form Unit formadan kodni tahrirlashga, tahrirlashdan formaga o'tkazuvchi tugma. Agar ilovada bir nechta forma qatnashsa, ularni qurish uchun **View Form (Shift+F12)** tugmadan foydalaniladi.

Yangi forma hosil qilish uchun **New Form** tugma bosiladi. Bunda yangi forma dizayneri paydo bo'ladi. Odatda sarlavhasi **Form2** kabi nomlangan bo'ladi. Yangi forma ilovada yangi oynalarni ko'rsatish va ifodalash uchun zarur bo'lishi mumkin.

Debug tugmasi.

Bu guruhdagi tugmalar ilovani ishga tushirish uchun mo'ljallangan. **Run (F9)** tugmasi ilovani bajarish va ishga tushirish uchun, **Pause** ilovani vaqtincha to'xtatib turish uchun, **TraceInto (F7)** va **Stepover (F8)** ilovani qadamba-qadam bajarish uchun mo'ljallangan.

Custom tugmasi guruhi.

Bu guruhda Help contents tugmasi mavjud bo'lib, muhitning yordam tizimini chaqirish uchun ishlatiladi. Bu tugmalar guruhining joyi qat'iy tayinlanmagan, o'zimiz xoxishimizga qarab asosiy panelning istalgan joyiga joylashtirishimiz mumkin. Buning uchun kursorni tugmalar guruhining chap chekkasiga olib boramiz va sichqonchaning chap tugmasini bosib istalgan joyga joylashtirishimiz mumkin. Agar sichqonchaning o'ng tugmasini bossak 21-rasmdagi kabi menyu paydo bo'ladi. Bunda asosiy menyu panelidagi barcha tugmalar guruhini qayta o'rnatish mumkin.



21-rasm. Tugmalar guruhini o'rnatish

Mavzu: Borland C++ Builder komponentlar palitrasi. Hodisa jarayonlarini yaratish

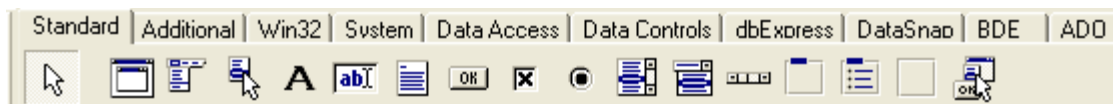
Reja:

1. Borland C++ Builder muhitining VCL vizual komponentalari.
2. Borland C++ Builder muhitining forma va kod muharriri oynasi.
3. Borland C++ Builder muhitida hodisa jarayonlarini yaratish.

VCL-komponentlar palitrasi

Borland C++ Builder 6 muhitini VCL vizual komponentalari bilan tanishamiz.

Borland C++ Builder 6 muhitida ilova yaratish VCL vizual komponentalaridan keraklilarini formaga joylashtirishdan boshlanadi. Komponentalar ilovaning g'ishtchalari (elementlari, qismlari) deyishimiz mumkin. Builder so'zining ma'nosi quruvchi degan ma'noni anglatadi. Barcha VCL komponentalari palitrasi, asosiy menyuning o'ng tomonida joylashgan bo'ladi. Komponentalar palitrasi ma'lum bir vazifalariga ko'ra komponentalar to'plamidan iborat komponentalar sahifasiga ajratilgan. Bu sahifalar ma'no va vazifasiga ko'ra bir-biriga yaqin bo'lgan komponentalarni bitta guruhga birlashtirgan. Sichqonchaning chap tugmasini sahifa komponentalari nomi ustida bossak, mazkur sahifa komponentalari guruhi ekranda paydo bo'ladi. Masalan, Standard sahifasini bossak, ekranda qo'yidagi komponentalar paydo bo'ladi (22- rasm).



22-rasm. Komponentalar oynasi

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 (formaga) olib o'tiladi.

Kutubxona Windows operatsion tizimlaridagi Foydalanuvchi Grafik Interfeysi standart interfeys obyektlarining to'liq inkapsulyatsiyalanishini 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 OYD imkoniyatlaridan to'liq foydalanadi. C++ Builder bu OMD ekan, OLE (OCX) boshqaruvchi elementlarni kiritish uncha qiyinchilik tug'dirmaydi.

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 funktsional imkoniyatlari ozgina kengaytirilgan yoki o'zining mutlaqo o'ziga xos ko'rinishi, xulq-atvori va kodining mazmuni bilan farqlanadigan bo'ladi. Komponentalarning yaratilishi OYD ning vorislik mexanizmiga tayanadi, cheklanishlarga deyarli ega bo'lmaydi hamda qo'yidagi bosqichlardan o'tadi:

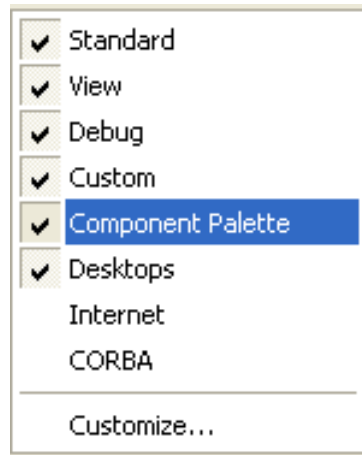
- mavjud komponenta turiga vorislik;
- yangi xususiyatlar, metodlar va voqealarni aniqlash;
- yaratilgan komponentani qayd etish.

Qidirish oson bo'lishi uchun, Palitra funktsional jihatdan o'xshash komponentalarni birlashtiradigan qo'shimcha ilovalar bilan bo'lingan. Tanlab olingan komponentaning kontekst menyusini unga sichqonchani o'ng tugmasini bosib ochish mumkin.

Bular ichidan **OK** yozuvli **Button** tugmasini, **A** yozuvli **Label** tugmasini osonlik bilan topish mumkin.

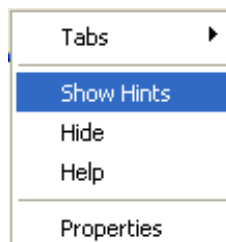
Agar sahifada komponentalar juda ko'p bo'lsa, harakatlanuvchi tugmalar orqali o'ng va chapga panelni harakatlantirish orqali barcha komponentalarni ko'rishimiz mumkin. Barcha komponentalar biror nom yoki maxsus ikonka orqali berilgan, agar sichqoncha kursorni ikonka ustiga oborsak shu komponenta nomi paydo bo'ladi. E'tibor bergan bo'lsangiz, ko'pchilik komponentalar to'rtburchak yoki dumaloq shaklga ega. Ixtiyoriy komponentani formaga ikki xil yul bilan joylashtirish mumkin. Komponenta ustida sichqonchani olib borib chap tugmasini ikki marta bossak mazkur komponenta forma markazida paydo bo'ladi. Ikkinchi usul sichqoncha chap tugmasini komponenta ustida bitta bosib siljitish bilan formaga joylashtirish mumkin. Formaga joylashtirilgan komponentani formaning ixtiyoriy qismiga siljitishimiz mumkin. Komponentaning o'lchamlarini sichqoncha yordamida o'zgartirish mumkin. Formada turgan ixtiyoriy komponentani aktivlashtirish yoki passivlashtirish mumkin. Aktivlashtirish degani shu komponentaning ustida amal bajarish xususiyatini o'zgartirish imkoniyatini beradi, buning uchun, mazkur komponentaning ustida sichqoncha tugmasini bosish kifoya. Aktivlik belgisi shu komponentani boshqalaridan ajralib belgilanib ko'rinib turadi. Buni obyektlar inspektoridan ham ko'rish mumkin. Sichqoncha yordamida tanlangan komponentani formadan o'chirib tashlash mumkin.

Komponentalar palitrasini asosiy menyu panelidan olib tashlash mumkin. Buning uchun sichqonchani o'ng tugmasini bosish va hosil bo'lgan menyu (23-rasm) dagi **Component Palette** so'z oldidagi belgini sichqoncha tugmasini ikkita bosib, olib tashlash kifoya qiladi.



23-rasm. Kontekst menyu oynasi

Komponentlar palitrasining o'zni Borland C++ Builder 6 interfeysida qat'iy tayinlanmagan. Uni o'zimiz xoxlagancha joylashtirishimiz mumkin. Buning uchun kursorni palitrani chap chekkasiga olib borib sichqonchani chap tugmasini bosib qo'yib yubormagan holda, yangi joyga joylashtirish mumkin. Agar palitra asosiy menyudan tashqariga chiqib ketsa yangi oyna ochiladi. Bunda komponentlar palitrasiga qo'shimcha buyruqlar kiritish uchun sichqonchani tugmasi bosiladi. Bunda kontekst menyu paydo bo'ladi (24-rasm).

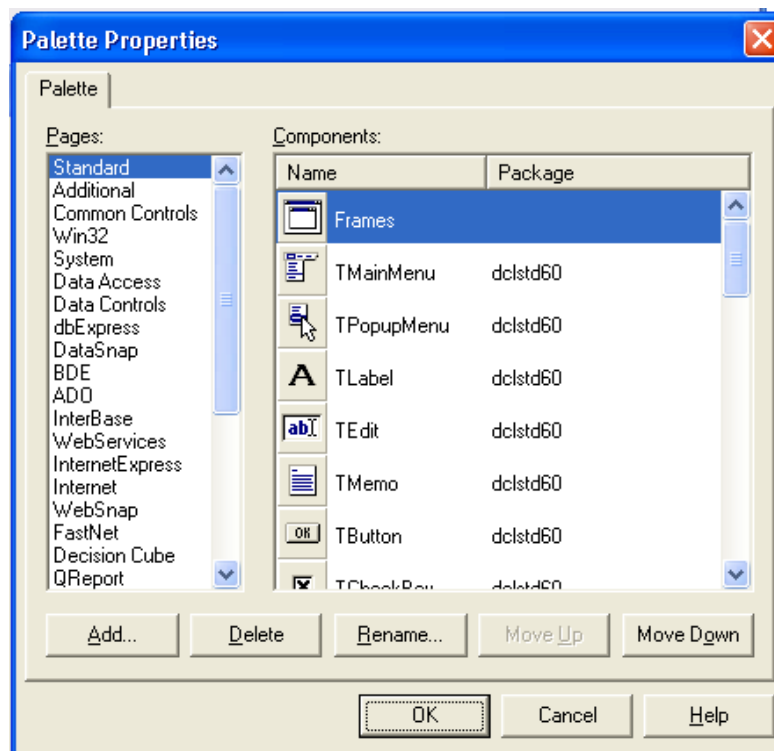


24-rasm. Buyruqlar palitrasi kontekst menyusi.

Bu oynada ixtiyoriy sahifaga **Tabs vositasida** tez o'tish imkoniyati mavjud (25-rasm), **ShowHints** tushuntirish ma'lumotlari beriladi, **Hide** palitrani bekitish, **Help** ma'lumotlarni olish, yordamni chaqirish, **Properties** xususiyatlarni o'rnatish.

Bu oynada ixtiyoriy sahifani qaytadan ko'rib chiqish mumkin. **Rename** buyrug'i orqali sahifani qayta nomlash mumkin. **Add** buyrug'i orqali yangi sahifa qo'shish mumkin. **Delete** ixtiyoriy sahifani o'chirish mumkin. Bu sahifalarning berilish ketma-ketligini o'zgartirish imkoniga egamiz. Masalan, **System** sahifasi **Standard** sahifasidan keyin berilgan edi, **System** satrini aktivlashtiramiz va sichqonchani chap tugmasini bosamiz.

MoveUp buyrug'i orqali **Standard dan** tepaga joylashtiramiz va **OK** ni bosamiz. Endi **System** sahifasi Standard, sahifasidan oldin joylashgan bo'ladi. Xuddi shuningdek komponentlarni **MoveUp** va **MoveDown** buyruqlari, joylashish ketma-ketligini o'zgartirish mumkin.



25-rasm. Komponentalar palitrasini o'rnatish oynasi.

Yaratilayotgan ilova, ya'ni yangi loyixa ustida ishlash dialog oynasini, ya'ni boshlang'ich formani qurishdan boshlanadi.

Boshlang'ich forma **Form1** formasining hususiyatlarini o'zgartirish hamda unga extiyojga qarab kerakli komponentalarni (kiritish va chiqarish maydonlar, buyruqli tugmalar) ni o'rnatish orqali yaratiladi.

Formaning hususiyatlari (1-jadval) uning tashqi ko'rinishi, o'lchamlari, sarlavha matni hoshiyasining ko'rinishini belgilaydi. Forma va uning komponentalari hususiyatlari va qiymatlarini o'zgartirish uchun **Object Inspector** oynasidan foydalaniladi. Bu oynaning yuqori qismida ob'ektning nomi hamda hususiyatlarining joriy vaqtdagi qiymatlari ko'rsatiladi. **Properties** (xususiyati) qistirmasining chap kolonkasida ob'ektning hususiyatlari, o'ng tomonda esa ularning qiymatlari keltiriladi.

Formaning (*mform* ob'ektining) hususiyatlari

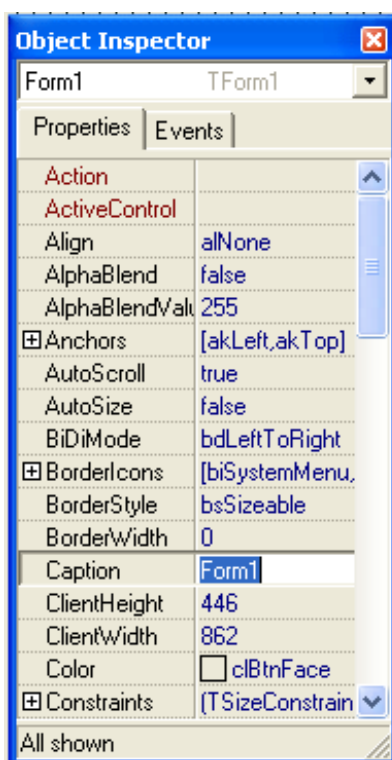
1-jadval

Hususiyat	Mazmuni
Name	Formaning nomi. Dasturda formaning nomi formani boshqarish va forma komponentalariga murojaat qilish uchun foydalaniladi.

Caption	Sarlavha matni
Width	Formaning kengligi
Height	Formaning balandligi
Top	Formaning yuqorichegarasidan ekranning yuqori chegarasigacha bo‘lgan masofa
Left	Formaning chap chegarasidan ekranning chap chegarasigacha bo‘lgan masofa
BorderStyle	CHegaraning ko‘rinishi. CHegara oddiy (bsSizeable), ingichka (bs Single) bo‘lishi yoki umuman bo‘lmasligi (bsNone) mumkin. Agar oynaning chegarasi oddiy bo‘lsa, uni foydalanuvchi sichqonchadan foydalanib, o‘zgartirish mumkin. Ingichka chegarali oyna o‘lchamlarini o‘zgartirib bo‘lmaydi. Agar chegara bo‘lmasa, u holda ekranga sarlavhasiz oyna chiqarilishi mumkin. Bunday oynaning holati va o‘lchamlarini dasturning ishi mobaynida o‘zgartirish mumkin emas.
BorderIcons	Oynani boshqarish tugmalari. Xususiyatinig qiymati dasturning ishi davomida foydalanuvchilar qaysi tugmalardan foydalanish mumkinligini ko‘rsatadi. Hususiyatning qiymati biSystemMenu, biMinimize, biMaximize va biHelp hususiyatlarining qiymatlarini aniqlash orqali beriladi. biSystemMenu xususiyati ixchamlash va sistema tugmalariga, biMinimize— ixchamlash tugmasiga, biMaximize — kengaytirish tugmasiga, biHelp — ma’lumotnomalarni chiqarish tugmasi bilan ishlashga ruxsat beradi.
Icon	Dialog oynasi sarlavhasidagi sistema menyusini chaqirishni anglatuvchi nishon.
Color	Fon rangi. Rangni uning nomini ko‘rsatib yoki operatsion sistemaning ranglari gammasiga bog‘lab qo‘yish orqali belgilash mumkin. Ikkinchi holda ranglar joriy ranglar sxemasi bo‘yicha aniqlanadi va operatsion sistemaning ranglar sxemasi o‘zgarganda o‘zgaradi.
Font	SHrift. Forma sirtida "ko‘rsatilmaganda ham" foydalaniladigan joriy shrift. Formaning xususiyati o‘zgartirilganda forma sirtida joylashgan komponentalarning Font xususiyati avtomatik tarzda o‘zgaradi, ya’ni komponentalar formadan Font xususiyatini meros qilib oladi.

Forma yaratishda birinchi navbatda caption (sarlavha) xususiyatining qiymatini o'zgartirish lozim. Bizning misolimizda "Form1" matning "yugurish tezligi" bilan almashtirish kerak. Buning uchun **Object Inspector** oynasida sichqoncha tugmasini **Caption** satrida chertamiz. Natijada hususiyatning joriy qiymati ajratib ko'rsatiladi va shu satrda kursor paydo bo'ladi. SHundan keyin "YUgurish tezligi" matnini kiritish mumkin. (12-rasm).

Xuddi shu usul bilan formaning kengligi va balandligini aniqlovchi **Height** va **width** hususiyatlarni o'zgartirish mumkin. Formaning o'lchamlari, uning holati hamda boshqa boshqaruv elementlarining o'lchamlari va ularning forma sirtidagi holatlari piksellarda (ekrandagi nuqtalar) beriladi. **Height** va **width** hususiyatlarini mos ravishda 250 va 330 qilib belgilang.



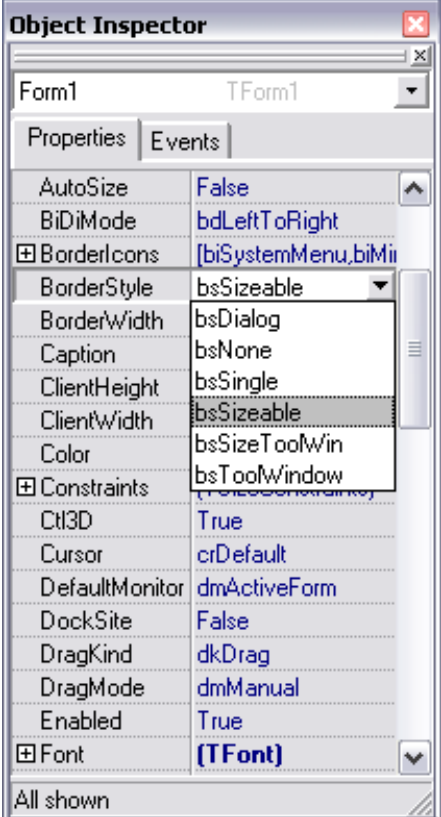
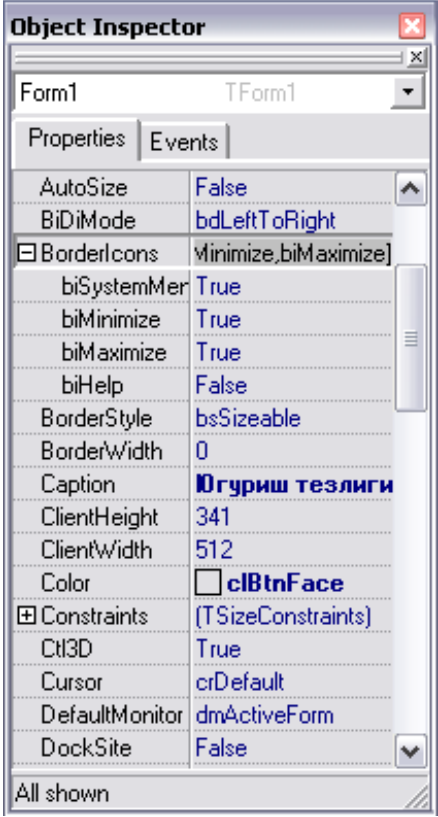
Hususiyat qiymatini qiymatni kiritish orqali o'zgartirish

Forma — bu oddiy oynadir. SHuning uchun uning o'lchamlarini boshqa oynalar kabi sichqoncha yordamida o'zgartirish mumkin. Bunda **Height** va **Width** hususiyatlarining qiymatlari avtomatik tarzda o'zgaradi.

Dialog oynasining ekrandagi holati formani tashkil qilishdagi holatiga mos keladi. Bu holatni **Tor** (ekranning yuqori chegarasidan chekinish) va **Left** (ekranning chap chegarasidan chekinish) hususiyatlarining qiymatlari belgilab beradi. Bu qiymatlarni ham sichqoncha yordamida o'zgartirish mumkin.

Ayrim hususiyatlarni tanlashda, masalan, **BorderStyle**, hususiyatning joriy qiymatini belgilashda, o'ng tomonda ochiladigan ro'yxat taklif qilinadi. Qiymatni ana shu ro'yxatdan tanlash mumkin.

Ayrim hususiyatlar murakkab hisoblanadi, chunki ularning qiymatlari boshqa hususiyatlarning qiymatlaridan kelib chiqib aniqlanadi. Bunday hususiyatlarning oldida "□" nishoni turadi. U chertilsa, aniqlovchi hususiyatlar ro'yxati taklif qilinadi. Masalan, BorderIcons xususiyati oynalarni boshqarishning qaysi tugmalari bilan dastur ishi davomida ishlash mumkinligini belgilaydi. Agar biMaximize xususiyatiga False qiymati berilsa, dasturning ishlashi jarayonida **Razvernut** tugmasi oyna sarlavhasida ko'rinmaydi.

	
<p>Hususiyat qiymatlarini ro'yxatdan tanlash</p>	<p>Murakkab hususiyatlarning ochilgan ro'yxati</p>

Ayrim hususiyatlarning yonida uch nuqtali tugma joylashgan. Bu hususiyat qiymatini aniqlashda yangi dialog oynasidan foydalanish mumkinligini anglatadi. Masalan, **Font** murakkab xususiyatining qiymatini belgilashda shrift tanlashning standart oynasidan foydalanish mumkin.

2 - jadvalda yaratilayotgan formaning o'zgartiriladigan hususiyatlari keltirilgan. O'zgaraydigan hususiyatlar bu jadvalga kiritilmagan.

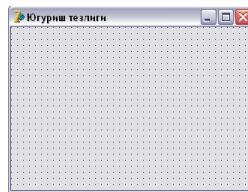
Boshlang'ich formaning hususiyatlari

2-jadval.

Hususiyat	qiymat
Caption	YUgurish tezligi
Height	250
Width	330

BorderStyle	bsSingle
BorderIcons . biMinimize	False
BorderIcons . biMaximize	False
Font. Size	10

Keltirilgan jadvalda ayrim hususiyatlarning yonida nuqta (.) belgisi turibdi. Bu hususiyatning aniqlanadigan qiymatini belgilashni bildiradi. Boshlang'ich formaning jadvaldagi hususiyat lari o'rnatilganidan so'ng, forma rasmdagi ko'rinishni oladi.



Yugurish tezligining dasturi foydalanuvchidan boshlang'ich ma'lumotlar – masofa hamda shu masofani yugurib bosib o'tish vaqtini olishi lozim. Bunday hollarda odatda boshlang'ich ma'lumotlar kiritish maydonlariga klaviatura yordamida kiritiladi. SHuning uchun formaga kiritish maydoni **Edit** komponentalarini joylashtirish lozim.

Eng ko'p foydalaniladigan komponentalar **Standard** qurollar panelida joylashtirilgan.

Formaga komponentani qo'shish uchun komponentalar palitrasidan shu komponenta piktogrammasi ustiga sichqonchani keltirib, chertiladi. So'ngra kursorni komponentaning chap yuqori burchagi formada joylashishi kerak bo'lgan nuqtaga o'rnatiladi va chap tugmani yana bir marta chertiladi. Natijada formada standart o'lchamli komponenta paydo bo'ladi.



Standard qurollar panelining komponentalari ro'yxati

Komponentaning o'lchamlarini ularni formaga qo'shish jarayonida belgilanishi mumkin. Buning uchun komponenta sichqoncha yordamida tanlanganidan so'ng, kursorni formaning komponenta chap yuqori burchagi turishi kerak bo'lgan nuqtasiga keltirib, chap tugmasi bosiladi va uni qo'yib yubormagan holda kursorni komponentaning quyi o'ng burchagi turishi kerak bo'lgan nuqtaga olib kelinadi. SHundan so'ng sichqonchani chap tugmasini qo'yib yuborish mumkin. Formada ko'rsatilgan o'lchamdagi komponenta paydo bo'ladi.

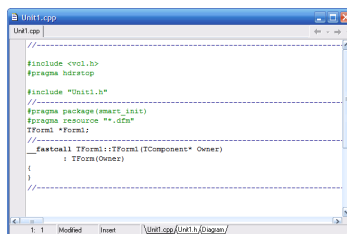
Delphi formaga qo'shilayotgan xar bir komponentaga nom beradi. Bu nom komponenta nomi va uning tartib nomeridan iborat bo'ladi. Masalan, formaga ikkinchi

Edit komponentasi qo‘shilgan bo‘lsa, ularning nomlari mos ravishda Edit1 va Edit2 bo‘ladi. Dasturchi Name xususiyati qiymatini o‘zgartirib, bu nomlarni boshqasiga almashtirishi mumkin. Odatda sodda dasturlarda komponentalarning nomlari almashtirilmaydi.

Rasmda formaning ikkita Edit komponentalari, ya’ni kiritish maydonlarini qo‘shilganidan keyingi holati keltirilgan. Komponentalarning biri ajratilgan. Ajratilgan komponentaning xususiyatlari **Object Inspector** oynasida tasvirlangan. Boshqa komponenta xususiyatlarini ko‘rish uchun sichqonchani chap tugmasini shu komponenta ustida chertish lozim. Shuningdek, komponenta nomini **Object TreeView** oynasidan yoki **Object Inspector** oynasining yuqori qismidagi ob’ektlarning ochiladigan ro‘yxatidan ham tanlash mumkin.

Kod muharriri

Dastur matnini hosil qilish uchun kod muharriridan foydalaniladi, bu oynaning tashqi ko‘rinishi 30-rasmda keltirilgan.



30-rasm. Kod muharriri oynasi

- Unit1.cpp ilovangizning bajarilayotgan ishga tushirish kodini saqlaydi. Aynan shu yerda siz foydalanuvchining komponentalar obyektlariga ta’siri paytidagi dastur reaksiyasiga javob beradigan voqealarning qayta ishlatgichlarini yozib qo‘yasiz.

- Unit1.h barcha obyektlar va ularning konstruktorlarining e’lonlariga ega. Voqealarni qayta ishlash funktsiyalari e’lonlaridagi kalit-so‘zga e’tibor bering (C++ Builder bu funktsiyalarni avtomatik tarzda generatsiya qiladi). `_fastcall` tufayli parametrlar stek orqali emas, balki markaziy protsessor registrari orqali uzatiladi. Voqealarni qayta ishlatgichlarning chaqirishlari tez-tez ro‘y berib turadi, shuning uchun stek xotirasidan parametrlarni tanlab olishga sarflanadigan vaqtning tejalishi ancha sezilarli natijalarni beradi. C++ Builder kompilyatsiya qiladigan va to‘playdigan ilovalarning yuqori darajada tez harakatlanishining sabablaridan biri ham shu yerda yashiringan.

- Project.cpp ilovada mujassamlangan barcha obyektlarga xizmat ko‘rsatadi. Har qanday yangi shakl, dasturiy modul yoki ma’lumotlar moduli avtomatik tarzda loyihaviy faylga kiritiladi.

Mavzu: Borland C++ Builder muhitining Standard komponentlar palitrasi



Standard komponentalar

Komponentalar palitrasining Standard qo'shimcha ilovalari komponentalari sizning dasturingizga Windows standart interfeysli elementlarning 14 tasining ulanishini amalga oshiradi.

TMainMenu



Bosh menyu komandolari Panelini va ularga mos keladigan tushib qoladigan menyularni yaratadi. Barcha menyu komandalarining identifikatorlari menyuning har qanday konkret komandasiga kirish huquqiga ega bo'lgan Items xususiyati bilan aniqlanadi. AutoMerge xususiyati Merge va UnMerge metodlari bilan birgalikda turli shakldagi menyularning birlashish jarayonini boshqaradi.

TPopupMenu



Shakl yoki bironta boshqa komponenta uchun maxsus menyu yaratadi. E'tiborga oling, aynan shu maqsad uchun har qanday boshqa komponenta PopUpMenu xususiyatiga ega bo'lib, bu xususiyatda siz uning bilan bog'liq menyuga iqtibos qilishingiz mumkin.

Agar siz sichqonchanning o'ng tugmasini shaklga yoki berilgan komponenta mansub bo'lgan biron boshqa elementga bosish bilan maxsus menyu ekranda paydo bo'lishini xoxlasangiz, AutoPopUp xususiyatining true qiymatini o'rnatib. Voqea qayta ishlatgichi - OnPopUp yordamida bevosita maxsus menyuning paydo bo'lishi oldidan bajariladigan protsedurani aniqlashi 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'rnatib. Kalta soha ichida matnning hammasini ko'rish imkoniga ega bo'lish uchun, WordWrap xususiyatining true qiymatini bering. Transparent xususiyatining true

qiymatini o'rnatangiz, 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 hatti- 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 bekorqilish 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 check-boxni yaratadi (bunda tavsifiy matn check-boxning vazifasini spetsifikatsiya qiladi).

Box holatini bildiruvchi «Check» biron-bir variantning tanlanishiga mos keladi (box ustidan tortilgan chiziq bilan belgilanadi). «UnCheck» 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. AllowGraed xususiyatining true qiymatini o'rnatib, boxni

to'qroq rangli (masalan, kulrang) qilish mumkin. State xususiyati joriy holatni va box 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. Radio-tugma bosilganda, Checked komponentasining xususiyati ham mos ravishda o'zgaradi va OnClick voqeasi yuzaga keladi.

Odatda radio-tugmalar avvaldan shaklda o'rnatilgan konteyner ichiga joylashtiriladi. Agar bitta tugma tanlangan bo'lsa, ushbu guruhga mansub barcha boshqa tugmalarning tanlovlari avtomatik tarzda olib tashlanadi. Masalan, shakldagi ikkita radio-tugma, agar ular boshqa-boshqa konteynerlarda joylashgan bo'lsagina bir paytning o'zida tanlab olinishi mumkin. Agar radio- tugmalarning guruhlanishi ochiq-oydin berilmagan bo'lsa, bu holda ularning hammasi, yashirin holda, konteyner darchalari (TForm, TGroupBox yoki TPanel) dan birida guruhlanadi.

TRadioButton komponentasi TButtonControl sinfining hosilasidir.

TListBox



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

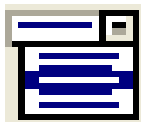
Agar ro'yxatdagi barcha elementlar ajratilgan sohaga sig'masa, ro'yxatni aylantirish lineykasi yordamida ko'rib chiqish mumkin. Ro'yxat elementlari Items xususiyatining ichida, dastur bajarilish vaqtida tanlab olinadigan element raqami esa ItemIndex xususiyatining ichida joylashgan bo'ladi. Ro'yxat elementlari matn muharririning darchasi Items xususiyati qiymatining grafasida tugmacha bilan ochiladi. Ro'yxat elementlarini Items obyektining Add, Append, Delete va Insert metodlari yordamida dinamik tarzda qo'shish, o'chirish, orasiga joylash va o'rnini almashtirish mumkin. Masalan:

ListBox1->Items->Add («Ro'yxatning oxirgi elementi»);

Sorted xususiyatining true qiymati ro'yxat elementlarini alifbo tartibida ajratib joylashtiradi.

TListBox komponentasi TCustomListBox sinfining hosilasi bo'lib, uning barcha xususiyat, metod va voqealariga vorislik qiladi.

TComboBox



Tahrir sohasi hamda matn variantlarining tushib qoladigan ro'yxati kombinatsiyasini tanlash uchun ishlatiladi.

Text xususiyatining qiymati bevosita tahrir sohasiga kiritib qo'yiladi. Foydalanuvchi tanlab olishi mumkin bo'lgan ro'yxat elementlari Items xususiyatining ichida bo'ladi. Dasturning bajarilish paytida tanlab olinishi mumkin bo'lgan element raqami ItemIndex xususiyatining ichida bo'ladi. 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.

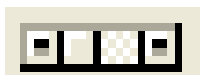
Items obyektining Add, Append, Delete va Insert metodlari yordamida ro'yxat elementlarini dinamik tarzda qo'shish, o'chirish orasiga qo'yish va o'rnini almashtirish mumkin. Masalan:

`ComboBox->Items->Insert(0, «Ro'yxatdagi birinchi element»);`

Sorted xususiyatining true elementi ro'yxat elementlarini alifbo tartibida navlarga ajratilishini ta'minlaydi. TComboBox komponentasining turini Style xususiyatidan tanlab olish mumkin.

TComboBox komponentasi TCustomComboBox sinfining hosilasi bo'lib uning barcha xususiyatlari, metodlari va voqealariga vorislik qiladi.

TScrollBar



Darcha, shakl yoki boshqa komponenta ichidagilarini ko'rib chiqish uchun ishlatiladi. Masalan, biron-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 SmallShange 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.

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 Windows ning bir nomdagi obyektning inkapsulyatsiyalanishidan iborat.

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

TPanel



Boshqa komponentlarni 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 komponentasi TCustomPanel sinfining hosilasi bo'lib, uning barcha xususiyatlar, metodlari va voqealari to'liq vorislik qiladi.

Mavzu: Borland C++ Builder Additional komponentlar palitrasi.



38-rasm. Additional komponentlari

Komponentalar palitrasining Additional qo'shimcha ilovalar komponentalari sizning dasturingizga Borland korporatsiyasi maxsus C++ Builder muhiti uchun ishlab chiqqan 9 ta boshqarish elementini kiritadi.

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 uchun xususiyatlarga ega bo'ladi. Siz C++ Builder qurilmasiga kirgan alohida tasvirlar katalogidagi grafik tugmalarning tayyor stillaridan foydalanishingiz ham, yoki bo'lmasa, tasvirlarni tahrir qilish tizimlarining biri tomonidan yaratilgan suratlardan foydalanishingiz ham mumkin.

Tugmaning turli holatlariga (masalan «bosilgan», «qo'yib yuborilgan», «ta'qiqlangan» va h.k.) turli bit obrazlari mos kelishi mumkin.

Tasvirlar Fayllari Muharririning bmp kengaytmali darchasi (39-rasm) Glyph xususiyatining qiymatlari tugmasi bilan ochiladi.



39-rasm. BMP kengaytmali bit obrazlari fayllari tasvirlarining muharriri

King xususiyati sizga yozuvlar va tegishli grafika (OK, Cancel, Hello va boshqalar) bilan ta'minlangan standartlashtirilgan tugmalarni yaratish imkonini beradi.

TSpeedButton



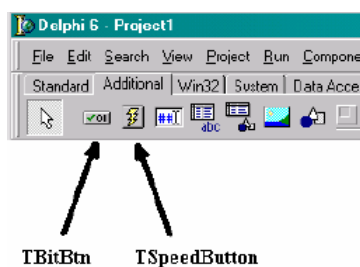
Odatda ma'lum menyu komandalarini tez chaqirish yoki rejimlarni o'rnatish Paneli (TPanel) da joylashtiriladigan grafik tugmani yaratadi.

Tezkor tugmaning turli holatlariga (masalan «bosilgan», «qo'yib yuborilgan»,

«ta'qiqlangan» 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 qiymtlarining grafasidagi tugma bilan ochiladi. Tez tugmalarning boshqa xususiyatlari ularning biron-bir guruhdagi ishini tashkil etadi.

TSpeedButton va TBitBtn tugmalari.

Bu tugmalar **TButton** vazifalarini bajaradi. Yagona farqi matndan tashqari racmlarni ham aks ettiradi. **TSpeedButton** tugmasi fokus olmaydi. Bu shuni bildiradiki, agar matn qatorida satr terib, bu tugma bosilsa, shu hodisa qayta ishlangandan so'ng fokus yana matn qatoriga qaytib keladi. TAB tugmasi bilan bu tugmani ajratib bo'lmaydi.



Tugmaga rasm o'rnatish uchun ikki marta *Glyph* xossasi qatorida chertish lozim. Natijada rasm paydo bo'lgan yuklash oynasida *Load* tugmasini bosish lozim. Ko'p rasmlar Program Files\Common Files\Borland Shared\Images\Buttons katalogida joylashgandir.



TBitBtn va **TSpeedButton** tugmalari deyarli bir xil xossalarga egadir. Ular uchun umumiy *Layout*, xossasi rasm va matn o'zaro joylashuvini o'zgartirishga imkon beradi. Quyidagi rasmda har xil qiymatlar mos variantlari ko'rsatilgan



TBitBtn tugmasining yana bir xossasi *Kind* bo'lib oldindan tayyorlangan standart tugmalarni tanlash imkonini beradi. Quyidagi rasmda standart tugmalar va ularga mos qiymatlarni ko'rish mumkin.

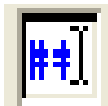


Рис 11.1.7 Различные значения свойства Kind.

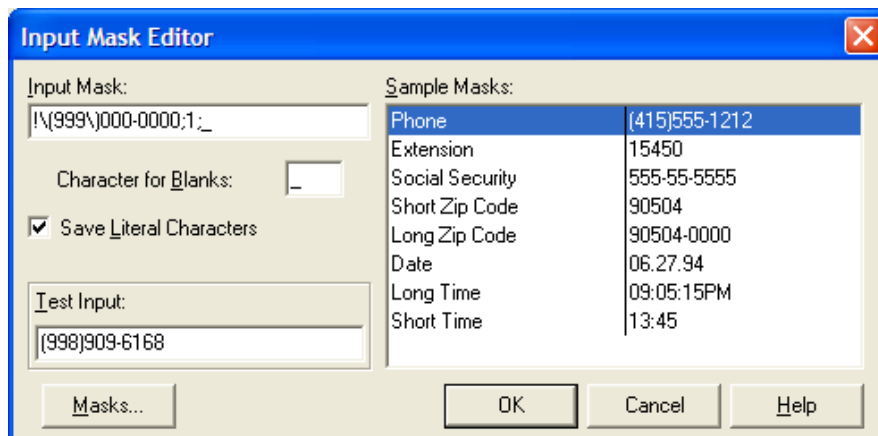
Yana bir xossa *ModalResult* – dialog oynasi uchun tugma qaytaradigan natijani tanlashga imkon beradi.

TSpeedButton tugmasining *GroupIndex* xossasi tugmalarni guruhlashga imkon beradi. Buning uchun bir guruhga tegishli tugmalarning *GroupIndex* xossasi bir xil qiymatga masalan 1 ga teng bo'lishi kerak. Guruhlangan tugmalarning biri bosilsa, qolganlaridan ajralib qoladi. Buning uchun *Down* xossasi qiymati *true* ga teng bo'lishi kerak.

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 (40-rasm) ushbu xususiyat qiymatlari grafasida tugma bilan ochiladi.



40-rasm. Telefon raqamlarini kiritish uchun niqobni yaratish

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

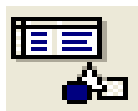
TStringGrid



Ushbu 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 to'tish imkonini beradi. Setkaning barcha simvolli ketma-ketliklari setkaning kerakli uyasini adreslaydigan (manzillaydigan) Cells xususiyatida mujassamdir.

TDrawGrid



To'zilma holiga keltirilgan grafik ma'lumotlarni satrlar yoki ustunlar bo'yicha aks ettirish uchun muntazam setka yaratadi. RowCount va ColCount xususiyatlari vertikal bo'yicha 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 xususiyatli tomonidan belgilanadi, aylantirish chiziqchalari esa ScrollBars xususiyati tomonidan qo'shiladi. FixedCol va FixedRows xususiyatlari ustunlar va satrlarning aylantirilishini ta'qiqlab qo'yish imkonini beradi, Fixed Color 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. Default Drawing xususiyatining false qiymatini o'rnatish uchun, setka uyalarini «qo'lda» to'ldirish uchun mo'ljallangan OnDrawCell voqeasi qayta ishlatgichining yozilishini talab qiladi. DefaultColWidths va DefaultRowHeights xususiyatlari yordamida yashirin tanlanayotgan barcha ustunlar va satrlarning enini o'rnatish mumkin. ColWidth va RowHeight xususiyatlari konkret ustun enini va konkret satr bo'yini spetsifikatsiyalaydi.

Dasturning ishlash paytida siz CellRest metodi yordamida biron-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.

Dastur bajarilish paytida qaysi satr setkaning ustki satri bo'lishini aniqlash yoki TopRow xususiyati yordamida ko'rsatilgan satrni ustki holatga qo'yib qo'yish mumkin. Qaysi ustun setkaning ko'rinib turadigan ustunni bo'lishini aniqlash uchun, LeftCol xususiyatidan foydalaning. VisibleColCount va VisibleRowCount xususiyatlarining qiymatlari setkaning ko'rinib turgan ustunlari va satrlarining umumiy sonini spetsifikatsiyalaydi.

TImage



Shaklda grafik tasvir konteynerini yaratadi (bu bit obrazi, piktogramma yoki metafile bo'lishi mumkin).

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

Tasvirlar fayllarining dinamik yuklanishi va saqlanishi uchun, Picture obyekt xususiyatining LoadFromFile va SaveToFile metodlaridan qo'yidagi turlar yordamida foydalaning:

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

TShape



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

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

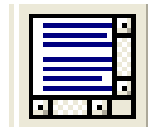
TBevel



Xuddi uskuna bilan o'yilgandek xajmli ko'rinadigan chiziqlar, box lar yoki ramkalarni yaratadi.

Komponenta chizayotgan obyektни Shape xususiyati aniqlaydi, Style xususiyatining qiymati esa obyekt ko'rinishini o'zgartirib, uni bo'rtiq yoki botiq holga keltiradi. Foydalanuvchi shakl o'lchamlarini o'zgartirganda ham obyektning nisbiy holatini o'zgarmas qoldirish uchun, Align xususiyatining true qiymatini o'rnating.

TScrollBar



Darchada o'lchamlari o'zgaruvchan box ni yaratadi, bu box 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 biron-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.

Mavzu: Borland C++ Builder Dialogs komponentlar palitrasi.

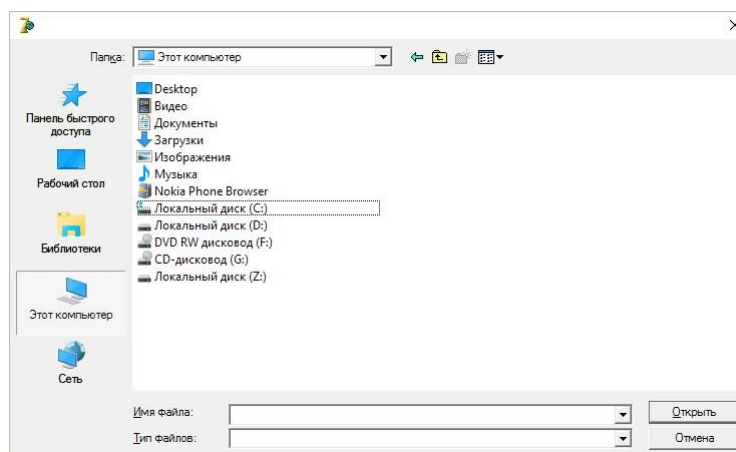
Dialogs sahifasi komponentalari Windowsning odatiy dasturlari muloqot oynalarini hosil qilishda foydalaniladi. Dialogs komponentalari fayllarni ko'rsatish yoki sozlashlarni o'rnatish uchun ishlatiladi. C++ Builderning Dialogs komponentalaridan foydalanish ilovalarni yaratishda vaqtini tejaydi va Windowsning odatdagi muloqot oynalaridan foydalanish imkonini yaratadi.



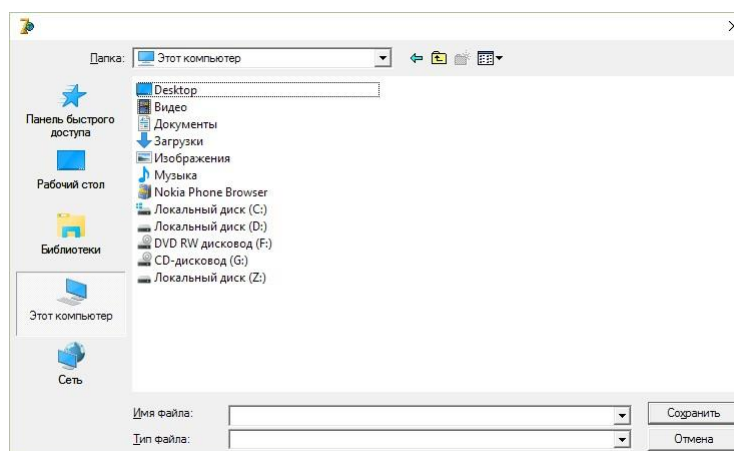
№	Komponenta	Tavsifi
1.	 OpenDialog	«Открыть файл» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
2.	 SaveDialog	«Сохранить файл как...» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
3.	 OpenPictureDialog	«Открыть рисунок» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
4.	 SavePictureDialog	«Сохранить рисунок как...» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
5.	 FontDialog	«Шрифты» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
6.	 ColorDialog	«Цвет» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
7.	 PrintDialog	«Печать» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
8.	 PrinterSetupDialog	«Установка принтера» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
9.	 FindDialog	«Найти» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
10	 ReplaceDialog	«Заменить» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega

.		emas.
11	 PageSetupDialog	«Параметры страницы» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.

OpenDialog – faylni ochish muloqot oynasi va **SaveDialog** faylni saqlash muloqot oynasi komponentalari deyarli barcha dasturlarda ishlatiladi. Muloqot oynalari quyidagi ko'rinishda bo'ladi.

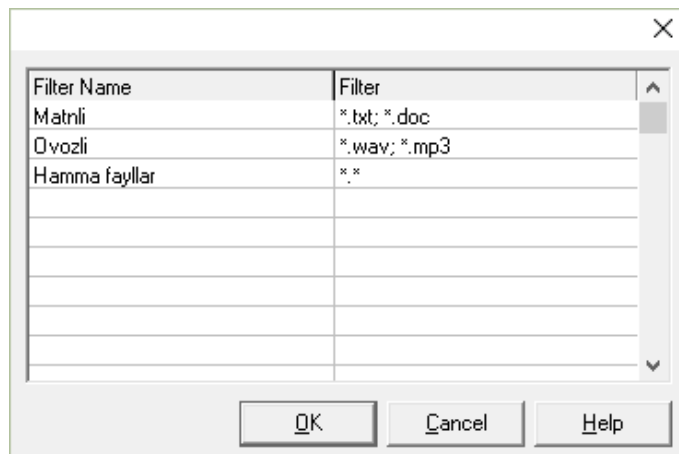


Faylni ochish muloqot oynasi.



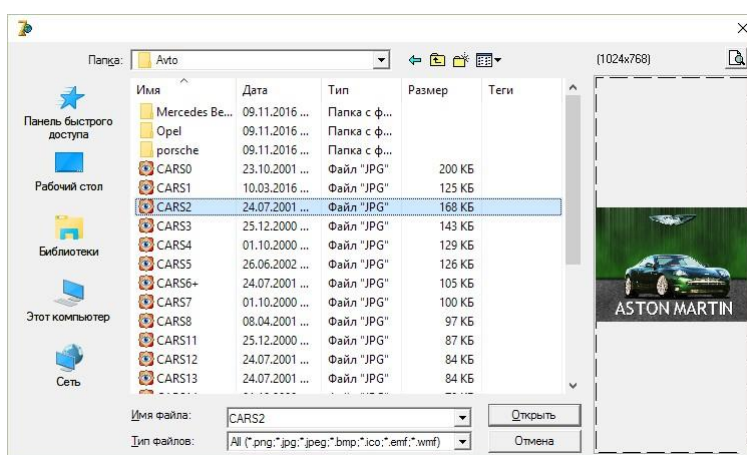
Faylni saqlash muloqot oynasi.

Ushbu komponentlarning barcha xususiyatlari bir xil, yagona farq ularning biri faylni ochishda so'ralsa, ikkinchisi faylni yopishda so'raladi. Bu komponentlarning eng asosiy xususiyati FileName (tanlangan faylning nomi) dir. Xususiyatning qiymatini muloqot oynasi ochilmasdan oldin ham ko'rsatish mumkin. Xususiyatga kiritilgan fayl nomi yuqoridagi muloqot oynalarining Имя файла bandidan paydo bo'ladi. Muloqot oynalarining Тип файла bo'limida fayllarning turini ko'rsatish uchun komponentlarning Filter xususiyati sozlanadi. Masalan,

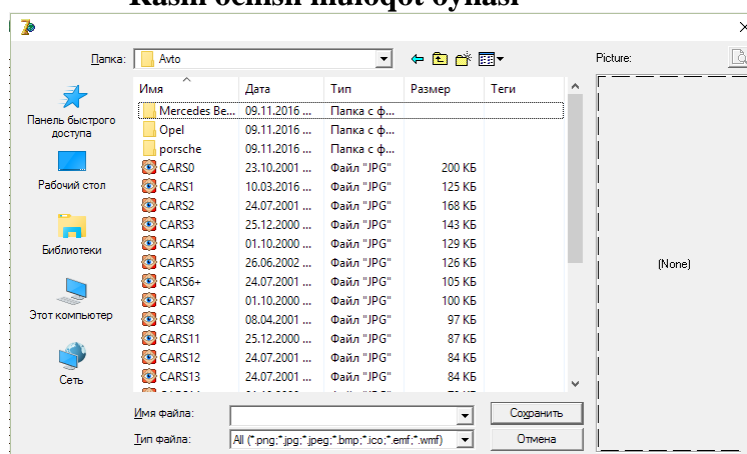


Title xususiyati orqali muloqot oynasiga nom berish mumkin.

C++ Builderning kutubxonasida maxsus **OpenPictureDialog** – rasm ochish muloqot oynasi va **SavePictureDialog** – rasmni saqlash muloqot oynasini hosil qiluvchi komponentalar mavjud.

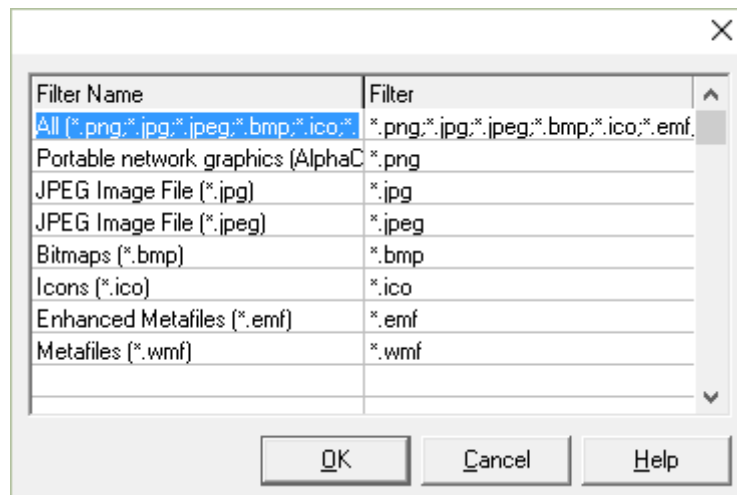


Rasm ochish muloqot oynasi



Rasmni saqlash muloqot oynasi

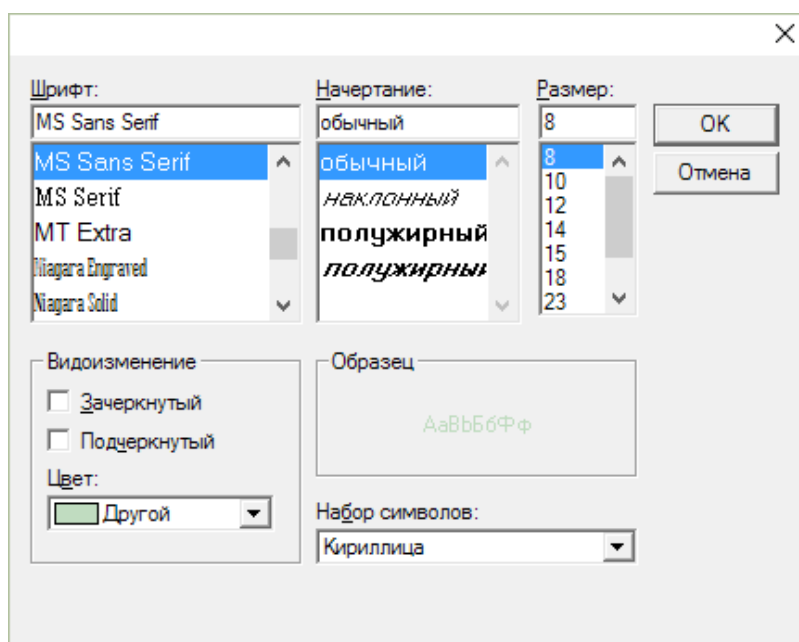
OpenPictureDialog va SavePictureDialog komponentalarning xususiyatlari OpenFileDialog va SaveDialog komponentalarining xususiyatlari bilan bir xil. Yagona farq shundaki OpenPictureDialog va SavePictureDialog komponentalarning Filter xususiyati oldindan sozlangan bo`ladi. Ya'ni,



Filter oynasi

Ushbu feltirda barcha grafik fayl turlari kiritilgan. Foydalanuvchi hohlasa ularni o`chirishi yoki qo`shimcha kiritishi mumkin.

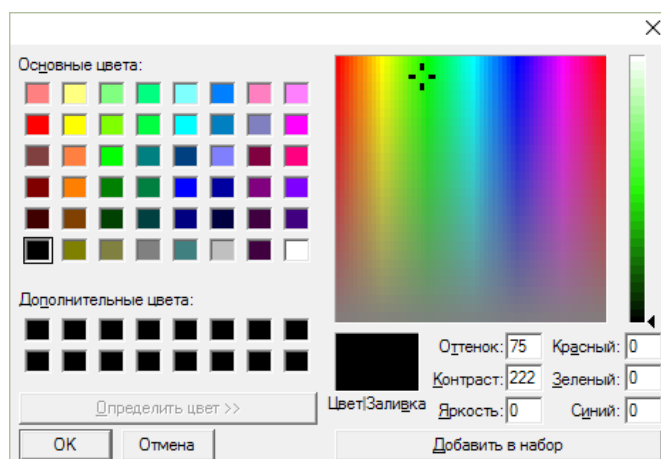
FontDialog komponentasi shrift xususiyatlarini sozlash muloqot oynasini chaqiradi. Ushbu oynada shrift nomi, yozilish shakli, o`lchami va boshqa xususiyatlarni sozlash mumkin.



Шrift xususiyatini sozlash oynasi

Komponentning asosiy xususiyati — Font. **MaxFontSize** va **MinFontSize** xususiyatlari shrift o`lchamining eng katta va eng kichik qiymatini o`rnatadi. Agar xususiyatlarning qiymati 0 (nol) ga teng bo`lsa (по умолчанию), o`lchamni kiritishda hech qanday cheklashlar kuzatilmaydi. Agarda xususiyatlarga qiymat kiritilsa, muloqot oynasining Размер bo`limida kiritilgan diapazondagi qiymatlar hosil bo`ladi. Agar diapazondan tashqaridagi qiymat kiritilsa, «Размер должен лежать в интервале ...» ogohlantirish oynasi hosil bo`ladi.

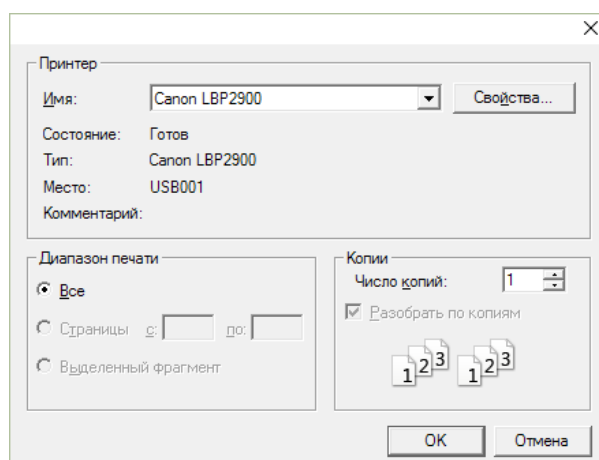
ColorDialog komponentasi rang tanlash muloqot oynasini chaqiradi. Bundan foydalanuvchi bazadagi ranglar palitrasidan yoki Определить цвет tugmasi orqali kerakli rangni tanlashi mumkin.



Rang tanlash oynasi

ColorDialog komponentasining asosiy xususiyati – Color. Bu xususiyat foydalanuvchi oynadan tanlagan rangni bildiradi.

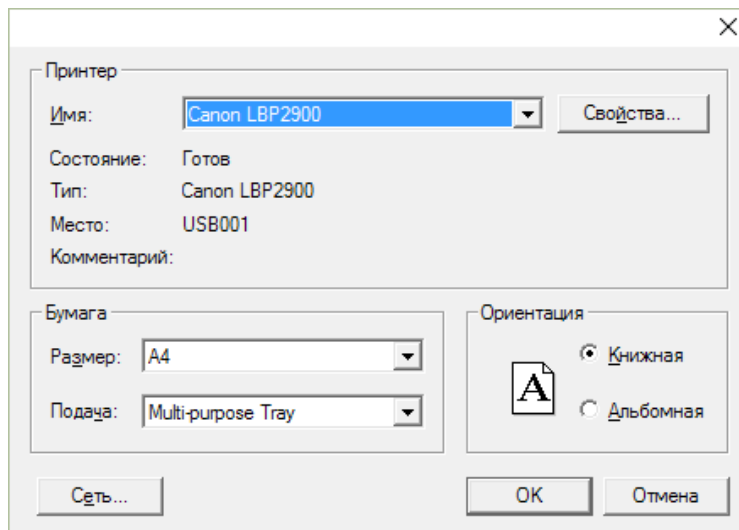
PrintDialog komponentasi chop qilish muloqot oynasini chaqiradi. Foydalanuvchi bunda printerni tanlashi va uning xususiyatini sozlashi, nusxalash sonini ko'rsatishi, sahifani yoki belgilangan fragmentni chop qilishi mumkin.



Chop qilish oynasi

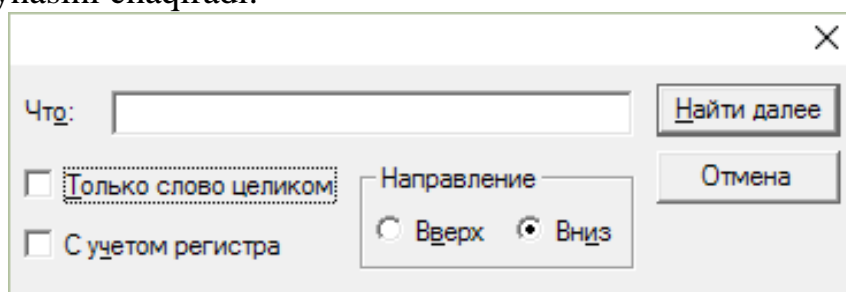
PrintDialog komponentasi chop qilishni amalga oshirmaydi. U faqat chop qilish xususiyatlarini sozlaydi. Chop qilish uchun esa **Printer** komponentasidan foydalaniladi.

PrinterSetupDialog komponentasi printer xususiyatlarini sozlash muloqot oynasini chaqiradi. Bu komponent o'rnatilishi yoki o'qilishi kerak bo'lgan o'ziga xos xususiyatlari yo'q yagona komponentdir. Dialog printerni sozlash amallarini bajaradi. Bu dialog hech qanday parametr qaytarmaydi.



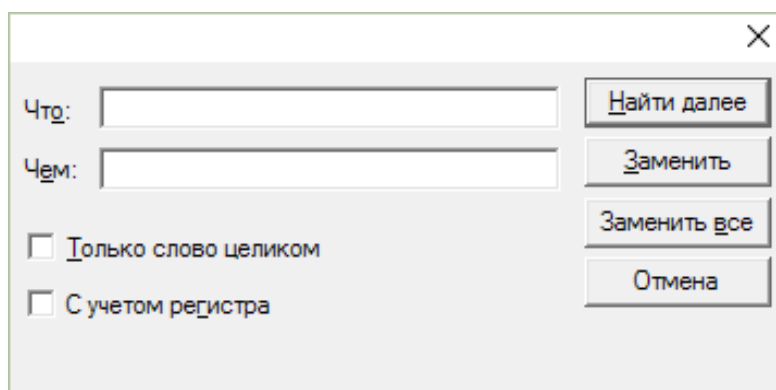
Printerni sozlash oynasi

FindDialog komponentasi satrning biror qismini qidirish (Найти) muloqot oynasini chaqiradi.



Satrni qidirish oynasi

ReplaceDialog komponentasi satrning biror qismini boshqasiga almashtirish (Замена) muloqot oynasini chaqiradi.



Satrni almashtirish oynasi

Ushbu komponentalar juda o`xshash va bir xil xususiyatlarga ega. Yagona farq

ReplaceDialog komponentasida satrni almashtirish mavjud.

Komponentlarning asosiy xususiyatlari:

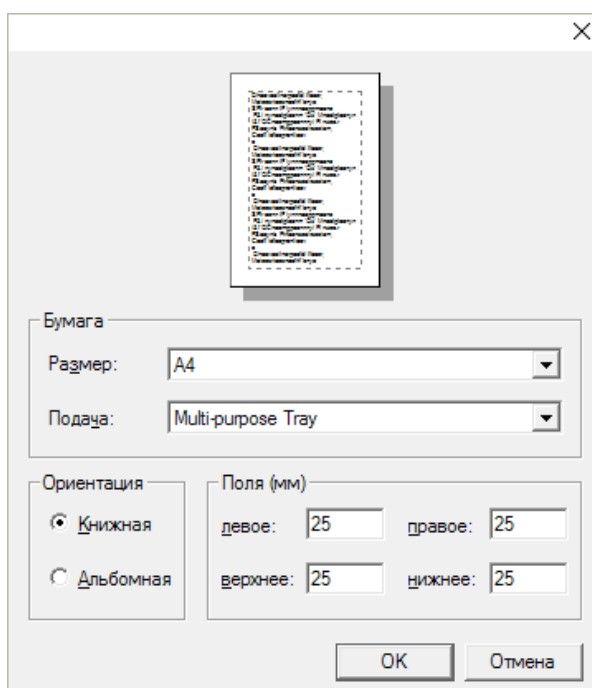
FindText – foydalanuvchi tomonidan kiritilgan qidiriladigan yoki almashtiriladigan satr. **ReplaceText** – **ReplaceDialog** komponentasiga tegishli.

FindText ga almashtirilishi kerak bo'lgan satr.

Options – ko'plab xususiyatlar.

FindDialog va **ReplaceDialog** komponentalari o'zlari qidiruv yoki almashtirishni amalga oshirmaydi. Ular faqat foydalanuvchi interfeysini sozlaydi. Qidirish yoki almashtirishni amalga oshirish uchun kerakli buyruqlarni kiritish talab qilinadi. Muloqot oynasidagi Найти далее tugmasi ishlashi uchun **OnFind** hodisasiga, Заменить yoki Заменить все tugmalari uchun esa **OnReplace** hodisasiga foydalanuvchi tegishli buyruqlarni kiritishi lozim.

PageSetupDialog komponentasi sahifa parametrlarini sozlash muloqot oynasini chaqiradi. Oynada sahifa o'lchamini (formati, odatda A4), sahifa holatini (ориентация, odatda Книжная), sahifa chetki chegaralarini sozlash amalga oshiriladi.



Sahifa parametrlarini sozlash oynasi

Komponentaning asosiy xususiyatlari:

MarginBottom, **MarginTop**, **MarginLeft**, **MarginRight** – sahifaning mos ravishta yuqori, pastki, chap tomon va o'ng tomon chetki chegaralarini sozlash.

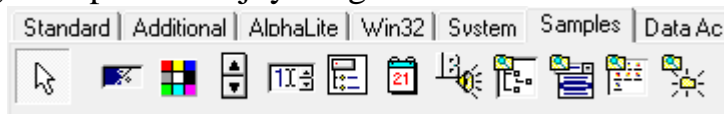
PageHeight – sahifaning balandligini kiritish.

PageWidth – sahifaning kengligi kiritish.

Mavzu: Borland C++ Builder Samples komponentlar palitrasi.

C++ Builder dasturlash muhiting Samples komponentalar palitrasi bilan tanishib chiqamiz.

Palitrada quyidagi komponentlar joylashgan:



Samples komponentalar palitrasi



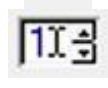
Gauge – jarayonlarni vizual tasvirlash indikator. Masalan, fayllarni nusxalash yoki dasturlarni kompyuterga o`rnatilishi jarayoni.



ColorGrid – foydalanish mumkin bo`lgan ranglar jadvalini hosil qiladi.



SpinButton – schyotchikli tugma hosil qiladi. Qiymatlarini orttiruvchi va kamaytiruvchi tugmalari mavjud.



SpinEdit – qiymatlarini orttiruvchi va kamaytiruvchi tugmalari mavjud bo`lgan tahrirlash oynasi.



DirectoryOutline – diskdagi kataloglar strukturasini hosil qiladi.



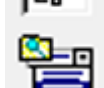
Cal}ar – standart ko`rinishdagi ko`rsatilgan oy bo`yicha kalendar yaratadi



IBEventAlerter – hodisa indikator hosil qiladi. Ma'lumotlar bazasi hodisalari bo`yicha habar beradi.



ShellTreeView – kataloglar daraxtini hosil qiladi.



ShellComboBox – tanlash mumkin bo`lgan kataloglar ro`yxatini hosil qiladi.



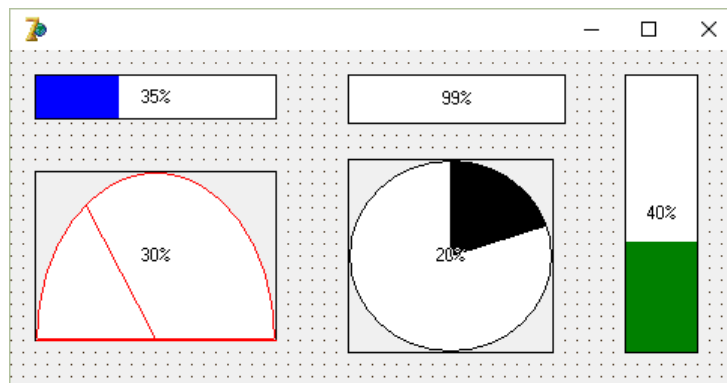
ShellListView – katalog va fayllar ro`yxatini hosil qiladi.



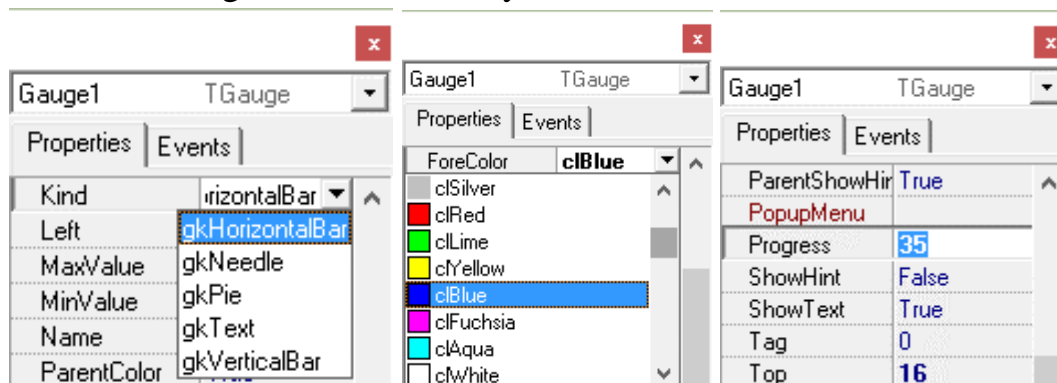
ShellChangeNotifier – joriy katalogni belgilash.

Borland C++ Builder 6 dasturlash muhitida Gauge komponentasidan foydalanish

Ushbu bo`limda Gauge komponentasi haqida ma'lumot beramiz. Gauge komponentasi jarayon bosqichlarini vizual tasvirlash uchun foydalaniladi. C++ Builder dasturini ishga tushiramiz. Forma oynasiga Samples palitrasidan Gauge komponentasini joylashtiramiz. Gauge komponentasining gorizontali, sektorli, doirali, matnli va vertikal ko`rinishlari mavjud.



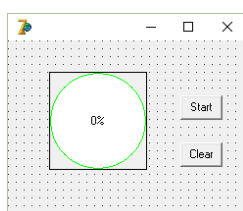
Komponenta koʻrinishi oʻzgartirish uchun Object Inspector oynasini Kind boʻlimidan, rangini oʻzgartirish uchun ForeColor boʻlimidan, jarayon bosqichini oʻzgartirish uchun esa Progress boʻlimdan foydalaniladi:



Komponentaning asosiy xususiyatlari

Xusiyat	Izo
MaxValue	Pozitsiyaning maksimal qiymati. Odatda qiymat 100 boʻladi
MinValue	Pozitsiyaning minimal qiymati. Odatda qiymat 0 boʻladi
Progress	Jarayonning ixtiyoriy pozitsiyani koʻrsatish.
BackColor	Komponentning fon rangi
ForeColor	Jarayonni toʻldirish rangi
Kind	Diagramma turi.

Misol. Forma oynasiga ikki button va Gauge komponentasini joylashtiramiz. Gaugening diagramma turini doirali holatda oʻtkazamiz. Rangini esa yashilga oʻzgartiramiz. Birinchi buttonni “Start” deb, ikkinchi buttonni “Clear” deb nomlaymiz. Quyidagicha forma oynasi hosil boʻladi:



Start tugmasining dasturlash qatoriga quyidagi

```

buyruqlarni kiritamiz: procedure TForm1-
>Button1Click(Sender: TObject);
i:int;
{
for i=1 to 100 do {
gauge1-
>Progress=gauge1-
>Progress+1; sleep(100);
};
};

```

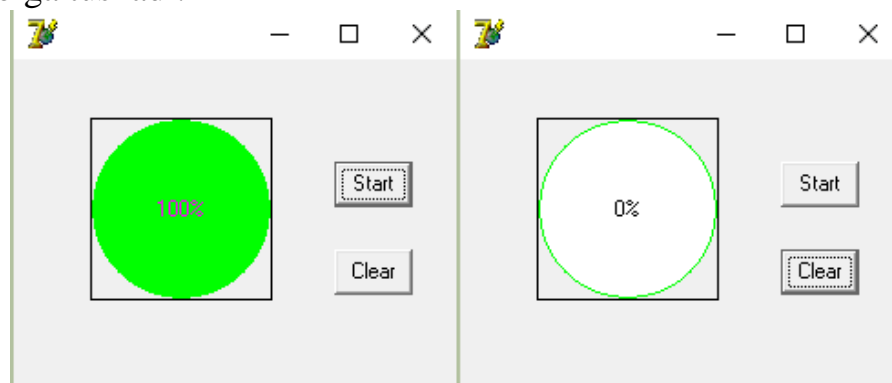
Clear tugmasining dasturlash qatoriga quyidagi buyruqlarni kiritamiz: procedure TForm1->Button2Click(Sender: TObject);

```

{
gauge1->Progress=0;
};

```

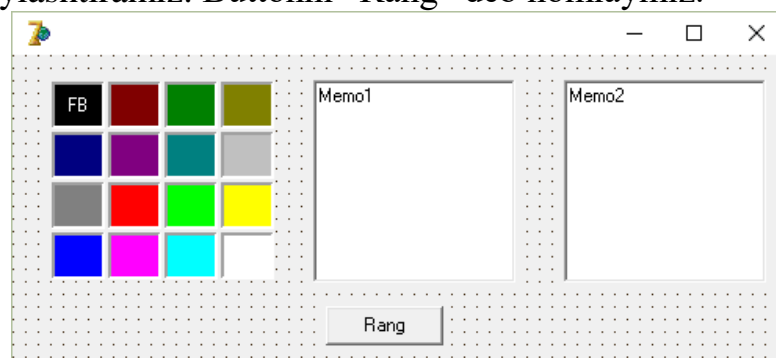
Loyihani kompyuter xotirasiga saqlaymiz va F9 tugmasini bosamiz. Start tugmasi bosilsa, diagramma to'lib boradi. Clear tugmasi bosilsa esa, pozitsiya nolga tushadi.



Borland C++ Builder 6 dasturlash muhitida ColorGrid komponentasidan foydalanish

Ushbu bo'limda ColorGrid komponentasi haqida ma'lumot beramiz. ColorGrid komponentasi foydalanish mumkin bo'lgan 16 ta rangli jadvalni hosil qiladi. Jadvalda ikki xil rangni FG - ForegroundColor (oldingi rang) va BG - BackgroundColor (fon rangi) ni tanlash mumkin.

Misol. Forma oynasiga bitta Button, bitta ColorGrid, ikkita Memo komponentasini joylashtiramiz. Buttonni "Rang" deb nomlaymiz.



Rang tugmasining dasturlash qatoriga quyidagi buyruqlarni kiritamiz:

```
procedure TForm1->Button1Click(S)er: TObject);  
{  
Memo1->Color=colorgrid1-  
>ForegroundColor; Memo2-  
>Color=colorgrid1-  
>BackgroundColor;  
};
```

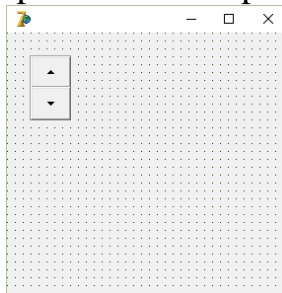
Loyihani kompyuter xotirasiga saqlaymiz va F9 tugmasini bosamiz. Rang tugmasi bosilsa ColorGridning FG yacheykasidagi rang Memo1 ning rangiga, BG yacheykasidagi rang Memo 2 ning rangiga o`zgaradi.



Borland C++ Builder 6 dasturlash muhitida SpinButton komponentasidan foydalanish

Ushbu bo`limda SpinButton komponentasi haqida ma'lumot beramiz. SpinButton komponentasi sanagichli tugma hosil qiladi. Komponentda ikkita Up va Down tugmalari mavjud.

Misol. Forma oynasiga bitta SpinButton komponentasini joylashtiramiz.



SpinButtonning Up tugmasi dasturlash qatoriga

```
quyidagilarni kiritamiz: procedure TForm1-  
>SpinButton1UpClick(S)er: TObject);  
{  
Form1-  
>Height=Form1-  
>Height+5; Form1-  
>Width=Form1-  
>Width+5;  
};
```

SpinButtonning Down tugmasi dasturlash qatoriga

```
quyidagilarni kiritamiz: procedure TForm1-  
>SpinButton1DownClick(S)er: TObject);
```

```

{
Form1-
>Height=Form1-
>Height-5; Form1-
>Width=Form1-
>Width-5;
};

```

Loyihani kompyuter xotirasiga saqlaymiz va F9 tugmasini bosamiz. Up tugmasi bosilsa forma oynasi balandligi va kengligi ortadi. Down tugmasi bosilsa, forma oynasi balandligi va kengligi kamayadi.

Borland C++ Builder 6 dasturlash muhitida SpinEdit komponentasidan foydalanish

Ushbu bo`limda SpinEdit komponentasi haqida ma'lumot beramiz. SpinEdit komponentasi tahrirlash maydoni bo`lgan sanagichli tugma hosil qiladi. Komponentda ikkita Up va Down tugmalari hamda Editga o`xshash maydoni mavjud. SpenEdit komponentasini SpinButton va Edit komponentalarining birlashmasi deyish mumkin.

Misol. Forma oynasiga ikkita SpinEdit va bitta StringGrid komponentalarini joylashtiramiz. Birinchi SpinEdit komponentasining dasturlash qatoriga quyidagilarni kiritamiz:

```

procedure TForm1->SpinEdit1Change(Sender: TObject);
{
Stringgrid1->ColCount=SpinEdit1->Value;
};

```

Ikkinchi SpinEdit komponentasining dasturlash qatoriga

```

quyidagilarni kiritamiz: procedure TForm1-
>SpinEdit2Change(Sender: TObject);
{
Stringgrid1->RowCount=SpinEdit2->Value;
};

```

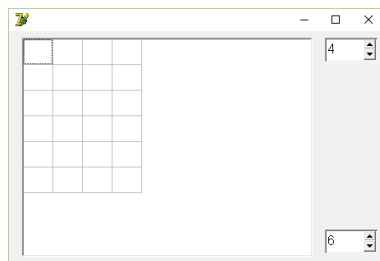
Form oynasining dasturlash qatoriga

```

quyidagilarni kiritamiz: procedure
TForm1->FormCreate(Sender:
TObject);
{
Stringgrid1-
>ColCount=SpinEdit1-
>Value; Stringgrid1-
>RowCount=SpinEdit2-
>Value;
};

```

Loyihani kompyuter xotirasiga saqlaymiz va F9 tugmasini bosamiz. SpinEdit1 tugmasiga mos holda StringGridning ustunlar soni o`zgaradi. SpinEdit2 tugmasiga mos holda StringGridning satrlar soni o`zgaradi.



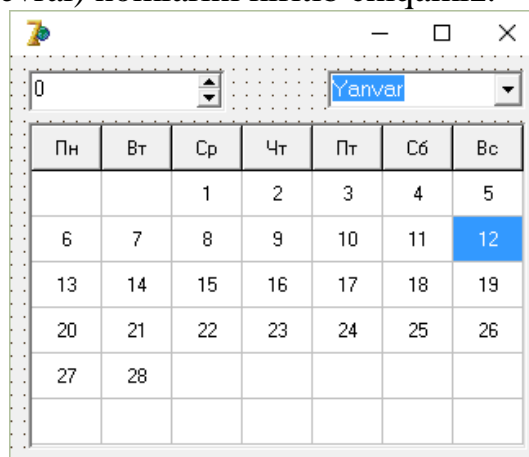
Borland C++ Builder 6 dasturlash muhitida Cal}ar komponentasidan foydalanish

Ushbu bo`limda Cal}ar komponentasi to`g`risida ma`lumot beramiz. Cal}ar komponentasi ko`rsatilgan oydagi standart ko`rinishdagi bir oyli kalendar yaratadi. Kalendardagi yil, oy va kunni aniqlash uchun Year, Month va Day xususiyatlari mavjud. Bu xususiyatlar butun sonlarni qabul qiladi. Oynada tasvirlashdan oldin yoki loyihalash jarayonida Month va Year xususiyatlariga qiymat kiritiladi. Natijada ko`rsatilgan oy va yildagi cal}ar hosil bo`ladi.

Komponentning asosiy xususiyatlar:

Xususiyat	Izo
Color	kalendar fon rangi
Day	joriy kunni kiritish
Month	joriy oyni kiritish
StartOfWeek	hafta boshini tanlash. (odatda 0 - воскресенье)
UseCurrentDate	joriy (bugungi) sanani tanlash
Year	joriy yilni kiritish

Misol. C++ Builder dasturida kalendar tayyorlash. Buning uchun forma oynasiga bitta Cal}ar, bitta SpinEdit va Bitta ComboBox komponentalarini joylashtiramiz. ComboBox1 komponentasining Items bo`limiga 12 ta oy (Yan-Fevral) nomlarini kiritib chiqamiz.



SpinEdit1 komponentasining dasturlash qatoriga quyidagilarni kiritamiz:

```
procedure TForm1->SpinEdit1Change(Sender: TObject);
{
```



```
Cal}ar1->Year=SpinEdit1->Value;
};
```

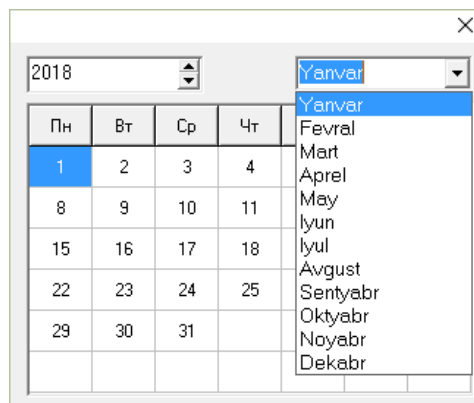
ComboBox1 komponentasining dasturlash qatoriga quyidagilarni kiritamiz: procedure TForm1->ComboBox1Change(S}er: TObject);

```
{
Cal}ar1->Month=ComboBox1->ItemIndex+1;
};
```

Form oynasining dasturlash qatoriga quyidagilarni kiritamiz: procedure TForm1->FormCreate(S}er: TObject);

```
{
ComboBox1-
>ItemIndex=Cal}ar1->Month-
1; SpinEdit1->Value=Cal}ar1-
>Year;
};
```

Loyihani kompyuter xotirasiga saqlaymiz va F9 tugmasini bosamiz. Natijada kalendar paydo bo`ladi. ComboBox1 komponentasining ro`yxatidan joriy oyni, SpinEdit komponentasi orqali esa kerakli yilni tanlash mumkin.



Borland C++ Builder 6 dasturlash muhitida ShellTreeView, ShellComboBox va ShellListView komponentalaridan foydalanib “Проводник” dasturini tayyorlash.

Ushbu bo`limda ShellTreeView, ShellComboBox, ShellListView komponentalari to`g`risida ma`lumot beramiz.

C++ Builder dasturining ShellTreeView komponentasi kataloglar daraxtini hosil qiladi. ShellComboBox komponentasi kataloglar ro`yxatini hosil qiladi. ShellListView komponentasi katalog tarkibidagi obyektlarni ko`rsatuvchi ro`yxat hosil qiladi.

Misol. Forma oynasiga ShellTreeView, ShellComboBox, ShellListView va to`rtta SpeedButton komponentalarini joylashtiramiz. Button tugmalarini Icon, List, Report va SmallIcon deb nomlaymiz.

ShellTreeView komponentasining ShellComboBox xususiyatiga ShellComboBox1 ni, ShellListView xususiyatiga esa ShellListView1 ni, Root xususiyatiga rfDesktop ni kiritamiz.

ShellComboBox komponentasining ShellTreeView xususiyatiga ShellTreeView1 ni, ShellListView xususiyatiga esa ShellListView1 ni kiritamiz.

ShellListView komponentasining ShellTreeView xususiyatiga ShellTreeView1 ni, ShellComboBox xususiyatiga ShellComboBox1 ni kiritamiz.

Natijada ushbu komponentalar o`zaro bog`lanadi.

Icon tugmasining dasturlash qatoriga quyidagilarni kiritamiz: procedure TForm1->SpeedButton1Click(S)er: TObject);

```
{  
  ShellListView1->ViewStyle=vsIcon;  
};
```

List tugmasining dasturlash qatoriga

quyidagilarni kiritamiz: procedure

TForm1->SpeedButton2Click(S)er: TObject);

```
{  
  ShellListView1->ViewStyle=vsList;  
};
```

Report tugmasining dasturlash qatoriga

quyidagilarni kiritamiz: procedure

TForm1->SpeedButton3Click(S)er: TObject);

```
{  
  ShellListView1->ViewStyle=vsReport;  
};
```

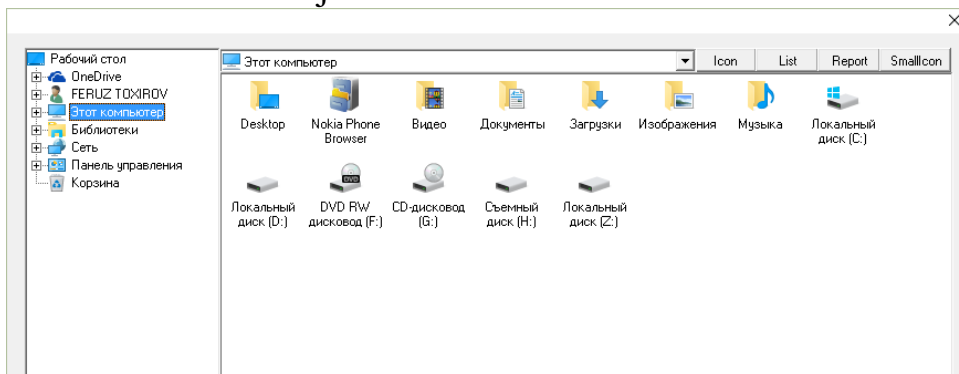
SmallIcon tugmasining dasturlash qatoriga

quyidagilarni kiritamiz: procedure TForm1-

>SpeedButton4Click(S)er: TObject);

```
{  
  ShellListView1->ViewStyle=vsSmallIcon;  
};
```

Loyihani kompyuter xotirasiga saqlaymiz va F9 tugmasini bosamiz. Natijada mini provodnik dasturi ishga tushadi. ShellTreeView yoki ShellComboBox komponentalaridan kerakli katalogni tanlaymiz. Tanlangan katalog tarkibi ShellListView komponentasida namoyon bo`ladi. Icon, List, Report va SmallIcon tugmalari orqali esa ShellListView komponentasi tarkibi ko`rinishini o`zgartirish mumkin ya'ni katta ikonkali, ro`yxatli, fayl haqidagi ma'lumotlar bilan va kichik ikonkali ko`rinishlari mavjud.



Mavzu: Borland C++ Builderning grafik vositalari.

Reja:

1. C++ Builder muhitida grafikani qo'llab-quvvatlash
2. Sodda geometrik figuralarni hosil qilish
3. C++ Builder muhitida grafik shakllarni chizish
4. C++ Builder muhitida matematik funksiyalar grafiklarini hosil qilish

C++ Builder da grafikani qo'llab-quvvatlash

C++ Builder dasturi Windows GDI funksiyalarini turli darajalarda inkapsulalaydi. Bu o'rinda bir usul muhim bo'lib, uning vositasida grafik komponentalar o'z tasvirlarini monitor ekranida taqdim etadi. GDI funktsiyasi to'g'ridan-to'g'ri chaqirilganda, ushbu grafik komponentalarga *qurilma konteksti deskriptori (device context handle)* ni o'zlash kerak. Bu deskriptor siz tanlab olgan rassomchilik ashyolari - perolar, mo'yqalamlar, shriftlarni chiqarib beradi. Grafik tasvirlar bilan ishlash tugagach, siz qurilma kontekstini dastlabki holatga keltirib qo'yishga majbursiz va shundan keyingina undan ozod bo'lishingiz mumkin.

Shu darajada detallashtirilgan grafika bilan ishlashga sizni majbur qilish o'rniga, C++ Builder grafik komponentalarning Canvas (Canva - Asos) xususiyati vositasida sodda va tugal interfeysni taklif qiladi. Bu xususiyat qurilmaning to'g'ri kontekstini nomlaydi (initsiallashtiradi) hamda siz rasm chizishni to'xtatgan kerakli vaqtda uni ozod qiladi. Asos pero, mo'yqalam va shrift tavsiflari nomidan ish ko'radigan berilgan xususiyatlarga ega.

Grafik komponentalar bilash ishlashda foydalanuvchi amalga oshirishi lozim bo'lgan yagona ish - bu qo'llanayotgan rasm chizish ashyolarining tavsiflarini aniqlash. Ashyolarni yaratish, tanlash va ozod qilishda sizdan tizim zahiralarini ko'zatiib borish talab qilinmaydi. Asosning o'zi bu haqda qayg'uradi.

Grafika bilan ishlashda C++ Builder namoyon qiladigan afzalliklardan biri - bu tizimning grafik zahiralarini uchun keshlangan xotiradan foydalanish. Aytaylik, agar sizning dasturingiz biron- bir konkret turdagi peroni qaytadan yaratsa, qo'llasa va ozod etsa, siz ushbu perodan har gal foydalanganingizda bu qadamlarni takrorlashingizga to'g'ri keladi. C++ Builder kesh-xotiradan grafik zahiralarini saqlash uchun foydalanar ekan, tez-tez qo'llanadigan rasm chizish ashyosi har gal yangitdan qayta yaratilmay, balki kesh-xotiradan takroran tanlab olinishi ehtimoli oshadi. Buning natijasida sizning grafik ilovangizning takrorlanayotgan operatsiyalarining samarasi ancha ortishi aniq.

Qo'yida Windows uchun ilovalar darchasida ko'k kontur bilan aylantirilgan sariq ellipsni chizish masalasini hal qilayotgan kod fragmenti keltirilgan. Bu masala rasm chizish asosi vositasida yechiladi.

Palitrage xizmat ko'rsatish. Foydalanuvchilik interfeysining ko'pchilik elementlari biron- bir palitrage muhtojlik sezmaydi. Biroq, grafik tasvirlarga ega komponentalarga, komponentalar ma'lumotlarini tegishli tarzda aks ettirish uchun, Windows hamda uning ekran drayveri bilan o'zaro aloqaga kirishish zarur bo'lib qolishi mumkin. Windows operatsiya tizimiga oid hujjatlarda bu jarayon *palitralarni ishga tushirish (palett realizing)* deb ataladi. Palitrani ishga tushirish operatsiyasining vazifasi shundan iboratki, u eng ustki (ekranda sizga nisbatan eng yaqin turgan) faol darcha to'liq rang palitrasidan foydalanishini, fon darchalari esa o'z palitralarining

qolgan ranglaridan maksimal darajada foydalanishlarini ta'minlashi kerak. Bu degani, fon darchalari o'z ranglarini «real» palitradagi erishish mumkin bo'lgan eng yaqin ranglarga o'zgartira olishlari kerak. Darchalar bir-birini qisman yopib joy almashar ekan, Windows ham muttasil darcha palitralarini ishga solib boradi.

Mulohaza. C++ Builder bit obrazlari palitralaridan boshqa palitralarni yaratish va ularga xizmat ko'rsatish uchun mustaqil vositalarga ega emas. Biroq, agar siz biron-bir palitraning deskriptorini olgan bo'lsangiz, grafik komponentalar ular bilan ishlay oladi.

Display yoki printer turidagi qurilmalar bilan ishlashda C++ Builder komponentalari avtomatik tarzda palitralarni ishga tushirish mexanizmini qo'llab-quvatlaydi. Shunday qilib, siz TControl tayanch (bazaviy) komponentli sinfdan meros qilib olingan ikkita GetPalette va PaletteShanged metodlaridan foydalanishingiz mumkin. Bunda Windows bu palitraga qaynday munosabatda bo'lsa, siz ham uni xuddi shunday ishlata olasiz:

Palitraning komponenta bilan aloqasi. Agar grafik komponenta uchun biron-bir palitradan foydalanish zarurati tug'ilgan bo'lsa, sizning ilovangiz bu xaqda xabardor bo'lishi kerak. Palitrani komponentangizga o'xshatish uchun, uning GetPalette ob'ektlari metodini shunday ortiqcha yuklatingki, u ushbu palitra *deskriptori* (*handle*)ni qaytarsin. Shuning bilan birga siz, birinchidan, komponentangizning ma'lum bir palitrasi ishga tushishi lozimligini ilovangizga ma'lum qilasiz, ikkinchidan, ishga tushishda qaysi palitra konkret qo'llanishi kerakligini aniqlaysiz.

Palitra o'zgarishiga reaksiya (munosabat). Sizning komponentangiz GetPalette metodini ortiqcha yuklatish vositasida qandaydir palitra bilan o'xshatilgan bo'lsa, C++ Builder tizimi Palette Shanged metodi yordamida Windows ning palitralardan xabarlariga munosabat bildirishni avtomatik tarzda o'z zimmasiga oladi. Normal ish sharoitida siz hech qachon yashirin belgilangan bu metodning xulq-atvorini qayta aniqlash zaruratiga duch kelmaysiz. Palette Shanged metodining asosiy vazifasi palitrani ishga tushirish turini (fonli yoki faol darchalar uchun) aniqlashdan iborat. Palitralarning Windows tizimida ishga tushirilishiga nisbatan C++ Builder bir qadam ilgari ketdi: darcha deskriptorlari yordamida, nafaqat bir-birining ustiga taxlanadigan


«dasta» palitrasi, balki faol darchaning bir-birining ustiga taxlangan komponentalarining palitralari ham ishga tushiriladi. Agar xohlasangiz, siz palitralarning yashirin qabul qilingan bunday xulq-atvorini qayta aniqlashingiz va natijada biron-bir komponenta to'liq rang palitrasiga ega bo'lishi hamda ekranda sizga eng yaqin turgan komponentadek ko'rinishiga erishishingiz mumkin.

Ekrandan tashqaridagi bit obrazlari. Windows uchun murakkab grafik ilovalarni dasturlashning umum qabul qilingan metodikasi shundan iboratki, bunda ekrandan tashqari bit obrazi yaratiladi, bu obrazga konkret tasvir tushiriladi yoki to'ldiriladi va, nihoyat, yaratilgan tasvir to'laligicha bit obrazidan ekran darchasining ko'rsatilgan joyiga nusxa ko'chirib olinadi. Shu tufayli ekran darchasida bevosita takroran rasm chizish keltirib chiqaradigan va ko'zni charchatadigan monitor ekranidagi lipillashlar kamayadi.

C++ Builder sizning ilovangizda Tbitmap sinfi ob'ektlarini yaratish imkonini beradiki, bu ekrandan tashqari tasvirlar sifatida ishlay oladigan fayl va boshqa zahiralar tasvirlarini ham sizning ilovangizda taqdim etish uchun qilinadi.

C++ Builder muhitida grafik shakllarni chizish


Delphi dasturlash tizimining vizual komponentlar kutubxonasi bir qancha grafiklar hosil qilish imkoniyatini beruvchi ob'ektlarga egadir. Bulardan ayrimlari: **TImage**, **TShape** va **TBevel** ob'ektlari.

 - **Image** – bu ob'ekt formaning ixtiyoriy joyiga grafik tasvirni joylashtirish imkonini beradi. Bunda .bmp, .ico, .wmf formatidagi rasmlar tasvirlanadi. Ob'ekt **Additional** komponentlar palitrasida joylashgan. Rasmlarni ob'ektlar inspektori xususiyatidagi **Picture** qatori orqali qo'yish mumkin. Agar bu rasmlar formaga joylashtirilgan bo'lsa, u holda .exe kengaytmali fayli katta hajmga ega bo'lishi mumkin. Rasmlarni dastur bajarilishi davomida ham chiqarish mumkin. Buning uchun quyidagi qatorlar yozilishi kerak, ya'ni: **If OpenFileDialog1. Execute then Image1. Picture.LoadFromFile (OpenDialog1. File Name);**



Rasm o'rnatish

Bu ob'ektning ob'ektlar inspektori xususiyatidagi **Center** va **Stretch** qatorlari eng muhim ahamiyatga ega bo'lib, ikkala qator ham **Boolean** tipiga mansubdir. Agar **Center** qatorida true bo'lsa, u holda rasm formadagi **TImage** ob'ektining o'rtasida joylashadi. Agar **Stretch** qatorida true bo'lsa, u holda rasm formadagi **TImage** ob'ekti o'lchamini butunligicha egallab turadi.

 - **Shape** - oddiy grafik tasvirlarni formada hosil qilish uchun ishlatiladi, ya'ni doira, kvadrat, to'rtburchak va shunga o'xshash geometrik shakllar. Ob'ekt xususiyatidagi **Pen** qatori orqali chizma chiziqlarining rangi va ko'rinishini ko'rsatish mumkin. **Brush** qatori orqali chizma ichining rangi ko'rsatiladi. Xususiyatdagi bu amallarni dasturni bajarish davomida ham amalga oshirish mumkin.

Delphi dasturlash tilida dasturchiga grafik dasturlar sxema va chizma, illyustratsiyalar yaratishga imkon beradi. Dastur grafikani ob'ekt (forma yoki Image komponentasi) yuzasiga chiqaradi. Ob'ekt yuzasiga **Canvas** xossasi mos keladi. Ob'ekt yuzasiga grafik element (to'g'ri chiziq, aylana, turtburchak va hokazo), chiqarish uchun bu ob'ektning **Canvas** xossasiga mos usul qo'llash lozim. Misol uchun **Form1.canvas. Rectangle(10,10,100,100)** instruktsiyasi dastur oynasida to'rtburchak chizadi.

Chizish sohasi

Yuqorida ko'rilgan **Canvas** xossasi –**Tcanvas** tipidagi ob'ektdir. Grafik primitivlarni chiqarish usullari Canvas xossasini abstrakt chizish sohasi deb qaraydi. Chizish sohasi alohida nuqtalar – piksellardan iborat. Piksel holati uning gorizontal (X) va vertikal (Y) koordinatalari bilan aniqlanadi. Chap yuqori piksel koordinatalari (0,0). Koordinatalar yuqoridan pastga va chapdan o'ngga qarab o'sib boradi. Soha o'lchovlarini image komponentasining **Height** va **Width** xossalari va formaning **ClientHeight** va **Clientwidth** xossalari orqali aniqlash mumkin.

Qalam

Qalam geometrik figuralarni chizish uchun ishlatiladi. Chiziq ko‘rinishi **Tren** ob’ektni quyidagi jadvalda ko‘rsatilgan xossalari orqali aniqlanadi.

Tren (qalam) xossalari.

Xossa	Ta’rifi
Color	Chiziq rangi
Width	Chiziq qalinligi
Style	Chiziq ko‘rinishi
Mode	Akslantirish rejimi

Chiziq qalinligi width xossasi orqali piksellarda beriladi. Chiziq turini style xossasi belgilaydi. Quyidagi jadvalda chiziq turini belgilovchi nomlangan konstantalar sanab o‘tilgan.

Style xossasi qiymatlari

Константа	Chiziq ko‘rinishi
psSolid	Uzluksiz chiziq
psDash	Punktir chiziq, uzun shtixlar
psDot	Punktir chiziq, qisqa shtixlar
psDashDot	Punktir chiziq, uzun va qisqa chiziqlar ketma-ketligi
psDashDotDot	Punktir chiziq, bitta uzun va ikkita qisqa shtixlar ketma-ketligi
psClear	Chiziq aks ettirish

Mode xossasi chiziq rangining fon rangiga munosabatini ko‘rsatadi. Odatda chiziq rangi

Pen.Color xossasi qiymati bilan belgilanadi.

Dasturchi chiziq uchun fon rangiga nisbatan invers rang berishi mumkin. Bu holda hatto chiziq va fon rangi bir xil berilgan bo‘lsa ham chiziq ajralib turadi.

Quyidagi jadvalda **Mode** xossasi qiymati sifatida ishlatish mumkin bo‘lgan konstantalar berilgan.

Mode xossasi qiymatlari

Konstanta	Chiziq rangi
pmBlack	Qora, Pen. Color xossasi qiymatiga bog‘liq emas
pmWhite	Oq, Pen. Color xossasi qiymatiga bog‘liq emas
pmCopy	Chiziq rangi Pen. Color xossasi qiymatiga bog‘liq
pmNotCopy	Chiziq rangi Pen. Color xossasi qiymatiga invers
pmNot	Chiziq rang sohasining mos nuqtasi rangiga invers

Mo‘yqalam

Mo‘yqalam (Canvas.Brush) yopiq sohalarni chizish va soha ichini bo‘yash uchun mo‘ljallangan usullardan foydalaniladi. Mo‘yqalam ob’ekt jadvalda ko‘rsatilgan ikki xossaga ega.

Tbrush (muyqalam) xossalari

Xossa	Ta'rifi
Color	Yopiq sohani bo'yash rangi
Style	Sohani to'ldirish uslubi

Kontur ichidagi soha bo'yalishi yoki shtrixlanishi mumkin. Sohani to'ldirish usulini belgilovchi konstantalar quyidagi jadvalda berilgan.

Brush.style xossasi qiymatlari

Konstanta	Soha bo'yash uslubi
bsSolid	Uzluksiz bo'yash
bsClear	Sohani bo'yalmaydi
bsHorizontal	Gorizantal shtrixlash
bsVertical	Vertikal shtrixlash
bsFDiagonal	Diogonal shtrixlash, oldinga og'ish
bsBDiagonal	Diogonal shtrixlash, orqaga og'ish
bsCross	Katakli gorizantal-vertikal shtrixlash
bsDiagCross	Katakli diogonal shtrixlash

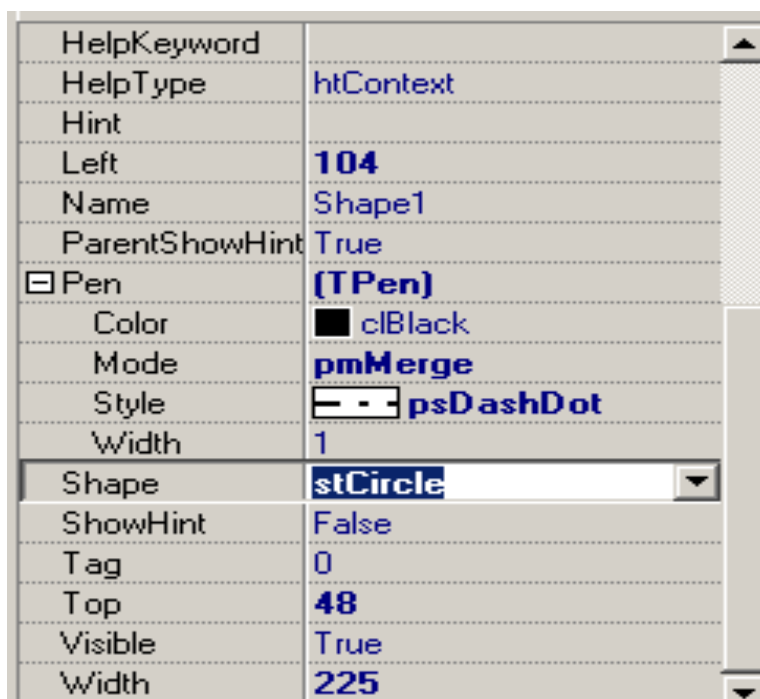
Yuqorida aytilganlardan foydalangan holda bir nechta misollar ko'rib o'tamiz:

1-misol. Shape- tugmasidan foydalanib aylana hosil qiling.

1. Additional komponentalar palitrasidan **Shape** tugmachasini tanlaymiz.

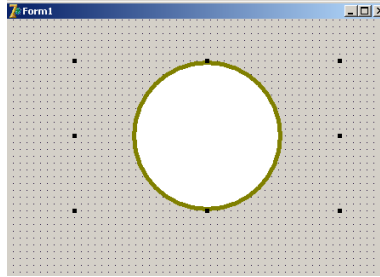
2. Ob'ektlar inspektorining **Pen** bandini tanlaymiz

3.



Aylana chizish.

4. Pen qatorining **stCircle** bandini tanlaymiz– (aylana chizish buyrug‘ini tanlaymiz).



Chizish sirti

C++ Builder muhitida chizish sirti - TCanvas sinfi dastur ishlash paytida chizish rasm chizish imkoniyatini beradi. Bu sinf ob'ekti sirt bo'yicha ko'chish grafik primitivlar chizish, rasmlar va sirlarni biror qismini nusxalash, hamda matnni chop qilish imkonini beruvchi xossa va metodlarni o'z ichiga oladi.

Har bir Canvas xossasiga ega komponenta o'z navbatida qalam, kist shrift ob'ektlarini tarkibiga oladi va mos ravishda Pen, Brush va Font xossalariga egabo'ladi.

Pen xossasi rangga (Canvas->Pen-> Color), chiziqning piksellardagi qalinlikda (Canvas-

>Pen-> Width), chizilayotgan chizikning toifasiga (Canvas->Pen->Style) ega. Chizikning toifasi quyidagi qiymatlarni qabul qilishi mumkin:

psSolid – o'zluksiz chiziq;

psDash – chiziqchalardan iborat chiziq; psDot – nuqtalardan iborat chiziq;

psDashDot – nuqta va chiziqchalardan iborat chiziq; psDashDotDot – chiziq va nuqtalardan iborat chiziq; psClear – ko'rinmas chiziq;

psInsideFrame – chizish sirtini chegaralovchi to'rtburchak ichidagi chiziq.

Brush xossasi geometrik shakllar, masalan, to'g'ri to'rtburchak va ellips ichini chiziqlar bilan to'ldirish. U quyidagi xossalarga ega:

Canvas->Brush->Color – kist rangi;

Canvas->Brush->Style – kist toifasini aniqlaydi va quyidagi qiymatlarni qabul qilishi mumkin:

bsSolid – berilgan rang bilan shakl yuzasini to'liq bo'yaladi; bsClear –shakl yuzasini bo'yalmaydi;

bsHorizontal – shakl yuzasi parallel chiziqlar bilan to'ldiriladi; bsVertical -shakl yuzasi vertikal chiziqlar bilan to'ldiriladi; bsFDiagonal - shakl yuzasi yuqoriga qaragan chiziqlar bilan to'ldiriladi; bsFDiagonal - shakl yuzasi yuqoriga pastga chiziqlar bilan to'ldiriladi; bsCross - shakl yuzasi to'r bilan to'ldiriladi;

bsDiagCross - shakl yuzasi egri chiziqlardan hosil bo'lgan to'r bilan to'ldiriladi. Canvas ob'ektning muhim xossalaridan biri

Canvas->Pixels[x][y] xossasi bo'lib, bu ko'rsatilgan koordinatadagi piksel rangini aniqlaydi. Bu xossa qiymatlari o'qish va unga qiymat yozish mumkin.

Geometrik shakllar chizish uchun qo'yida keltirilgan funktsiyalardan foydalanish mumkin: Arc(int X1, int Y1, int X2, int Y2, int X3, int Y3, int X4, int X4) – yoy chizish. Bu erda (X1,Y1) va (X2,Y2) – mos ravishda yoy chiziladigan

to'rtburchak sohaning chap yuqori va o'ng past uchlarini koordinatasi. (X3,Y3) va (X4,Y4) nuqtalar mos holda yoy boshlanishi va oxiri koordinatasi.

Chord(int X1, int Y1, int X2, int Y2, int X3, int Y3, int X4, int Y4) – ellips vatarini chizish. Bu erda (X1,Y1) va (X2,Y2) – mos ravishda yoy chiziladigan to'rtburchak sohaning chap yuqori va o'ng past uchlarini koordinatasi. (X3,Y3) va (X4,Y4) nuqtalar mos holda vatar boshlanishi va oxiri koordinatasi. Ellipse(int X1, int Y1, int X2, int Y2) – rang bilan to'ldirilgan ellipsni chizish. Bu erda (X1,Y1) va (X2,Y2) – mos ravishda yoy chiziladigan to'rtburchak sohaning chap yuqori va o'ng past uchlarini koordinatasi.

Rectangle(int X1, int Y1, int X2, int Y2) – rang bilan to'ldirilgan to'g'ri to'rtburchakni chizish. Bu erda (X1,Y1) va (X2,Y2) – mos ravishda yoy chiziladigan to'rtburchak sohaning chap yuqori va o'ng past uchlarini koordinatasi.

Misol. Forma sirtida rasm chizish

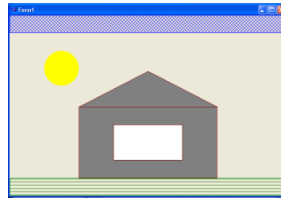
Canvas xossalariidan foydalangan holda uy rasmini chizishda ellips, to'g'ri to'rtburchak, ko'pburchak shakllari ishlatilgan. Rangni boshqarish, shakl yuzasini to'ldirish uchun qalam va kist xossalari ishlatilgan. Rasm o'lchamlari forma o'lchamiga mos ravishda o'zgaradi.

```
#include <vcl.h> #pragma hdrstop #include "Unit1.h"
//-----
#pragma package(smart_init) #pragma resource "*.dfm" TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//-----
void __fastcall TForm1::FormResize(TObject *Sender)
{
int w, h, wm, hm; Form1->Refresh();
wm=Form1->ClientWidth; w=wm/8; hm=Form1->ClientHeight; h=hm/10;
// Osmon
Form1->Canvas->Brush->Color=clBlue; Form1->Canvas->Brush-
>Style=bsDiagCross; Form1->Canvas->Pen->Color=clBlue;
Form1->Canvas->Rectangle(0,0,wm,h);
// Maysalar
Form1->Canvas->Brush->Color=clGreen; Form1->Canvas->Brush-
>Style=bsHorizontal; Form1->Canvas->Pen->Color=clGreen; Form1->Canvas-
>Rectangle(0,hm-h,wm,hm);
// Quyo'sh
Form1->Canvas->Brush->Color=clYellow; Form1->Canvas->Brush-
>Style=bsSolid; Form1->Canvas->Pen->Color=clYellow; Form1->Canvas-
>Ellipse(w,2*h,2*w,2*h+w);
// Uycha
Form1->Canvas->Brush->Color=clGray; Form1->Canvas->Brush->Style=bsSolid;
```

```

Form1->Canvas->Pen->Color=clMaroon;
Form1->Canvas->Rectangle(2*w, hm-5*h, 6*w, hm-h); TPoint point[4];
point[0].x=2*w; point[0].y=hm-5*h; point[1].x=4*w; point[1].y=hm-7*h;
point[2].x=6*w; point[2].y=hm-5*h; point[3].x=2*w; point[3].y=hm-5*h; Form1-
>Canvas->Polygon(point,3); Form1->Canvas->Brush->Color=clWhite; Form1-
>Canvas->Brush->Style=bsSolid; Form1->Canvas->Pen->Color=clMaroon;
Form1->Canvas->Rectangle(3*w, hm-4*h, 5*w, hm-2*h);
}

```



20.42-rasm. Dastur natijasida xosil bo'lgan uycha rasmi.

Dastur ishga tushishi bilan Formaning OnResize hodisasi ro'y beradi va ekranda uycha rasmi paydo bo'ladi.

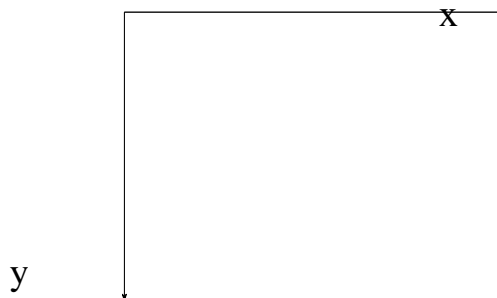
Matematik funktsiyalar grafiginin chizish. C++ Builder muhitida grafik shakllarni chizish Canvas komponentasi vositasida amalga oshiriladi. Ayrim vizual komponentalar bu komponentaga ega. Masalan, TForm, Image, PaintBox va boshqalar.

Formaning (Form1) grafik shakllar chizish sohasi bu mijoz sohasi hisoblanadi va uning o'lchami Form1->ClientWidth (gorizontaliga) va Form1-> ClientHeight (vertikaliga) bilan aniqlanadi.

Grafika sohasi adreslanuvchi nuqtalarning to'g'ri burchakli massiv ko'rinishida bo'ladi va ixtiyoriy tasvir yonib yoki o'chib turgan pikseller (tasvirning minimal elementi) kompozitsiyasidan hosil bo'ladi. Bu nuqtalar ikkita butun son: nx - nuqtaning gorizontal nomeri va ny nuqtaning vertikal nomeri bilan adreslanadi:

$0 \leq nx \leq nx_Max$; $0 \leq ny \leq ny_Max$,
bu erda $nx_Max = Form1->ClientWidth$ va $ny = Form1->ClientHeight$.

0,0



Grafika sohasining chap yuqori burchagi (0,0) koordinataga ega bo'ladi. (nx,ny) qurilma koordinatalari ham deyiladi va ular faqat butun qiymatlarni qabul qiladi.

Kompyuter grafikasida yana ikkita koordinata tizimi qabul qilingan. Birinchisi

(px, py)- ekran koordinata tizimi bo'lib, unda nx- gorizonta bo'yicha ekrandagi masofa, nu-gorizonta bo'yicha. Bu erda koordinata o'qlari millimetr va dyumlarda o'lchanadi. Ikkinchi koordinata tizimi - dunyoviy (olam) koodinata tizimidir. U (x, u) dekart tizimi bo'lib, dastur to'zuvchisi tomonidan aniqlanadi va tasvirlash qurilmasiga bog'liq bo'lmaydi:

$X_{min} \leq x \leq X_{max}$ $Y_{min} \leq y \leq Y_{max}$

Dekart koordinatalar tizimida X va Y o'zgarish diapazonlari (X_{min} , X_{max} , Y_{min} , Y_{max}) mavhum matematik ikki o'lchamli fazoning to'g'ri burchakli sohasini aniqlaydi. Bu sohani qurilma koordinatasiga akslantirish qo'yidagicha amalga oshiriladi: $n_x = \text{Round}((x - X_{min}) / (X_{max} - X_{min})) * nx_Max$;

$ny = \text{Round}((y - Y_{min}) / (Y_{max} - Y_{min})) * ny_Max$,

bu erda (x,y)- dekart koordinatasidagi nuqta va uning ekrandagi koordinatasi (n_x, n_y) bo'ladi.

Grafik kursor. Grafik kursor matn kursori bajaruvchi ishni bajaradi, lekin u ekranda ko'rinmaydi. Ma'lumki, matn kursori ekrandagi belgi o'rnini (80*25 bo'lganida) ko'rsatadi va bu o'rinda belgi chop qilinganda avtomatik ravishda bir o'rin o'ngga suriladi. Grafik kursor esa chiqariluvchi grafik shaklning boshlang'ich koordinatasini ko'rsatadi va uni keyingi joyga (n_x, n_y) nuqtaga ko'chirish uchun maxsus funktsiya ishlatiladi: `Form1->Canvas->MoveTo(n_x, n_y)`;

Chiziqlarni chizish. Sohada chiziqni (kesmani) chizish uchun

`Form1->Canvas->LineTo()` funktsiyasidan foydalaniladi. Masalan, (x_1, y_1) va (x_2, y_2) nuqtalarni tutashtiruvchi kesma chizish uchun qo'yidagi amallar bajarilishi kerak:

`Form1->Canvas->MoveTo(x_1, y_1)` ; `Form1->Canvas->LineTo(x_2, y_2)`;

Ekranda ko'p miqdordagi siniq chiziqlardan tashkil topgan shaklni chizish uchun `Canvas-`

`>Polyline(Jadval, n)`; funktsiyasidan foydalaniladi. U berilgan sondagi sonlar juftligi majmuasi bilan aniqlangan siniq chiziqni chizadi. n parametri siniq chiziq tugun nuqtalari soni. Jadval parametri `TPoint` to'rida bo'lib, grafik soha nuqta koordinatasini aniqlovchi strukturalar massivdir.

Siniq chiziq tugun nuqtalari `Jadval` massivi sifatida beriladi. Quyida `PaintBox1` komponentasi sohasida `sin(x)` funktsiya grafigini chizish funktsiyasi keltirilgan.

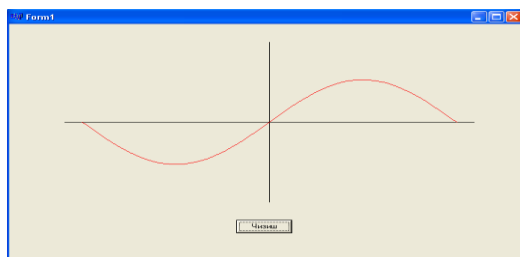
`void Sin_Grafigini_Chizish()`

```
{
const double Pi= 3.14151828; double Qadam = 0.1;
double Burchak_Radian=0; const int Nuqtalar_Soni=100; int Mashtab=50;
TPoint Sin_func[Nuqtalar_Soni]; int Absissa = PaintBox1->Height/2; for (int i=0;
i<Nuqtalar_Soni; i++)
{
Sin_func[i].x = (int)(Mashtab * Burchak_Radian) + 10; Sin_func[i].y = Absissa -
(int)(Mashtab * sin(Burchak_Radian)); Burchak_Radian+=Qadam;
}
PaintBox1->Canvas->Pen->Color=clBlack;
PaintBox1->Canvas->Polyline(Sin_func,Nuqtalar_Soni-1);
}
```

Quyidagi dasturda $\sin(x)$ funtsiya grafigini chizishning 1- varianti qaralgan.

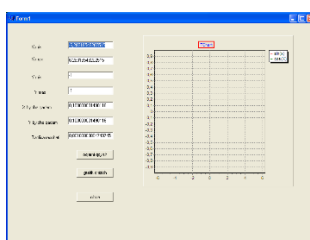
```
#include <vcl.h>
#include <math.h> #pragma hdrstop #include "Unit1.h"
#pragma package(smart_init) #pragma resource "*.dfm" TForm1 *Form1;
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
void __fastcall TForm1::Button1Click(TObject *Sender)
{
const float Pi=3.1415;
int Xe0,Ye0,Xe,Ye, // Ekran koordinatalari
Rect_X,Rect_Y, // Koordinata chegaralarining soha chegarasidan farqi Mashtab; //
Soha koordinatasining Haqiqiysiga nisbati, Mashtab
float h,X,Y; // y=f(x) funktsiya va h qadam Mashtab=80; //
Mashtabni tanlash
Rect_X =10; // Chegaralar Rect_Y =10;
h=0.1; // h qadam
Xe0 = PaintBox1->Width/2; //Koordinata markazi - Soha markazi tanlandi Ye0 =
PaintBox1->Height/2;
PaintBox1->Canvas->MoveTo(Rect_X,Ye0);
PaintBox1->Canvas->LineTo(PaintBox1->Width-Rect_X,Ye0); //OX-o'qi
PaintBox1->Canvas->MoveTo(Xe0,Rect_Y);
PaintBox1->Canvas->LineTo(Xe0,PaintBox1->Height-Rect_Y); // OY-o'qi X=-Pi-
h;
X=X+h;
Y=sin(X); Xe=Xe0+(int)(Mashtab*X); Ye=Ye0-(int)(Mashtab*Y);
PaintBox1->Canvas->MoveTo(Xe,Ye); //Grafik kursorni o'rnatish PaintBox1-
>Canvas->Pen->Color=clRed;
do
{
X=X+h;
Y=sin(X);
Xe=Xe0 + (int)(Mashtab*X); Ye=Ye0 - (int)(Mashtab*Y);
if (Xe>Rect_X && Xe < PaintBox1->Width-Rect_X && Ye>Rect_Y &&
Ye<PaintBox1->Height-Rect_Y)
PaintBox1->Canvas->LineTo(Xe,Ye); //Koordinata chegarasida chizish
}
while (X<=Pi);
}
```

Dastur ishlashi natijasida Formadagi PaintBox1 komponenta sohasida qo'yidagi chizma paydo bo'ladi.

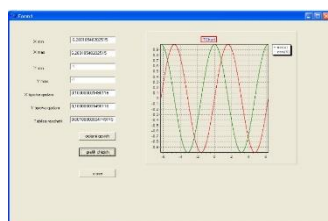


20.43-rasm. Dastur natijasida xosil bulgan rasm.

Quyidagi dasturda esa $\sin(x)$ va $\cos(x)$ funtsiyasining grafigini chizishning 2-varianti qaralgan.



20.44-rasm. Dasturning dastlabki xolati.



20.45-rasm. Dastur natijasida xosil bulgan rasm.

Dastur kodi #include <vcl.h> #pragma hdrstop #include <math.h> #include "Unit1.h"

//-----

#pragma package(smart_init) #pragma resource "*.dfm" TForm1 *Form1;
float Xmin,Xmax,Ymin,Ymax,Hx,Hy,h,Pi;

//-----

__fastcall TForm1::TForm1(TComponent* Owner)

: TForm(Owner)

{

}

//-----

void __fastcall TForm1::FormCreate(TObject *Sender)

{ Xmin=-2*M_PI; Xmax=2*M_PI; Ymin=-1; Ymax=1; Hx=0.1;

```

Hy=0.1; h=0.001;
Edit1->Text=FloatToStr(Xmin); Edit2->Text=FloatToStr(Xmax); Edit3-
>Text=FloatToStr(Ymin); Edit4->Text=FloatToStr(Ymin); Edit5-
>Text=FloatToStr(Hx); Edit6->Text=FloatToStr(Hy); Edit7->Text=FloatToStr(h);
Chart1->BottomAxis->Automatic=False; Chart1->BottomAxis->Minimum=Xmin;
Chart1->BottomAxis->Maximum=Xmax; Chart1->LeftAxis->Automatic=False;
Chart1->LeftAxis->Minimum=Ymin; Chart1->LeftAxis->Maximum=Ymax;
Chart1->BottomAxis->Increment=Hx; Chart1->LeftAxis->Increment=Hy;
}
//-----
void_fastcall TForm1::Button1Click(TObject *Sender)
{
Xmin=StrToFloat(Edit1->Text); Xmax=StrToFloat(Edit2->Text);
Ymin=StrToFloat(Edit3->Text); Ymin=StrToFloat(Edit4->Text);
Hx=StrToFloat(Edit5->Text); Hy=StrToFloat(Edit6->Text);
Chart1->BottomAxis->Minimum=Xmin; Chart1->BottomAxis->Maximum=Xmax;
Chart1->LeftAxis->Minimum=Ymin; Chart1->LeftAxis->Maximum=Ymax;
Chart1->BottomAxis->Increment=Hx; Chart1->LeftAxis->Increment=Hy;
}
//-----
void_fastcall TForm1::Button2Click(TObject *Sender)
{ float x,y1,y2; Series1->Clear(); Series2->Clear();
Xmin=StrToFloat(Edit1->Text); Xmax=StrToFloat(Edit2->Text);
h=StrToFloat(Edit7->Text); x=Xmin;
o { y1=sin(x);
Series1->AddXY(x,y1,"",clTeeColor); y2=cos(x);
Series2->AddXY(x,y2,"",clTeeColor); x=x+h;}
while(x<Xmax);
}
//-----
void_fastcall TForm1::BitBtn1Click(TObject *Sender)
{Form1->Close();}
//-----

$$Y=ax^2+bx+c$$

funksiyani grafigini chizish.

```

Form oynasiga 1 ta PaintBox1, 1ta button, 3 ta edit joylashtiramiz. Button1 ga quyidagi dasturni kiritamiz.

```

void_fastcall TForm1::Button1Click(TObject *Sender)
{
const float Pi=3.1415;
int Xe0,Ye0,Xe,Ye,a,c,b, // Ekran koordinatalari Chegara_X,Chegara_Y, //Soha
chegaralaridagi bo'sh joy o'lchami Mashtab, // Soha koordinatasining
haqiqiysiga nisbati, mashtab XY;
float h,X,Y; // y=f(x) funktsiya va h qadam Mashtab=20; //
Mashtabni tanlash Chegara_X=10; // Chegaralar Chegara_Y=10;
h=0.1; // funksiya argumenting h qadam a=StrToInt(Edit1-
>Text);

```

```

b=StrToInt(Edit2->Text); c=StrToInt(Edit3->Text);
Xe0 = PictureBox1->Width/2; //Koordinata markazi - Soha markazi tanlandi Ye0 =
PictureBox1->Height/2;
//OX-o'qini chizish
PictureBox1->Canvas->MoveTo(Chegara_X,Ye0);
PictureBox1->Canvas->LineTo(PictureBox1->Width-Chegara_X,Ye0);
// OY-o'qini chizish
PictureBox1->Canvas->MoveTo(Xe0,Chegara_Y);
PictureBox1->Canvas->LineTo(Xe0,PictureBox1->Height-Chegara_Y);
// OX son o'qidagi yo'nalish belgisini chizish (->)
PictureBox1->Canvas->MoveTo(PictureBox1->Width-Chegara_X,Ye0); PictureBox1->Canvas->LineTo(PictureBox1->Width-Chegara_X-5,Ye0-5); PictureBox1->Canvas->MoveTo(PictureBox1->Width-Chegara_X,Ye0); PictureBox1->Canvas->LineTo(PictureBox1->Width-Chegara_X-5,Ye0+5);
// OY son o'qidagi yo'nalish belgisini chizish (^) PictureBox1->Canvas->MoveTo(Xe0,Chegara_Y); PictureBox1->Canvas->LineTo(Xe0-5,Chegara_Y+5);
PictureBox1->Canvas->MoveTo(Xe0,Chegara_Y); PictureBox1->Canvas->LineTo(Xe0+5,Chegara_Y+5);
// OX son o'qini manfiy bo'lagini son kesmalariga bo'lish XY=0;
do
{
XY-=1;
Xe=Xe0+(int)(Mashtab*XY);
PictureBox1->Canvas->MoveTo(Xe,Ye0-2); PictureBox1->Canvas->LineTo(Xe,Ye0+2);
PictureBox1->Canvas->TextOutA(Xe-1,Ye0+2,IntToStr(XY));
}
while(Xe>Chegara_X);
// OX son o'qini musbat bo'lagini son kesmalariga bo'lish XY=0;
do
{
XY+=1;
Xe=Xe0+(int)(Mashtab*XY); if(Xe>PictureBox1->Width-Chegara_X)break;
PictureBox1->Canvas->MoveTo(Xe,Ye0-2); PictureBox1->Canvas->LineTo(Xe,Ye0+2);
PictureBox1->Canvas->TextOutA(Xe-1,Ye0+2,IntToStr(XY));
}
while(1);
// OY son o'qini musbat bo'lagini son kesmalariga bo'lish XY=0;
do
{
XY+=1;
Ye=Ye0-(int)(Mashtab*XY); if(Ye<Chegara_Y)break;
PictureBox1->Canvas->MoveTo(Xe0-1,Ye); PictureBox1->Canvas->LineTo(Xe0+1,Ye);
PictureBox1->Canvas->TextOutA(Xe0+2,Ye-2,IntToStr(XY));
}

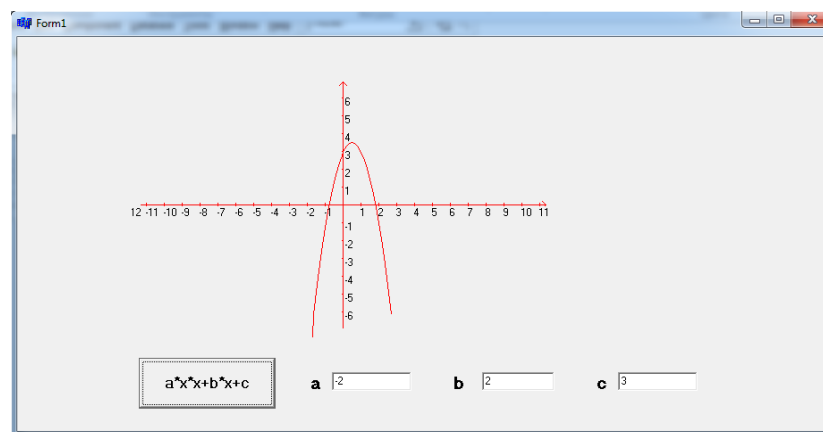
```

```

}
while(1);
// OY son o'qini manfiy bo'lagini son kesmalariga bo'lish XY=0;
do
{
XY=-1;
Ye=Ye0-(int)(Mashtab*XY); if(Ye>PaintBox1->Height-Chegara_Y)break;
PaintBox1->Canvas->MoveTo(Xe0-1,Ye); PaintBox1->Canvas-
>LineTo(Xe0+1,Ye);
PaintBox1->Canvas->TextOutA(Xe0+2,Ye-2,IntToStr(XY));
}
while(1);
X=-10-h;          // X o'zgaruvchisining chap chegarada qiymatini o'rnatish
X=X+h;
Y= a*X*X+b*X+c; // funksiya qiymati Xe=Xe0+(int)(Mashtab*X); // X qiymatiga
mos ekrandagi piksel Ye=Ye0-(int)(Mashtab*Y); // Y qiymatiga mos ekrandagi
piksel PaintBox1->Canvas->MoveTo(Xe,Ye); //grafik kursorni o'rnatish PaintBox1-
>Canvas->Pen->Color=clRed; // chiziq rangini tanlash

// funksiya grafigini chizish
do
{
X=X+h;
Y= a*X*X+b*X+c;
Xe=Xe0 + (int)(Mashtab*X); Ye=Ye0 - (int)(Mashtab*Y);
if (Xe>Chegara_X && Xe < PaintBox1->Width-Chegara_X && Ye>Chegara_Y &&
Ye<PaintBox1->Height-Chegara_Y)
PaintBox1->Canvas->LineTo(Xe,Ye); //Koordinata chegarasida chizish
}
while (X<=Pi);
}
Run tugmasini yoki klaviaturadan F9 tugmasini bosib natijani ko'ramiz.

```



Mavzu: Borland C++ Builderda multimedia va animatsiyalar.

Reja:

1. Borland C++ Builderda AlphaLite komponentalar palitrasi.
2. Borland C++ Builderda MediaPlayer komponentasidan foydalanish.

AlphaLite komponentalar palitrasi asosan dizaynni o'zgartirish uchun ishlatiladi. Bu palitra tarkibida ham Standart palitradagi TButton, TEdit ... komponentalari mavjud faqat AlphaLite palitrasidagi komponentlarni rangi, shakli umuman dizaynini o'zgartirish mumkin. Dizaynlar skinlar yordamida tanlanadi. Skinslar ro'yxati Skins papkasida joylashgan. Papkani C diskka nusxalaymiz. Palitrani kompyuterga o'rnatish uchun komponentalar joylashgan D7 papkasini C:\Program Files\Borland\C++ Builder7\Lib papkasiga nusxalaymiz. C++ Builder dasturini ishga tushiramiz va File/Open amalini bajaramiz. Ochilgan muloqot oynasida D7 papkasida joylashgan acntD7_R.dpk faylini tanlaymiz va Открыть tugmasini bosamiz. Ochilgan dastur oynasidan Compile tugmasini bosamiz, faylni saqlab oynani yopamiz. Bundan keyin yana File/Open amalini bajaramiz. Ochilgan muloqot oynasida D7 papkasida joylashgan acntD7.dpk

faylini tanlaymiz va Открыть tugmasini bosamiz. Ochilgan dastur oynasidan Install tugmasini bosamiz so'ngra dasturdan chiqamiz.

C++ Builder dasturning Tools menyusining Environment Options... bandini tanlaymiz. Natijada Environment Options... darchasi ochiladi. Darchaning Library bandini ochamiz. Bo'limning Library path bandini ochib C:\Program Files\Borland\C++ Builder7\Lib\D7 papkasini kutubxonaga qo'shamiz. Natijada dasturning komponentalar palitrasining so'ngida AlphaLite palitrasi qo'shiladi. Palitraning joylashgan o'rnini sichqoncha yordamida o'zgartirish mumkin. Buning uchun AlphaLite yozuvi ustida sichqoncha chap tugmasini bosgan holda kerakli joyga joylashtiramiz.

AlphaLite komponentalar palitrasida quyidagi komponentalar joylashgan: sSkinManager
sSkinProvider sEdit sCheckBox sPanel sButton sBitBtn sScrollBar sLabel
sWebLabel sLabelFX sComboBox sListBox
sCheckListBox sGauge sAlphaImageList

Ushbu komponentalar ham odatiy komponentalar kabi ishlatiladi. Biroq komponentalarning nomi "s" harfi bilan boshlangan. Bu skin so'zidan kelib chiqqan. Komponentalarga murojaat qilish ham ushbu nomlar orqali amalga oshiriladi. Masalan sEdit1-

>text, sLabel2->caption yoki sComboBox1 ...

Endi ushbu komponentalardan foylanishni ko'rib chiqamiz. Buning uchun C++ Builder dasturini ishga tushiramiz. Forma oynasiga AlphaLite komponentalar palitrasidan sSkinManager, sEdit, sLabel va boshqa komponentalarni joylashtiramiz. Komponentalar odatdagidek ko'rinishda bo'ladi. Ularning ko'rinishini o'zgartirish uchun sSkinManager komponentasini sichqoncha bilan belgilaymiz. Object Inspector oynasining Skin Name bo'limi ro'yxatidan kerakli skin nomini tanlaymiz. Natijada formadagi komponentalar tanlangan skiniga mos holda ko'rinishini o'zgartiradi. Masalan,

LikeOperaStyle skini Retro skini

SoapSky – Lime skini

Borland C++ Builderda multimediali ilova yaratish (mp3 player misolida)
Endi esa C++ Builder muhitida audio player dasturini yaratishni ko`rib chiqamiz.
Buning uchun C++ Builder dasturini ishga tushiramiz. Forma oynasiga Win3->1
komponentalar palitrasidan DirectoryListBox1, FileListBox1, DriveComboBox1,
FilterComboBox1 komponentalarni joylashtiramiz. System komponentalar
palitrasidan Timer1 va MediaPlayer1 komponentasini joylashtiramiz. AlphaLite
komponentalar palitrasidan sSkinManager1, sPanel1,

sComboBox1, sLabel1, sGauge1, sScrollBar1, sBitBtn1, sLabel2
komponentalarini joylashtiramiz. Formaning ko`rinishi quyidagicha:

Forma oynasini tayyorlab oldik. Keyingi bosqich dastur kodlarini kiritib chiqish.
Buning uchun DriveComboBox1 komponentasini belgilab Object Inspector
oynasining DirList bo`limiga DirectoryListBox1 ni kiritamiz. DirectoryListBox1 ni
belgilab Object Inspectorning FileList bo`limiga FileListBox1 ni kiritamiz.
FilterComboBox1ni belgilab Object Inspectorning FileList bo`limiga FileListBox1 ni
Filter bo`limiga Audio- *.mp3; *.wav ni kiritamiz. sScrollBar1 ni belgilab Object
Inspectorning Position bo`limiga 0 (nol), SkinManager bo`limiga sSkinManager1 ni
kiritamiz. sGauge1 ni belgilab Object Inspectorning Position bo`limiga 0 (nol)
kiritamiz. MediaPlayer1 ni belgilab Object Inspectorning VisibleButtons bo`limining
btPlay, btPause, btStop bandlariga true, qolgand bandlariga falseni moslaymiz.
sComboBox1 ni belgilab Object Inspectorning Items bo`limini tanlaymiz. Hosil
bo`lgan Items oynasiga bir nechta skinlar nomini kiritamiz va OK tugmasini bosamiz.
Forma oynasining dastur oynasiga o`tamiz va quyidagi buyruqlarni kiritamiz:

```
{  
    sSkinManager1->SkinDirectory=  
    'c:\Skins\'; sSkinManager1->SkinName =  
    sComboBox1->Text; sSkinManager1-  
    >Active = true;  
    mediaplayer1-  
    >FileName=ExtractFilePath(Application.ExeName)+'musiqqa.mp3';  
    mediaplayer1->AutoOpen=true;  
    mediaplayer1-  
    >DeviceType=dtAutoSelect;  
    mediaplayer1->Open;  
    mediaplayer1->Play;  
    sScrollBar1-  
    >Max=MediaPlayer1->Length;  
    sScrollBar1->Position=0;  
    sgauge1-  
    >MaxValue=MediaPlayer1-  
    >Length; sgauge1->Progress=0;  
}
```

sComboBox1 komponentasining dastur oynasiga o`tamiz va quyidagi

buyruqlarni kiritamiz:

```
{
sSkinManager1->SkinDirectory=
'c:\Skins\'; sSkinManager1->SkinName =
sComboBox1->Text; sSkinManager1-
>Active = true;
}
```

FileListBox1 komponentasining Events – DblClick hususiyatiga quyidagi buyruqlarni kiritamiz:

```
{
MediaPlayer1->FileName =fileListBox1->Items.Strings[fileListBox1-
>ItemIndex]; MediaPlayer1->Visible = true;
MediaPlayer1->DeviceType =
dtAutoSelect; MediaPlayer1-
>Open;
MediaPlayer1->Play;
sScrollBar1-
>Max=MediaPlayer1->Length;
sScrollBar1->Position=0;
}
```

Timer1 komponentasining dastur oynasiga quyidagi buyruqlarni kiritamiz:

```
{

sScrollBar1->Position=MediaPlayer1-
>Position; sgauge1-
>Progress=mediaplayer1->Position;
}
```

sScrollBar1 komponentasining Event – OnScroll hususiyatida quyidagi buyruqlarni kiritamiz:

```
{
MediaPlayer1->Position= sScrollBar1-
>Position; MediaPlayer1->Play;
}
```

sBitBtn1 komponentasining dastur oynasiga quyidagi buyruqlarni kiritamiz:

```
{
form1->Close;
}
```

Buyruqlarni kiritib bo`ldik. Dasturni ishga tushirishdan oldin loyiha saqlangan papka ichiga musiqa.mp3 faylini joylashtiramiz. Buning sababi dastur ishga tushganda birinchi bo`lib musiqa.mp3 fayli ijro etiladi. Keyinchalik DirectoryListBox1 dan kerakli papkaga o`tamiz va FileListBox1 ro`yxatidan ijro etilishi kerak bo`lgan faylni sichqoncha yordamida ikki marta chertamiz. Natijada tanlangan fayl ijro etiladi.

Loyihani saqlaymiz va F9 tugmasi orqali dasturni ishga tushiramiz.

Amaliy mashg'ulot ishlanmalari

1-Amaliy mashg'ulot.

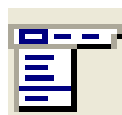
Mavzu: Borland C++ Builder Standart komponentalar palitrasi.



Standard komponentalar

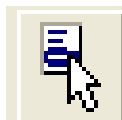
Komponentalar palitrasining Standard qo'shimcha ilovalari komponentalari sizning dasturingizga Windows standart interfeysli elementlarning 14 tasining ulanishini amalga oshiradi.

TMainMenu



Bosh menyu komandolari Panelini va ularga mos keladigan tushib qoladigan menyularni yaratadi. Barcha menyu komandalarining identifikatorlari menyuning har qanday konkret komandasiga kirish huquqiga ega bo'lgan Items xususiyati bilan aniqlanadi. AutoMerge xususiyati Merge va UnMerge metodlari bilan birgalikda turli shakldagi menyularning birlashish jarayonini boshqaradi.

TPopupMenu



Shakl yoki bironta boshqa komponenta uchun maxsus menyu yaratadi. E'tiborga oling, aynan shu maqsad uchun har qanday boshqa komponenta PopUpMenu xususiyatiga ega bo'lib, bu xususiyatda siz uning bilan bog'liq menyuga iqtibos qilishingiz mumkin.

Agar siz sichqonchaning o'ng tugmasini shaklga yoki berilgan komponenta mansub bo'lgan biron boshqa elementga bosish bilan maxsus menyu ekranda paydo bo'lishini xoxlasangiz, AutoPopUp xususiyatining true qiymatini o'rnatish. Voqea qayta ishlatgichi - OnPopUp yordamida bevosita maxsus menyuning paydo bo'lishi oldidan bajariladigan protsedurani aniqlashi 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'rnatish. Kalta soha ichida matnning hammasini ko'rish imkoniga ega bo'lish uchun, WordWrap xususiyatining true qiymatini berish. Transparent xususiyatining true qiymatini o'rnatish, 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 hatti- 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 bekorqilish 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 check-boxni yaratadi (bunda tavsifiy matn check-boxning vazifasini spetsifikatsiya qiladi).

Box holatini bildiruvchi «Check» biron-bir variantning tanlanishiga mos keladi (box ustidan tortilgan chiziq bilan belgilanadi). «UnCheck» 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. AllowGraed xususiyatining true qiymatini o'rnatib, boxni to'qroq rangli (masalan, kulrang) qilish mumkin. State xususiyati joriy holatni va box 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. Radio-tugma bosilganda, Checked komponentasining xususiyati ham mos ravishda o'zgaradi va OnClick voqeasi yuzaga keladi.

Odatda radio-tugmalar avvaldan shaklda o'rnatilgan konteyner ichiga joylashtiriladi. Agar bitta tugma tanlangan bo'lsa, ushbu guruhga mansub barcha boshqa tugmalarning tanlovlari avtomatik tarzda olib tashlanadi. Masalan, shakldagi ikkita radio-tugma, agar ular boshqa-boshqa konteynerlarda joylashgan bo'lsagina bir paytning o'zida tanlab olinishi mumkin. Agar radio-tugmalarning guruhlanishi ochiq-oydin berilmagan bo'lsa, bu holda ularning hammasi, yashirin holda, konteyner darchalari (TForm, TGroupBox yoki TPanel) dan birida guruhlanadi.

TRadioButton komponentasi TButtonControl sinfining hosilasidir.

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 lineyasi

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

ListBox1->Items->Add («Ro'yxatning oxirgi elementi»);
Sorted xususiyatining true qiymati ro'yxat elementlarini alifbo tartibida ajratib joylashtiradi.

TListBox komponentasi TCustomListBox sinfining hosilasi bo'lib, uning barcha xususiyat, metod va voqealariga vorislik qiladi.

TComboBox



Tahrir sohasi hamda matn variantlarining tushib qoladigan ro'yxati kombinatsiyasini tanlash uchun ishlatiladi.

Text xususiyatining qiymati bevosita tahrir sohasiga kiritib qo'yiladi. Foydalanuvchi tanlab olishi mumkin bo'lgan ro'yxat elementlari Items xususiyatining ichida bo'ladi. Dasturning bajarilish paytida tanlab olinishi mumkin bo'lgan element raqami ItemIndex xususiyatining ichida bo'ladi. 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.

Items obyektining Add, Append, Delete va Insert metodlari yordamida ro'yxat elementlarini dinamik tarzda qo'shish, o'chirish orasiga qo'yish va o'rnini almashtirish mumkin. Masalan:

ComboBox->Items->Insert(0, «Ro'yxatdagi birinchi element»);

Sorted xususiyatining true elementi ro'yxat elementlarini alifbo tartibida navlarga ajratilishini ta'minlaydi. TComboBox komponentasining turini Style xususiyatidan tanlab olish mumkin.

TComboBox komponentasi TCustomComboBox sinfining hosilasi bo'lib uning barcha xususiyatlari, metodlari va voqealariga vorislik qiladi.

TScrollBar



Darcha, shakl yoki boshqa komponenta ichidagilarini ko'rib chiqish uchun ishlatiladi. Masalan, biron-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 SmallShange 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. SetPcirusms metodi bir paytning o'zida Min, Max va Position ga tegishli barcha xususiyatlar qiymatlarini aniqlab beradi.

TGroupBox



To'g'ri burchakli ramka ko'rinishidagi konteyner bo'lib, u qandaydir bir interfeys elementlarining mantiqan bog'langan guruhini shaklda vizual birlashtiradi. Bu komponenta Windows ning bir nomdagi obyektning inkapsulyatsiyalanishidan iborat.

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

TPanel



Boshqa komponentlarni 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 komponentasi TCustomPanel sinfining hosilasi bo'lib, uning barcha xususiyatlar, metodlari va voqealari to'liq vorislik qiladi.

2-Amaliy mashg'ulot.

Mavzu: Borland C++ Builder Additional komponentalar palitrasi.



20.38-rasm. Additional komponentalari

Komponentalar palitrasining Additional qo'shimcha ilovalar komponentalari sizning dasturingizga Borland korporatsiyasi maxsus C++ Builder muhiti uchun ishlab chiqqan 9 ta boshqarish elementini kiritadi.

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 uchun xususiyatlarga ega bo'ladi. Siz C++ Builder qurilmasiga kirgan alohida tasvirlar katalogidagi grafik tugmalarning tayyor stillaridan foydalanishingiz ham, yoki bo'lmasa, tasvirlarni tahrir qilish tizimlarining biri tomonidan yaratilgan suratlardan foydalanishingiz ham mumkin.

Tugmaning turli holatlariga (masalan «bosilgan», «qo'yib yuborilgan», «ta'qiqlangan» va h.k.) turli bit obrazlari mos kelishi mumkin.

Tasvirlar Fayllari Muharririning bmp kengaytmali darchasi (20.39-rasm) Glyph xususiyatining qiymatlari tugmasi bilan ochiladi. King xususiyati sizga yozuvlar va tegishli grafika (OK, Cancel, Hello va boshqalar) bilan ta'minlangan standartlashtirilgan tugmalarni yaratish imkonini beradi.



20.39-rasm. BMP kengaytmali bit obrazlari fayllari tasvirlarining muharriri

TSpeedButton



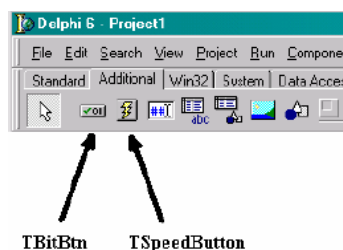
Odatda ma'lum menyu komandalarini tez chaqirish yoki rejimlarni o'rnatish Paneli (TPanel) da joylashtiriladigan grafik tugmani yaratadi.

Tezkor tugmaning turli holatlariga (masalan «bosilgan», «qo'yib yuborilgan»,

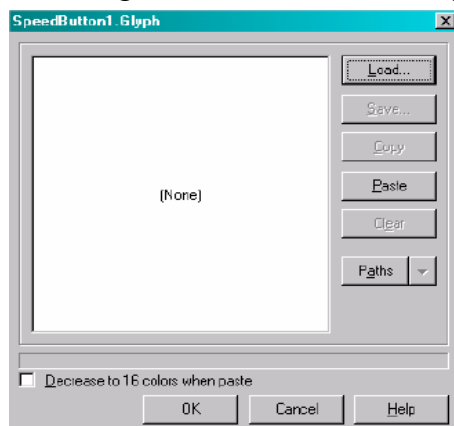
«ta'qiqlangan» va h.k.) turli bit obrazlari mos kelishi mumkin. Bir-birining o'rnini bosadigan tasvirlar va yozuv matnini tanlash uchun xususiyatlar mavjud. Kengayishli tasvirlar fayllari muharririning darchasi Glyph xususiyati qiymtlarining grafasidagi tugma bilan ochiladi. Tez tugmalarning boshqa xususiyatlari ularning biron-bir guruhdagi ishini tashkil etadi.

TSpeedButton va TBitBtn tugmalari.

Bu tugmalar **TButton** vazifalarini bajaradi. Yagona farqi matndan tashqari racmlarni ham aks ettiradi. **TSpeedButton** tugmasi fokus olmaydi. Bu shuni bildiradiki, agar matn qatorida satr terib, bu tugma bosilsa, shu hodisa qayta ishlangandan so'ng fokus yana matn qatoriga qaytib keladi. TAB tugmasi bilan bu tugmani ajratib bo'lmaydi.



Tugmaga rasm o'rnatish uchun ikki marta *Glyph* xossasi qatorida chertish lozim. Natijada rasm paydo bo'lgan yuklash oynasida *Load* tugmasini bosish lozim. Ko'p rasmlar Program Files\Common



Files\Borland Shared\Images\Buttons katalogida joylashgandir.

TBitBtn va **TSpeedButton** tugmalari deyarli bir xil xossalarga egadir. Ular uchun umumiy *Layout*, xossasi rasm va matn o'zaro joylashuvini o'zgartirishga imkon beradi.

Quyidagi rasmda har xil qiymatlar mos variantlari ko'rsatilgan



TBitBtn tugmasining yana bir xossasi *Kind* bo'lib oldindan tayyorlangan standart tugmalarni tanlash imkonini beradi. Quyidagi rasmda standart tugmalar va ularga mos qiymatlarni ko'rish mumkin.

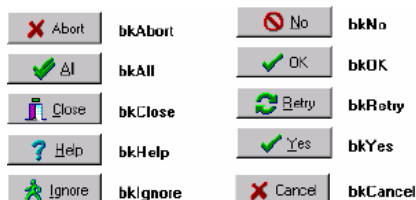


Рис 11.1.7 Различные значения свойства Kind.

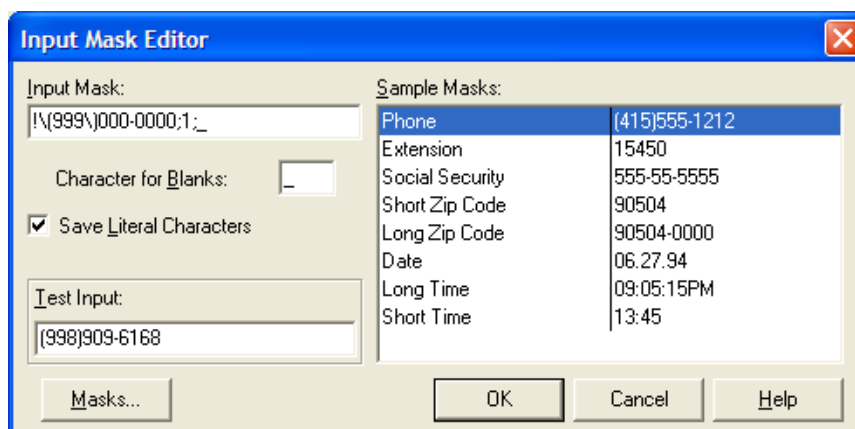
Yana bir xossa *ModalResult* – dialog oynasi uchun tugma qaytaradigan natijani tanlashga imkon beradi.

TSpeedButton tugmasining *GroupIndex* xossasi tugmalarni guruhlashga imkon beradi. Buning uchun bir guruhga tegishli tugmalarning *GroupIndex* xossasi bir xil qiymatga masalan 1 ga teng bo'lishi kerak. Guruhlangan tugmalarning biri bosilsa, qolganlaridan ajralib qoladi. Buning uchun *Down* xossasi qiymati *true* ga teng bo'lishi kerak.

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 (20.40-rasm) ushbu xususiyat qiymatlari grafasida tugma bilan ochiladi.



20.40-rasm. Telefon raqamlarini kiritish uchun niqobni yaratish

TMaskEdit komponentasi TCustomMaskEdit sinfining to'g'ridan-to'g'ri hosilasidir. U satrlar yoki ustunlar bo'yicha belgilar ketma-ketliklarni aks ettirish uchun mo'ljallangan muntazam (regulyar) to'rtburchak yaratadi.

TStringGrid



Ushbu 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 to'tish imkonini beradi. Setkaning barcha simvolli ketma-ketliklari setkaning kerakli uyasini adreslaydigan (manzillaydigan) Cells xususiyatida mujassamdir.

TDrawGrid



To'zilma holiga keltirilgan grafik ma'lumotlarni satrlar yoki ustunlar bo'yicha aks ettirish uchun muntazam setka yaratadi. RowCount va ColCount xususiyatlari vertikal bo'yicha 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 xususiyatli tomonidan belgilanadi, aylantirish chiziqlari esa ScrollBars xususiyati tomonidan qo'shiladi. FixedCol va FixedRows xususiyatlari ustunlar va satrlarning aylantirilishini ta'qiqlab qo'yish imkonini beradi, Fixed Color xususiyati esa barcha usutn 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. Default Drawing xususiyatining false qiymatini o'rnatish uchun, setka uyalarini «qo'lda» to'ldirish uchun mo'ljallangan OnDrawCell voqeasi qayta ishlatgichining yozilishini talab qiladi. DefaultColWidths va DefaultRowHeights xususiyatlari yordamida yashirin tanlanayotgan barcha ustunlar va satrlarning enini o'rnatish mumkin. ColWidth va RowHeight xususiyatlari konkret ustun enini va konkret satr bo'yini spetsifikatsiyalaydi.

Dasturning ishlash paytida siz CellRest metodi yordamida biron-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.

Dastur bajarilish paytida qaysi satr setkaning ustki satri bo'lishini aniqlash yoki TopRow xususiyati yordamida ko'rsatilgan satrni ustki holatga qo'yib qo'yish mumkin. Qaysi ustun setkaning ko'rinib turadigan usutni bo'lishini aniqlash uchun, LeftCol xususiyatidan foydalaning. VisibleColCount va VisibleRowCount xususiyatlarining qiymatlari setkaning ko'rinib turgan ustunlari va satrlarining umumiy sonini spetsifikatsiyalaydi.

TImage



Shaklda grafik tasvir konteynerini yaratadi (bu bit obrazi, piktogramma yoki meta fayla bo'lishi mumkin).

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

Tasvirlar fayllarining dinamik yuklanishi va saqlanishi uchun, Picture obyekt xususiyatining LoadFromFile va SaveToFile metodlaridan qo'yidagi turlar yordamida foydalaning:

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



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

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

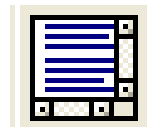
TBevel



Xuddi uskuna bilan o'yilgandek xajmli ko'rinadigan chiziqlar, box lar yoki ramkalarni yaratadi.

Komponenta chizayotgan obyektни Shape xususiyati aniqlaydi, Style xususiyatining qiymati esa obyekt ko'rinishini o'zgartirib, uni bo'rtiq yoki botiq holga keltiradi. Foydalanuvchi shakl o'lchamlarini o'zgartirganda ham obyektning nisbiy holatini o'zgarmas qoldirish uchun, Align xususiyatining true qiymatini o'rnatish.

TScrollBar



Darchada o'lchamlari o'zgaruvchan box ni yaratadi, bu box 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 biron- 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.

3-Amaliy mashg'ulot.

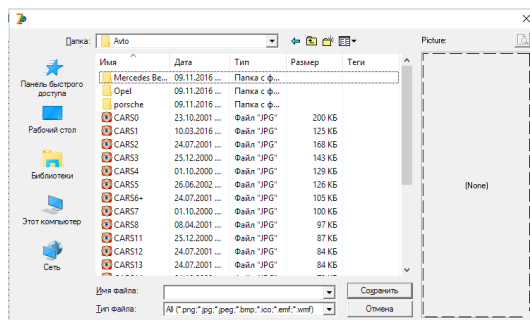
Mavzu: Borland C++ Builder Dialog komponentalar palitrasi.

Dialogs sahifasi komponentalari Windowsning odatiy dasturlari muloqot oynalarini hosil qilishda foydalaniladi. Dialogs komponentalari fayllarni ko'rsatish yoki sozlashlarni o'rnatish uchun ishlatiladi. C++ Builderning Dialogs komponentalaridan foydalanish ilovalarni yaratishda vaqtini tejaydi va Windowsning odatdagi muloqot oynalaridan foydalanish imkonini yaratadi.



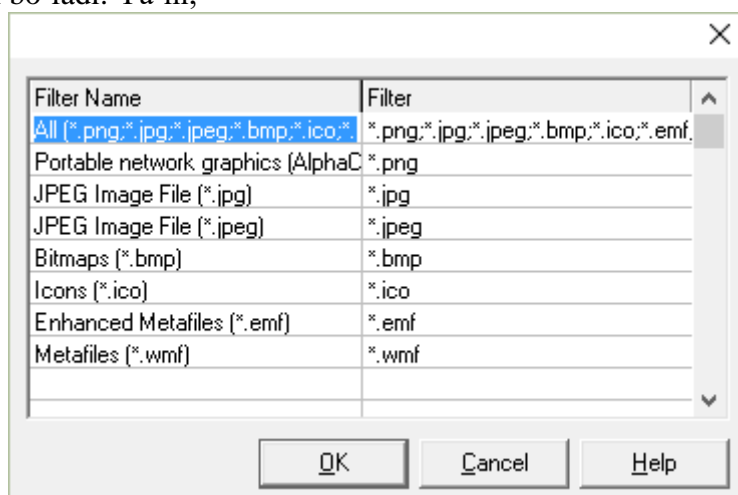
№	Komponenta	Tavsifi
12.	 OpenDialog	«Открыть файл» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
13.	 SaveDialog	«Сохранить файл как...» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
14.	 OpenPictureDialog	«Открыть рисунок» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
15.	 SavePictureDialog	«Сохранить рисунок как...» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
16.	 FontDialog	«Шрифты» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
17.	 ColorDialog	«Цвет» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
18.	 PrintDialog	«Печать» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
19.	 PrinterSetupDialog	«Установка принтера» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
20.	 FindDialog	«Найти» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
21.	 ReplaceDialog	«Заменить» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
22.	 PageSetupDialog	«Параметры страницы» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.

OpenDialog – faylni ochish muloqot oynasi va **SaveDialog** faylni saqlash muloqot oynasi komponentalari deyarli barcha dasturlarda ishlatiladi. Muloqot oynalari quyidagi ko'rinishda bo'ladi.



Rasmi saqlash muloqot oynasi

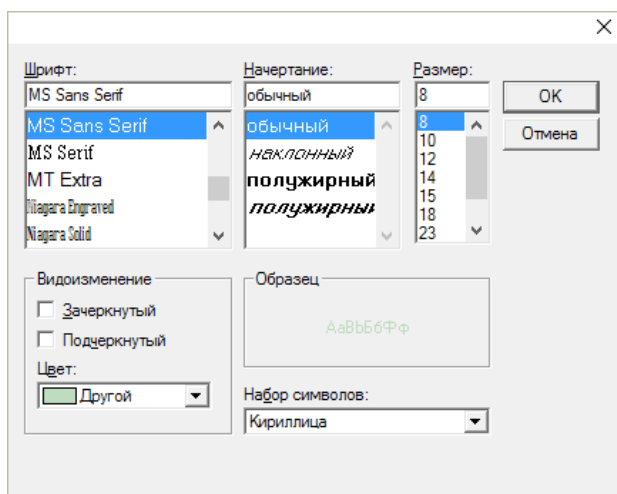
OpenPictureDialog va SavePictureDialog komponentalarning xususiyatlari OpenFileDialog va SaveDialog komponentalarining xususiyatlari bilan bir xil. Yagona farq shundaki OpenPictureDialog va SavePictureDialog komponentalarning Filter xususiyati oldindan sozlangan bo'ladi. Ya'ni,



Filter oynasi

Ushbu feltirda barcha grafik fayl turlari kiritilgan. Foydalanuvchi hohlasa ularni o'chirishi yoki qo'shimcha kiritishi mumkin.

FontDialog komponentasi shrift xususiyatlarini sozlash muloqot oynasini chaqiradi. Ushbu oynada shrift nomi, yozilish shakli, o'lchami va boshqa xususiyatlarni sozlash mumkin.

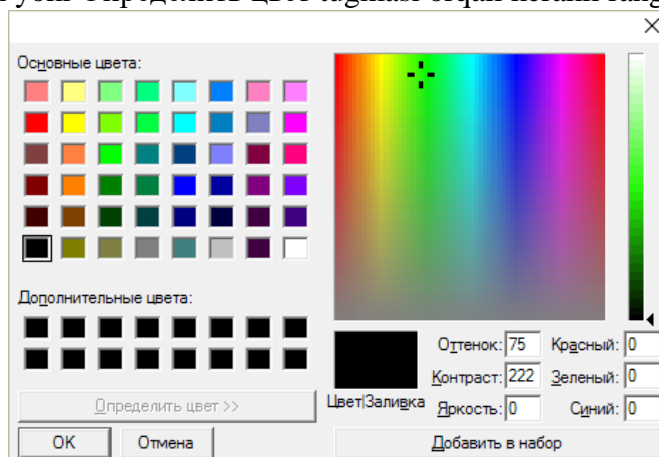


Shrift xususiyatini sozlash oynasi

Komponentning asosiy xususiyati – Font. **MaxFontSize** va **MinFontSize** xususiyatlari shrift o'lchamining eng katta va eng kichik qiymatini o'rnatadi. Agar xususiyatlarning qiymati 0 (nol) ga teng bo'lsa (по умолчанию), o'lchamni kiritishda hech qanday cheklashlar kuzatilmaydi. Agarda xususiyatlarga qiymat kiritilsa, muloqot oynasining **Размер** bo'limida kiritilgan diapazondagi

qiymatlar hosil bo`ladi. Agar diapazondan tashqaridagi qiymat kiritilsa, «Размер должен лежать в интервале ...» ogohlantirish oynasi hosil bo`ladi.

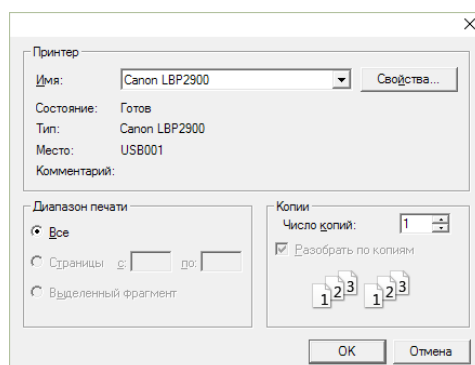
ColorDialog komponentasi rang tanlash muloqot oynasini chaqiradi. Bundan foydalanuvchi bazadagi ranglar palitrasidan yoki Определить цвет tugmasi orqali kerakli rangni tanlashi mumkin.



Рang tanlash oynasi

ColorDialog komponentasining asosiy xususiyati – Color. Bu xususiyat foydalanuvchi oynadan tanlagan rangni bildiradi.

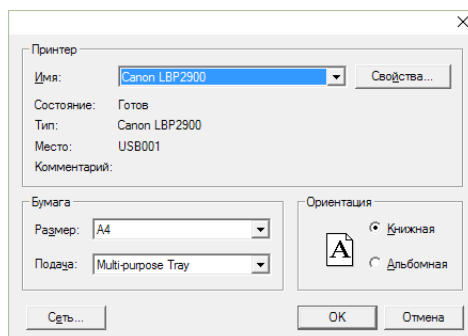
PrintDialog komponentasi chop qilish muloqot oynasini chaqiradi. Foydalanuvchi bunda printerni tanlashi va uning xususiyatini sozlashi, nusxalash sonini ko`rsatishi, sahifani yoki belgilangan fragmentni chop qilishi mumkin.



Chop qilish oynasi

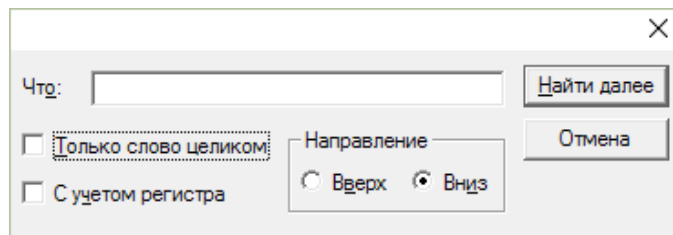
PrintDialog komponentasi chop qilishni amalga oshirmaydi. U faqat chop qilish xususiyatlarini sozlaydi. Chop qilish uchun esa **Printer** komponentasidan foydalaniladi.

PrinterSetupDialog komponentasi printer xususiyatlarini sozlash muloqot oynasini chaqiradi. Bu komponent o`rnatilishi yoki o`qilishi kerak bo`lgan o`ziga xos xususiyatlari yo`q yagona komponentdir. Dialog printerni sozlash amallarini bajaradi. Bu dialog hech qanday parametr qaytarmaydi.



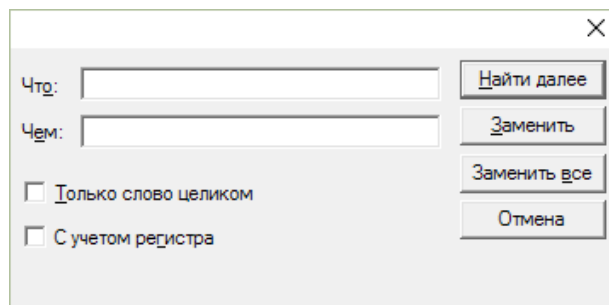
Printerni sozlash oynasi

FindDialog komponentasi satrning biror qismini qidirish (Найти) muloqot oynasini chaqiradi.



Satrnı qidirish oynasi

ReplaceDialog komponentasi satrning biror qismini boshqasiga almashtirish (Замена) muloqot oynasini chaqiradi.



Satrnı almashtirish oynasi

Ushbu komponentalar juda o`xshash va bir xil xususiyatlarga ega. Yagona farq

ReplaceDialog komponentasida satrni almashtirish mavjud.

Komponentlarning asosiy xususiyatlari:

FindText – foydalanuvchi tomonidan kiritilgan qidiriladigan yoki almashtiriladigan satr. **ReplaceText** – **ReplaceDialog** komponentasiga tegishli. **FindText** ga almashtirilishi kerak bo`lgan satr.

Options – ko`plab xususiyatlar.

FindDialog va **ReplaceDialog** komponentalari o`zlari qidiruv yoki almashtirishni amalga oshirmaydi. Ular faqat foydalanuvchi interfeysini sozlaydi. Qidirish yoki almashtirishni amalga oshirish uchun kerakli buyruqlarni kiritish talab qilinadi. Muloqot oynasidagi Найти далее tugmasi ishlashi uchun **OnFind** hodisasiga, Заменить yoki Заменить все tugmalari uchun esa **OnReplace** hodisasiga foydalanuvchi tegishli buyruqlarni kiritishi lozim.

4-Amaliy mashg`ulot.

Mavzu: Borland C++ Builderda massivlarga doir dastur tuzish.

Ko`p hollarda jadval yoki matritsalar ko`rinishidagi ma`lumotlar bilan ish yuritish kerak bo`ladi. Jadvalda ma`lumotlar juda ko`p bo`lgani sabab, ularning har bir yacheykasidagi sonni mos ravishda bitta o`zgaruvchiga qiymat qilib berilsa ular ustida ish bajarish ancha noqulayliklarga olib keladi. Shu sabab dasturlashda bunday muammolar massivlarni ishlatish yordamida hal qilinadi.

Massiv - bu bir nom bilan belgilangan qiymatlar guruhi yoki jadvaldir. Massivning har bir elementi massiv nomidan so`ng kvadrat qavs ichiga olingan raqam va arifmetik ifoda yozish bilan belgilanadi. Qavs ichidagi raqam massiv indeksini belgilaydi. Vektorni bir o`lchovli massiv, matritsani ikki o`lchovli massiv deb qarash mumkin.

Bir o`lchovli massivda uning har bir elementi o`zining joylashgan o`rin nomeri bilan aniqlanadi va nomeri qavs ichida indeks bilan yoziladi. Ikki o`lchovli massiv elementi o`zi joylashgan

satr va ustun nomerlari yordamida aniqlanadi. Shu sabab ikki o'ldchovli massiv elementi ikkita indeks orqali yoziladi. Masalan: A[i,j] bu yerda i-satr nomeri j-ustun nomerini bildiradi.

Massivni e'lon qilish dasturning bosh qismida berilib, uning yozilishi umumiy holda quyidagicha bo'ladi:

<Massiv nomi>:Array[o'ldham] of <element turi>;

Masalan: **A,B:array[1..10] of real; C,A1,D1:Array[1..10,1..20] of real;**

bu yerda A va B massivlari 10 tadan elementga ega. C,A1,D1 massivlari esa 10x20=200 tadan elementga ega.

Massivlarni e'lon qilishdan maqsad massiv elementlari uchun kompyuter xotirasidan joy ajratishdir. Yuqoridagi ma'lumotlarni tasdiqlash uchun massivlarga oid bir nechta misollar ko'rib o'tamiz.

1-misol. Guruhlar ro'yxatini chiqaruvchi dastur tuzing.

Dastur kodi va oynaning umumiy ko'rinishini keltiramiz:

unit unit1; interface uses

**Windows, Messages, SysUtils, Variants, Classes,
Graphics, Controls, Forms,Dialogs, StdCtrls;**

type

TForm1 =

class(TForm)

Button1:

TButton;

Label1:

TLabel;

procedure Button1Click(Sender:

TObject); private

{ Private declarations } public

{ Public

declarations }

end; var

Form1: TForm1; implementation

{ \$R *.dfm } const

NT = 6;

var

Massiv: array[1..NT] of

string[10] = ('3-A','3-

G','3-D','3-E','3-J','3-Z');

procedure TForm1.Button1Click(Sender:

TObject); var

st:string; // massiv ro'yxati

i:integer; // indeks, massiv

elementlarining nomeri begin

// formalashgan ro'yxat formada

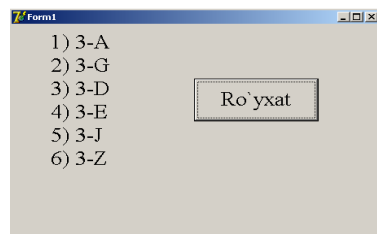
joylashish bo'yicha for i:=1 to NT do

st:= st + IntToStr(i)+ ' ' +

Massiv[i] + #13; Label1.Caption:=

st; // ro'yxatni chiqarish end;

end.



3.5.1-rasm. Guruhlar ro'yxatini chiqaruvchi dastur

2-misol. $A(5)$ massiv elementlarini yig'indisi va o'rta arifmetigini topish dasturi tuzing.

1-usul.

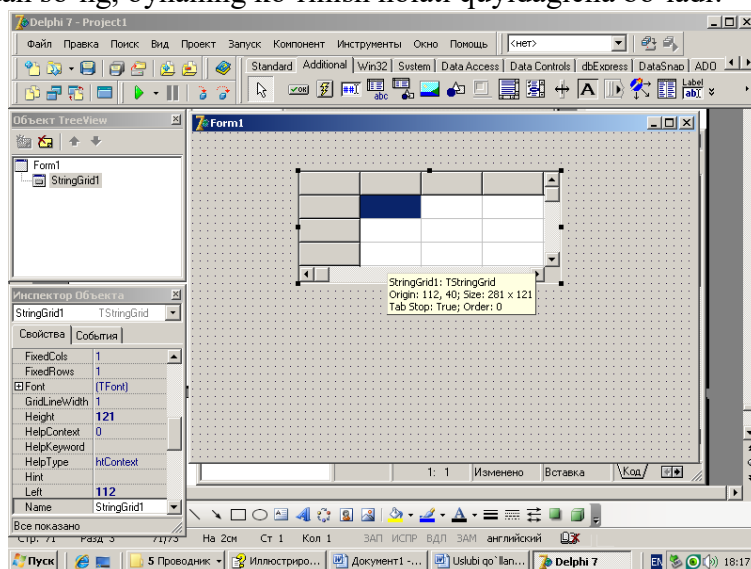
1-bosqich.

1. Komponentalar ro'yxatini Additional bandidan StringGrid1(abc) tugmani tanlaymiz, ushbu tanlash (3.5.2-rasmda) ko'rsatilgan.



3.5.2.-rasm. Additional panilining ko'rinishi

Tugma tanlangandan so'ng, oynaning ko'rinish holati quyidagicha bo'ladi:



3.5.3-rasm. StringGrid1(abc) tugmasidan foydalanish

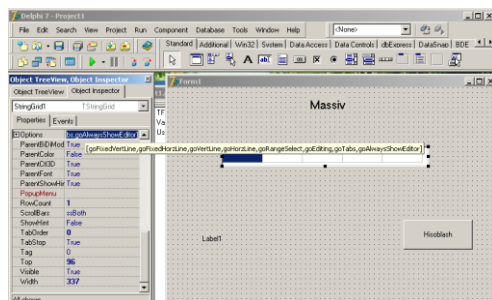
Hosil bo'lgan jadvaldan 5 ta ustun, 1 ta satr yarating. Bu jadvalni yaratish uchun parametrlar va xossalar bo'limidan **ColCount** hodisalar bandiga 5 raqamini kiritamiz. Satr bo'yicha o'zgartirish kiritish uchun **RowCount** bandiga 1 raqamini kiritamiz.

Umumiy holda oynani ishga tayyorlashimiz uchun parametrlar va xossalar bo'limidan jadvalda ko'rsatilgan buyruqlarning tarkibi quyidagicha:

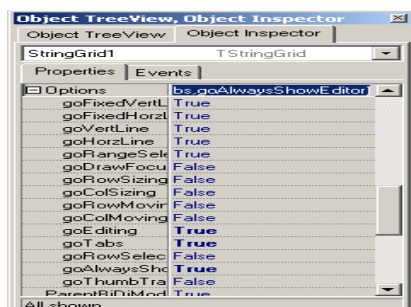
3.5.1-
jadval

Parametrlar va xossalar bo'limi	Hodisalar bo'limi
ColCount	5
FixedCols	0
RowCount	1
DefaultRowHeight	24
Height	24
DefaultColWidth	64
Width	328
Options . goEditing	True
Options . AlwaysShowEditing	True
Options .goTabs	True

Jadvalda ko'rsatilgan 3 ta **Options.goEditing**, **Options. AlwaysShowEditing**, **Options.goTabs** buyruqlarni hosil qilishimiz uchun parametrlar va xossalar bo'limidan "Options" parametirining "+" ishorasini "-" ishoraga o'tkazamiz (jadval ustiga sichqoncha tugmasini bir marta bosganimizdan so'ng, aytilgan topshiriqni bajarishimiz mumkin).



3.5.4-rasm. StringGrid1(abc) tugmasidan foydalanish



3.5.5-rasm. Options xossasidan foydalanish

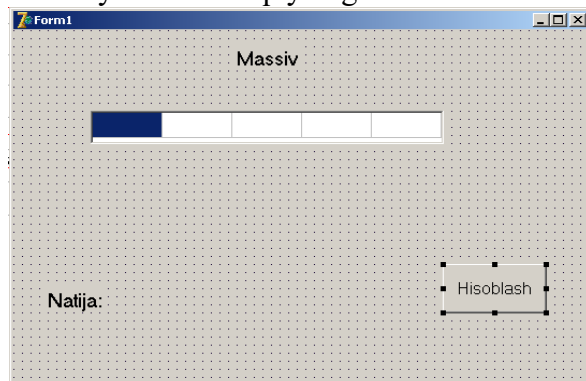
Hodisalar bo‘limining **goEditing** bandida 2 ta **False** va **True** buyrug‘i mavjud. **True** buyrug‘ini tanlasangiz katakchaga ma’lumot kiritgandan so‘ng dastur bilan bog‘lanish hosil qiladi, aks holda ya’ni **False** buyrug‘i tanlasangiz kiritilgan ma’lumot dastur bilan bog‘lashga ruxsat etmaydi.

Parametrlar va xossalar bo‘limidan “Options” parametirini “+” ishorasini “-” ishoraga o‘tkazganimizdan so‘ng oynani ko‘rinishi quyidagicha bo‘ladi:

2. “Label1” tugmasidan foydalanib “Massiv” so‘zini kiritamiz.

3. “Label2” tugmasiga “Natija” so‘zini kiritamiz.

Natijada oynaning umumiy ko‘rinishi quyidagicha bo‘ladi:



3.5.6-rasm. Dasturlashda “Button” tugmasidan foydalanish

2-bosqich.

1. Dasturlash maydoniga quyidagi dasturni kiritamiz:

unit Unit1; interface uses

**Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs, StdCtrls, Grids;**

Type

TForm1 =

class(TForm)

StringGrid1:

TStringGrid;

Label1: TLabel;

Button1:

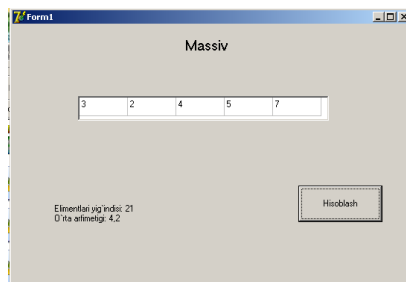
TButton;

```

    Label2:
    TLabel;
    procedure Button1Click(Sender:
    TObject); procedure
    StringGrid1Click(Sender: TObject);
    private
    { Private
    declarations }
    public
    { Public
    declarations }
    end;
var
    Form1:
    TForm1;
impleme
ntation
{$R *.dfm}
procedure TForm1.Button1Click(Sender:
TObject); var
a : array[1..5]
of integer;
summ: integer;
sr: real; i: integer; begin
    for i:= 1 to 5 do
        if Length(StringGrid1.Cells[i-1, 0]) <>0
        then a[i] :=
        StrToInt(StringGrid1.Cells[i-1,0])
        else a[i] := 0;
        summ := 0;
        for i :=1 to 5 do
            summ := summ + a[i]; sr :=
            summ / 5; Label1.Caption
            :=
            'Elementlari yig'indisi: ' + IntToStr(summ)
            + #13+ 'O'rta arifmetigi: ' +
            FloatToStr(sr); end;
    procedure TForm1.StringGrid1Click(Sender:
    TObject); var Key: Char;
    begin
        case Key of #8,'0'..'9' : ;
        #13:
        if      StringGrid1.Col      <
        StringGrid1.ColCount-1      then
            StringGrid1.Col := StringGrid1.Col
            + 1; else key := Chr(0);
end; end; end.

```

2. Dastur kiritilganidan so'ng, unga nom berib saqlaymiz.
3. Dastur ishini yakunlab ma'lum bir qiymatlarini yachyekalarga kiritamiz Oynani umumiy ko'rinishi:



3.5.7-rasm. Massiv elementlari yig'indisini va o'rta arifmetigini hisoblash dasturi

Izoh : Hosil bo'lgan dastur yordamida 5 ta elementdan tashkil topgan vektor elementlarining yig'indisi va o'rta arifmetigini topish mumkin. Buning uchun hosil qilingan dastur maydoniga yangi qiymat kiritish orqali natijani olishimiz mumkin.

2-usul.

1-masalani ikkinchi usul bilan hisoblaymiz.

1. Standart komponentalar palitrasidan 2 ta **button**, 1 ta **memo** va 1 ta **ComboBox** tugmalarini hosil qilamiz,
2. Sichqoncha tumachasini forma oynasiga ketma-ket ikki marta bosamiz va dasturni yuqori qismidagi

var

Form1:

TForm1;

Yozuvning

pastiga

a:Array[1..5]

of integer;

i:integer;

s,s1,s2:real;

yozuvlarni kiritamiz.

3. **"F12"** tugmasini bosgan holda forma oynasiga o'tamiz.

4. Sichqoncha tumachasini forma oynasiga ketma-ket ikki marata bosamiz va quyidagi:

procedure TForm1.FormCreate(Sender:

TObject); begin

i:=0;

ComboBo

x1.Clear;

end;

dasturni kiritamiz.

5. **"Button1"** ketma-ket ikki marta bosamiz va dasturlash maydoniga

begin

ComboBox1.Items.Add(ComboB

ox1.text);

i:=i+1;

a[i]:=StrToInt(ComboB

ox1.text);

ComboBox1.SetFocus;

end;

dasturni kiritamiz.

6. **"Button2"** ketma-ket ikki marta bosamiz va dasturlash maydoniga

begin

S:=0;

For i:=1 to 5 do s:=s+a[i]; Memo1.Clear;

```

Memo1.Lines.add('Yig'indi
    ='+'floattostr(s)); for i:=1 to 5 do
    s1:=s/5;
Memo1.Lines.add('O'rtacha='+'float
    tostr(s1));
end; end; end.

```

dasturni kiritamiz.

Dastur kodi va oynaning umumiy ko'rinishi:

unit Unit1; interface

uses

**Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs, StdCtrls;**

type

TForm1 =

class(TForm)

ComboBox1:

TComboBox;

Button1:

TButton;

Button2:

TButton;

Memo1: TMemo;

**procedure FormCreate(Sender:
TObject); procedure**

Button1Click(Sender: TObject);

**procedure Button2Click(Sender:
TObject); private**

{ Private

declarations }

public

{ Public

declarations }

end;

var

Form1:

TForm1;

a:Array[1..5]

of integer;

k,i,maxx,minn:

integer;

s,s1,s2:real;

implementation

n

{ \$R *.dfm }

**procedure TForm1.FormCreate(Sender:
TObject); begin**

i:=0;

ComboBo

x1.Clear;

end;

**procedure TForm1.Button1Click(Sender:
TObject); begin**

```

ComboBox1.Items.Add(ComboB
    ox1.text); i:=i+1;
a[i]:=StrToInt(ComboB
    ox1.text);
ComboBox1.SetFocus;

```

```

procedure TForm1.Button2Click(Sender:
TObject); begin

```

```

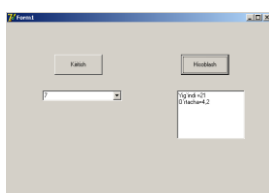
    S:=0;
    For i:=1 to 5 do s:=s+a[i];
    Memo1.Clear;
    Memo1.Lines.add('Yig'indi
    ='+'floattostr(s)); for i:=1 to 5 do
    s1:=s/5;
    Memo1.Lines.add('O'rtacha='+'floattostr(s1));

```

```

end; end; end.

```



3.5.8-rasm. Massiv elementlari yig'indisini va o'rta arifmetigini hisoblash dasturi

Izoh : Dastur natijasini olish uchun “**ComboBox**” maydoniga qiymat kiritib, “**Kiritish**” tugmasi tanlanadi. Yig'indisini va o'rta arifmetigini hisoblash uchun esa “**Hisoblash**” tugmasi tanlanadi.

3-usul.

1-masalani uchunchi usul bilan hisoblaymiz.

1. Standart komponentalar palitrasidan 2 ta **button**, 1 ta **memo**, 1 ta **edit**, va 1 ta **ListBox** tugmalarini hosil qilamiz.

2. Sichqoncha tumachasini forma oynasiga ketma-ket ikki marata bosamiz va dasturni yuqori qismidagi

var

Form1: TForm1; Yozuvning pastiga **a:Array[1..5] of integer; i:integer;**

key,s,s1:real;

yozuvlarni kiritamiz.

3. “F12” tugmasini bosgan holda forma oynasiga o'tamiz.

4. Sichqoncha tumachasini forma oynasiga ketma-ket ikki marata bosamiz va quyidagi:

```

procedure TForm1.FormCreate(Sender:

```

```

TObject); begin

```

```

    i:=0;

```

```

    ListBox1.Clear;

```

dasturni kiritamiz.

5. “Button1” ketma-ket ikki marta bosamiz va dasturlash maydoniga

```

begin

```

```

    ListBox1.Items.Add(E

```

```

    dit1.text);

```

```

        i:=i+1;

```



```

        a[i]:=StrToInt(E
        dit1.text);
        Edit1.SetFocus;
end;

```

dasturni kiritamiz.

6. **“Button2”** ketma-ket ikki marta bosamiz va dasturlash maydoniga

```

begin
    S:=0;
    For i:=1 to 5 do
        s:=s+a[i];
        Memo1.Clear;
        Memo1.Lines.add('Yig‘indi
        =' + floattostr(s)); for i:=1 to 5 do
        s1:=s/5;
        Memo1.Lines.add('O‘rtacha=' + float
        tostr(s1));
    end; end; end.

```

dasturni kiritamiz.

Dastur kodi va oynaning umumiy ko‘rinishini keltiramiz:

```

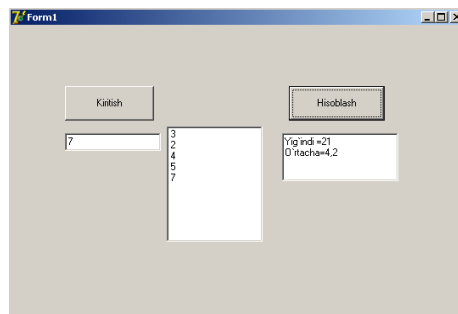
unit Unit1; interface uses
    Windows, Messages, SysUtils, Variants, Classes, Graphics,
    Controls, Forms, Dialogs, StdCtrls;
type
    TForm1 =
        class(TForm)
            Edit1: TEdit;
            ListBox1:
                TListBox;
            Button1:
                TButton;
            Button2:
                TButton;
            Memo1:
                TMemo;
        procedure FormCreate(Sender:
        TObject); procedure
        Button1Click(Sender: TObject);
        procedure Button2Click(Sender:
        TObject); procedure
        Edit1Change(Sender: TObject); private
        { Private
        declarations }
        public
        { Public
        declarations }
        end;
var
    Form1: TForm1; a:Array[1..5] of integer;
    k,i,maxx,minn:integer; key,s,s1,s2:real;
implementation
{$R *.dfm}
procedure TForm1.FormCreate(Sender: TObject);

```

```

ListBo
x1.Cl
ear;
end;
procedure TForm1.Button1Click(Sender:
TObject); begin
ListBox1.Items.Add(E
dit1.text);
i:=i+1;
a[i]:=StrToInt(E
dit1.text);
Edit1.SetFocus;
end;
procedure TForm1.Button2Click(Sender:
TObject); begin
S:=0;
For i:=1 to 5 do s:=s+a[i]; Memo1.Clear;
Memo1.Lines.add('Yig'indi
='+floattostr(s)); for i:=1 to 5 do
s1:=s/5;
Memo1.Lines.add('O'rtaacha='+floattostr(s1));
end;
procedure TForm1.Edit1Change(Sender:
TObject); begin
If key=13 Then
Button1.SetFocus; end;
end.

```



3.5.9-rasm. Massiv elementlari yig'indisini va o'rta arifmetigini hisoblash dasturi
Izoh : Hosil bo'lgan ko'rinichdagi **edit** qatoriga qiymat kiritib kiritish tugmasini bossak, **ListBox** maydonida paydo bo'ladi. Uning qiymatini hisoblash uchun esa, hisoblash tugmasini tanlash kifoya.

2-mi $A(5,5)$ massiv elementlarini yig'indisi va o'rta arifmetigini topish dasturi
sol

.

tuzamiz.

Ikki o'lovli massiv elementalri yig'indisi va o'rta arifmetigini hisoblash uchun **button**, **memo** va **StringGrid** tugmalaridan foydalanamiz.

Dastur kodi va oynaning umumiy ko'rinishini keltiramiz:

unit Unit1; interface uses

**Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs, StdCtrls, Grids;**

type

TForm1 =

class(TForm)

StringGrid1:

TStringGrid;

Button1:

TButton;

Memo1:

TMemo;

procedure Button1Click(Sender:

TObject); private

{ Private

declarations }

public

{ Public

declarations }

end;

var

Form1: TForm1;

implementation

{ \$R *.dfm }

procedure TForm1.Button1Click(Sender:

TObject); Var i,j,cod:integer;

A:array[1..5,1..

5] of Real;

S:real;

s1:String;

begin

For i:=1 to 5 do For j:=1 to 5 do

Val(StringGrid1.cells[i-1,j-

1],a[i,j],cod); S:=0;

For i:=1 to 5 do For j:=1 to 5 do

s:=s+a[i,j];

Str(s:7:2,s1);

Memo1.Clear;

Memo1.Lines.add('Yig'in

di =' +s1); s:=s/25;

Str(s:7:2,s1);

Memo1.Lines.add('O'rtac

ha=' +s1); end; end.

6	2	2	1	2
4	5	6	7	2
2	4	4	7	8
8	9	0	1	2
9	4	5	4	1

Yig'indi = 105.00
O'rtacha = 4.20

3.5.10-rasm. Massiv elementlari yig'indisini va o'rta arifmetigini hisoblash dasturi

Izoh: hosil bo'lgan dastur yordamida 5 ta ustun va 5 ta satrdan tashkil topgan massiv elementlarining yig'indisini va o'rta arifmetigini hisoblaydi.

3-Mis $A(5)$ massiv elementlarining eng kichigi va element turgan o'rnini topish ol.

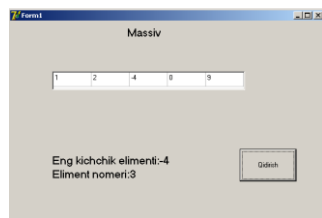
dasturini
tuzamiz.

Dastur kodi va oynaning umumiy ko‘rinishini keltiramiz:

**Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs, StdCtrls, Grids;**

```
type
  TForm1 =
    class(TForm)
      StringGrid1:
        TStringGrid;
      Label1: TLabel;
      Label2:
        TLabel;
      Button1:
        TButton;
      procedure Button1Click(Sender:
        TObject); procedure
        StringGrid1Click(Sender: TObject);
    private
      { Private
        declarations }
    public
      { Public
        declarations }
    end;
var
  Form1: TForm1; implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender:
  TObject); const
  k=5;
var
  a:array[1..k]of integer; min:integer;
  for i:=1 to k do
  a[i]:=StrToInt(StringGrid1.Cells[i-1,0]);
  min:=1;
  for i:=2 to k do
  if a[i]< a[min]then min:=i;
  label2.caption:='Eng kichik Elementi:'
  +IntToStr(a[min])+#13+'Element nomeri:'+
  IntToStr(min); end;
  procedure TForm1.StringGrid1Click(Sender:
  TObject); var Key: Char;
  begin
    case Key of #8,'0'..'9': ;

#13:
  if      StringGrid1.Col      <
  StringGrid1.ColCount-1      then
  StringGrid1.Col:= StringGrid1.Col
  + 1; else key := Chr(0);
  end; end; end.
```



3.5.11-rasm. Massiv elementlari ustida amallari

Izoh : Hosil bo'lgan dastur yordamida 5 ta elementdan tashkil topgan vektor elementlarining eng kichigini va element nomerini topish mumkin. Buning uchun hosil qilingan dastur maydoniga yangi qiymat kiritish orqali natijani olishimiz mumkin.

5-Amaliy mashg'ulot.

Mavzu: Borland C++ Builderda funksiyalarga doir dastur tuzish.

Foydalanuvchi Funksiyalari.

Funksiyalarni ta'riflash va ularga murojaat qilish. Funksiya ta'rifida funksiya nomi, tipi va formal parametrlar ro'yhati ko'rsatiladi. Formal parametrlar nomlaridan tashqari tiplari ham ko'rsatilishi shart. Formal parametrlar ro'yhati funksiya signaturasi deb ham ataladi. Funksiya ta'rifi umumiy ko'rinishi quyidagichadir:

Funksiya tipi funksiya nomi(formal_parametrlar_ta'rifi)

Formal parametrlarga ta'rif berilganda ularning boshlang'ich qiymatlari ham ko'rsatilishi mumkin. Funksiya qaytaruvchi ifoda qiymati funksiya tanasida `return <ifoda>` ; operatori orqali ko'rsatiladi.

Misol:

```
Float min(float, float b)
{ if (a<b) return a; return b;
}
```

Funksiyaga murojaat qilish quyidagicha amalga oshiriladi: Funksiya nomi (haqiqiy parametrlar ruyhati)

Haqiqiy parametr ifoda ham bo'lishi mumkin. Haqiqiy parametrlar qiymati hisoblanib mos formal parametrlar o'rnida ishlatiladi.

Misol uchun yuqoridagi funksiya qo'yidagicha murojaat qilish mumkin: `Int x=5,y=6,z; z=min(x,y)` ekin `int z=Min(5,6)` ekin `int x=5; int z=min(x,6)`

Funksiya ta'rifida formal parametrlar initsializatsiya qilinishi, ya'ni boshlang'ich qiymatlar ko'rsatilishi mumkin. Funksiyaga murojaat qilinganda biror haqiqiy parametr ko'rsatilmasa, uning urniga mos formal parametr ta'rifida ko'rsatilgan boshlang'ich qiymat ishlatiladi.

Misol uchun:

```
Float min(float a=0.0, float b)
{ if (a<b) return a; return b;
}
```

Bu funksiya yuqorida ko'rsatilgan murojaat usullaridan tashqari quyidagicha murojaat qilish mumkin:

`Int y=6,z; z=min(,y)` ekin `int z=Min(,6);`

Agar funksiya hech qanday qiymat qaytarmasa uning tipi void deb ko'rsatiladi. Misol uchun:

`Void print;`

```
{ Cout<<("\n Salom!");
};
```

Bu funksiyaga Print; shaklida murojlat qilish ekranga Salom! Yozilishiga olib keladi. Qiymat qaytarmaydigan funksiya formal parametrlarga ega bo'lishi mumkin:

```
Void Print_Baho(Int baho);
{
Switch(baho)
{case 2:Cout<<("\n emon");break; case 3:Cout<<("\n
urta");break; case 4:Cout<<("\n yahshi");break; case
5:Cout<<("\n a'lo");break;
default: Cout<<("\n baho notugri kiritilgan");
};
```

Bu funksiyaga Print_Baho(5) shaklida murojaat qilish ekranga a'lo so'zi yozilishiga olib

keladi.

Agar programmada funksiya ta'rifi murojaatdan keyin berilsa, yoki funksiya boshqa faylda

joylashgan bo'lsa, murojlatdan oldin shu funksiyaning prototipi joylashgan bulishi kerak. Prototip funksiya nomi va formal parametrlar tiplaridan iborat bo'ladi. Formal parametrlar nomlarini berish shart emas.

Misol uchun $y = \min(a,b) + 2 * \max(c,d)$ ifodani hisoblashni kuramaz: #Include <iostream.h>

```
int max(int a,int b)
{if (a<b) return a;else
return b}; void main()
{int a,b,c,d,y; int min(int ,int);
Cin>>("\n %f%f%f%f",&a,&b,&c,&d);
y=min(a,b)+2*max(c,d);
Cout<<("\n %f",y);
};
int min(int a,int b)
{if (a<b) return b;else return a};
```

Funksiyaga parametrlar uzatish. Funksiyaga parametrlar qiymat buyicha uzatiladi va quyidagi bosqichlardan iborat bo'ladi:

1. Funksiya bajarishga tayyorlanganda formal parametrlar uchun hotiradan joy ajratiladi, ya'ni formal parametrlar funksiyalarning ichki parametrlariga aylantiriladi. Agar parametr tipi float bo'lsa double tipidagi ob'ektlar hosil buladi, char va shortint bulsa int tipidagi ob'ektlar yaratiladi.

2. Haqiqiy parametrlar sifatida ishlatilgan ifodalar qiymatlari hisoblanadi.

3. Haqiqiy parametrlar ifodalar qiymatlari formal parametrlar uchun ajratilgan hotira qismlariga yoziladi. Bu jarayonda float tipi double tipiga, char va shortint tiplari int tipiga keltiriladi.

4. Funksiya tanasi ichki ob'ektlar – parametrlar yordamida bajariladi va qiymat chaqirilgan joyga qaytariladi.
5. Haqiqiy parametrlar qiymatlariga funksiya hech qanday ta'sir o'tkazmaydi.
6. Funksiyadan chiqishda formal parametrlar uchun ajratilgan hotira qismlari bo'shatiladi.

C++ tilida chaqirilgan funksiya chaqiruvchi funksiyadagi o'zgaruvchi qiymatini uzgartira olmaydi. U faqat o'zining vaqtinchalik nushasini o'zgartirishi mumkin holos.

6-Amaliy mashg'ulot.

Mavzu: Borland C++ Builderda grafika, multimedia va animatsiyalar.

C++ Builder muhitida chizish sirti - TCanvas sinfi programma ishlash paytida rasm chizish imkonini beradi. Bu sinf ob'ekti sirt bo'yicha ko'chish, grafik primitivlar chizish, rasmlarni va sirtning biror qismini nusxalash, hamda matnni chop qilish imkonini beruvchi xossa va metodlarni o'z ichiga oladi.

Har bir Canvas xossasiga ega komponenta o'z navbatida qalam, kist va shrift obektlarini tarkibiga oladi va mos ravishda Pen, Brush va Font xossalariga ega bo'ladi.

Pen xossasi rangga (Canvas->Pen-> Color), chizishning piksellerdagi qalinlikka (Canvas->Pen-> Width), chizilayotgan chiziq toifasiga (Canvas->Pen->Style) ega. Chiziq toifasi quyidagi qiymatlarni qabul qilishi mumkin:

psSolid – uzluksiz chiziq (kelishuv bo'yicha); psDash – tire belgilaridan

hosil bo'lgan chiziq; psDot –

nuqtalardan tashkil topgan chiziq;

psDashDot – nuqta va tire ketma-ketliklaridan iborat chiziq;

psDashDotDot – tire va nuqtalardan ketma-ketliklaridan

iborat chiziq; psClear – ko'rinmas chiziq;

psInsideFrame – chizish sirtini chegaralovchi to'g'ri to'rtburchak ichidagi chiziq.

Brush xossasi geometrik shakllar, masalan, to'g'ri to'rtburchak va ellips ichini to'ldirish naqshini aniqlaydi. U quyidagi xossalarga ega:

Canvas->Brush->Color – kist ranggi;

Canvas->Brush->Style – kist toifasini aniqlaydi va u quyidagi qiymatlarni qabul qilishi mumkin:

bsSolid – berilgan rang bilan shakl yuzasini to'liq

bo'yaladi; bsClear -shakl yuzasi bo'yalmaydi;

bsHorizontal - shakl yuzasi parallel chiziqlar bilan

to'ldiriladi; bsVertical - shakl yuzasi vertical chiziqlar bilan

to'ldiriladi; bsFDiagonal - shakl yuzasi yuqoriga qaragan

chiziqlar bilan to'ldiriladi;

bsFDiagonal - shakl yuzasi yuqoriga pastga chiziqlar bilan to'ldiriladi; bsCross - shakl yuzasi to'r bilan to'ldiriladi; bsDiagCross - shakl yuzasi egri chiziqlardan hosil bo'lgan to'r bilan to'ldiriladi. Canvas obektining muhim xossalariidan biri Canvas->Pixels[x][y] xossasi bo'lib, u ko'rsatilgan koordinatadagi piksel rangini aniqlaydi. Bu xossa qiymatini o'qish va unga qiymat yozish mumkin.

Geometric shakllar chizish uchun quyida keltirilgan funksiyalardan foydalanish mumkin:

Arc(int X1, int Y1, int X2, int Y2, int X3, int Y3, int X4, int X4) – yoy chizish. Bu erda (X1,Y1) va (X2,Y2) – mos ravishda yoy chiziladigan to'rtburchak sohaning chap yuqori va o'ng past uchlari koordinatasi. (X3,Y3) va (X4,Y4) nuqtalar mos holda yoy boshlanishi va oxiri koordinatasi.

Chord(int X1, int Y1, int X2, int Y2, int X3, int Y3, int X4, int X4) – ellips vatarini chizish. Bu erda (X1,Y1) va (X2,Y2) – mos ravishda yoy chiziladigan to'rtburchak sohaning chap yuqori va o'ng past uchlari koordinatasi. (X3,Y3) va (X4,Y4) nuqtalar mos holda vatar boshlanishi va oxiri koordinatasi.

Ellipse(int X1, int Y1, int X2, int Y2) – rang bilan to'ldirilgan ellipsni chizish. Bu erda (X1,Y1) va (X2,Y2) – mos ravishda yoy chiziladigan to'rtburchak sohaning chap yuqori va o'ng past uchlari koordinatasi.

Rectangle(int X1, int Y1, int X2, int Y2) – rang bilan to'ldirilgan to'g'rito'rtburchakni chizish. Bu erda (X1,Y1) va (X2,Y2) – mos ravishda yoy chiziladigan to'rtburchak sohaning chap yuqori va o'ng past uchlari koordinatasi.

Canvas xossalariidan foydalangan holda uy rasmini chizishda ellips, to'g'rito'rtburchak, ko'pburchak shakllari ishlatilgan. Rangni boshqarish, shakl yusasini to'ldirish uchun qalam va kist xossalari ishlatilgan. Rasm olchamlari forma o'lchamiga mos ravishda o'zgaradi.

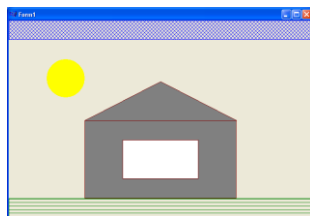
```
#include <vcl.h> #pragma hdrstop
#include "Unit1.h"
//-----
#pragma
package(smart_init
) #pragma resource
"*.dfm" TForm1
*TForm1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//-----
void __fastcall TForm1::FormResize(TObject *Sender)
{
int w, h, wm, hm; Form1->Refresh();
wm=Form1->ClientWidth;
w=wm/8; hm=Form1-
>ClientHeight; h=hm/10;
// Osmon
Form1->Canvas->Brush-
>Color=clBlue; Form1->Canvas-
>Brush->Style=bsDiagCross; Form1-
>Canvas->Pen->Color=clBlue;
```

```

Form1->Canvas->Rectangle(0,0,wm,h);
// Maysalar
Form1->Canvas->Brush->Color=clGreen;

Form1->Canvas->Brush-
>Style=bsHorizontal; Form1-
>Canvas->Pen->Color=clGreen;
Form1->Canvas->Rectangle(0,hm-
h,wm,hm);
// Quyo'sh
Form1->Canvas->Brush-
>Color=clYellow; Form1->Canvas-
>Brush->Style=bsSolid; Form1-
>Canvas->Pen->Color=clYellow;
Form1->Canvas-
>Ellipse(w,2*h,2*w,2*h+w);
// Uycha
Form1->Canvas->Brush-
>Color=clGray;          Form1-
>Canvas->Brush-
>Style=bsSolid;          Form1-
>Canvas->Pen-
>Color=clMaroon;
Form1->Canvas->Rectangle(2*w, hm-5*h,
6*w, hm-h); TPoint point[4];
point[0].x=2*w; point[0].y=hm-
5*h; point[1].x=4*w;
point[1].y=hm-7*h;
point[2].x=6*w; point[2].y=hm-
5*h; point[3].x=2*w;
point[3].y=hm-5*h; Form1-
>Canvas->Polygon(point,3);
Form1->Canvas->Brush-
>Color=clWhite; Form1-
>Canvas->Brush-
>Style=bsSolid; Form1-
>Canvas->Pen-
>Color=clMaroon;
Form1->Canvas->Rectangle(3*w, hm-4*h, 5*w, hm-2*h);
}

```



Programma ishga tushishi bilan Formmaning OnResize hodisasi ro'y beradi va ekranda uycha rasmi paydo bo'ladi.

2.2. Matematik funksiyalarning grafigini chizish

C++ Builder muhitida grafik shakllarni chizish Canvas komponentasi vositsida amalgam oshiriladi. Ayrim visual komponentalar bu komponentaga ega. Masalan, TForm, Image, PaintBox va boshqalar.

Formaning (Form1) grafik shakllar chizish sohasi bu mijoz sohasi hisoblanadi va uning 102

o'lchami Form1->ClientWidth (gorizontaliga) va Form1-> ClientHeight (vertikaliga) bilan aniqlanadi.

Grafika sohasi adreslanuvchi nuqtalarning to'g'ri burchakli massiv ko'rinishida bo'ladi va ixtiyoriy tasvir yonib yoki o'chib turgan piksellar (tasvirning minimal elementi) kompozitsiyasidan hosil bo'ladi. Bu nuqtalar ikkita butun son: nx - nuqtaning gorizental nomeri va ny nuqtaning vertikal nomeri bilan adreslanadi:

$$0 \leq nx \leq nx_Max;$$

$$0 \leq ny \leq ny_Max,$$

bu erda nx_Max=Form1->ClientWidth va ny=Form1->ClientHeight.



Grafika sohasining chap yuqori burchagi (0,0) koordinataga ega bo'ladi. (nx,ny) qurilma koordinatalari ham deyiladi va ular faqat butun qiymatlarni qabul qiladi.

Kompyuter grafikasida yana ikkita koordinata tizimi qabul qilingan. Birinchisi (px, py)- ekran koordinata tizimi bo'lib, unda px- gorizental bo'yicha ekrandagi masofa, py-gorizental bo'yicha. Bu yerda koordinata o'qlari millimetr va duymlarda o'lchanadi. Ikkinchi koordinata tizimi - dunyoviy (olam) koodinata tizimidir. U (x,y) dekart tizimi bo'lib, programma tuzuvchisi tomonidan aniqlanadi va tasvirlash qurilmasiga bog'liq bo'lmaydi:

$$Xmin < x < Xmax \quad Ymin < y < Ymax$$

Dekart koordinatalar tizimida X va Y o'zgarish diapazonlari (Xmin, Xmax, Ymin, Ymax) mavhum matematik ikki o'lchamli fazoning to'g'ri burchakli sohasini aniqlaydi. Bu sohani qurilma koordinatasiga akslantirish quyidagicha amalga oshiriladi:

$$nx = \text{Round}((x - Xmin) / (Xmax - Xmin)) * nx_Max; \quad ny = \text{Round}((y - Ymin) / (Ymax - Ymin)) * ny_Max,$$

bu yerda (x,y)- dekart koordinatasidagi nuqta va uning ekrandagi koordinatasi (nx,ny) bo'ladi.

Grafik kursor. Grafik kursor matn kursori bajaruvchi ishni bajaradi, lekin u ekranda ko'rinmaydi. Ma'lumki matn kursori ekrandagi belgi o'rnini (80*25 bo'lganida) ko'rsatadi va bu o'rinda belgi chop qilinganda avtomatik ravishda bir o'rin o'ngga suriladi. Grafik kursor esa chiqariluvchi grafik shaklning boshlang'ich koordinatasini ko'rsatadi va uni keyingi joyga (nx,ny) nuqtaga ko'chirish uchun maxsus funksiya ishlatiladi: Form1->Canvas->MoveTo(nx,ny);

Chiziqlarni chizish. Sohada chiziqni (kesmani) chizish uchun Form1->Canvas->LineTo() funksiyasidan foydalaniladi. Masalan, (x1,y1) va (x2,y2) nuqtalarni tutashtiruvchi kesma chizish uchun quyidagi amallar bajarilishi kerak:

Form1->Canvas->MoveTo(x1,y1); Form1->Canvas->LineTo(x2,y2);

Ekranda ko'p miqdordagi sinq chiziqlardan tashkil topgan shaklni chizish uchun Canvas->Polyline(Jadval, n);

funksiyasidan foydalaniladi. U berilgan sondagi sonlar juftligi majmuasi bilan aniqlangan sinq chiziqni chizadi. n parametri sinq chiziq tugun nuqtalari soni. Jadval parametri TPoint turida bo'lib, grafik soha nuqta koordinatasini aniqlovchi strukturalar massivdir. Sinq chiziq tugun nuqtalari Jadval massivi sifatida beriladi.

Misol3. Quyida PaintBox1 komponentasi sohasida $y=\cos(x)$ funksiya grafigini chizish funksiyasi keltirilgan.

```
#include <vcl.h> #include <math.h> #pragma hdrstop #include "Unit1.h"
#pragma package(smart_init) #pragma resource "*.dfm" TForm1 *Form1;
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    const float Pi=3.1415;
    int Xe0,Ye0,Xe,Ye,      // Ekran koordinatalari
        Rect_X,Rect_Y, // Koordinata chegaralarining soha
        chegarasidan farqi Mashtab;      // Soha koordinatasining
        Haqiqiysiga nisbati, Mashtab
    float h,X,Y;      // y=f(x) funktsiya
    va h qadam Mashtab=80;      //
    Mashtabni tanlash
    Rect_X =10;      //
    Chegaralar Rect_Y
    =10;
    h=0.1;      // h qadam
    Xe0 = PaintBox1->Width/2; //Koordinata markazi - Soha
    markazi tanlandi Ye0 = PaintBox1->Height/2;
    PaintBox1->Canvas->MoveTo(Rect_X,Ye0);
    PaintBox1->Canvas->LineTo(PaintBox1->Width-Rect_X,Ye0);
    //OX-o'qi PaintBox1->Canvas->MoveTo(Xe0,Rect_Y);
    PaintBox1->Canvas->LineTo(Xe0,PaintBox1->Height-
    Rect_Y); // OY-o'qi X=-Pi-h;
    X=X+h;
    Y=cos(X);
    Xe=Xe0+(int)(M
    ashtab*X);
    Ye=Ye0-
    (int)(Mashtab*Y)
    ;
    PaintBox1->Canvas->MoveTo(Xe,Ye); //Grafik
    kursorni o'rnatish PaintBox1->Canvas->Pen-
    >Color=clRed;
    do
    {
        X=X+h;
        Y=cos(X);
        Xe=Xe0 +
        (int)(Mashtab*X);
        Ye=Ye0 -
        (int)(Mashtab*Y);
        if (Xe>Rect_X && Xe < PaintBox1->Width-
        Rect_X && Ye>Rect_Y &&
        Ye<PaintBox1->Height-Rect_Y)
        PaintBox1->Canvas->LineTo(Xe,Ye); //Koordinata chegarasida chizish
    }
    while (X<=Pi);
```

}
 Programma ishlashi natijasida Formadagi PaintBox1 komponenta sohasida quyidagi chizma paydo bo'ladi.



2.3. $y=x^2-1$ funksiyaning grafigini chizish

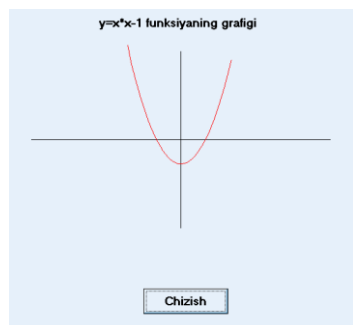
Misol4. Quyida PaintBox1 komponentasi sohasida $y=x^2-1$ funktsiya grafigini chizish funktsiyasi keltirilgan.

```
#include <vcl.h> #include <math.h> #pragma hdrstop #include "Unit1.h"
#pragma package(smart_init) #pragma resource "*.dfm" TForm1 *Form1;
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    const float Pi=3.1415;
    int Xe0,Ye0,Xe,Ye, // Ekran koordinatalari
        Rect_X,Rect_Y, // Koordinata chegaralarining soha
        chegarasidan farqi Mashtab; // Soha koordinatasining
        Haqiqiysiga nisbati, Mashtab
    float h,X,Y; // y=f(x) funktsiya va h qadam
    Mashtab=80; //
    Mashtabni tanlash Rect_X
    =10; //
    Chegaralar Rect_Y =10;
    h=0.1; // h qadam
    Xe0 = PaintBox1->Width/2; //Koordinata markazi - Soha
    markazi tanlandi Ye0 = PaintBox1->Height/2;
    PaintBox1->Canvas->MoveTo(Rect_X,Ye0);
    PaintBox1->Canvas->LineTo(PaintBox1->Width-Rect_X,Ye0);
    //OX-o'qi PaintBox1->Canvas->MoveTo(Xe0,Rect_Y);
    PaintBox1->Canvas->LineTo(Xe0,PaintBox1->Height-
    Rect_Y); // OY-o'qi X=-Pi-h;
    X=Xe0
    +h;
    Y=
    pow(
    X,2)-
    1;
    Xe=Xe0+(int)(Mashtab*
    X);
    Ye=Ye0-
    (int)(Mashtab*Y)
    ;
```

```

PaintBox1->Canvas->MoveTo(Xe,Ye); //Grafik
kursorni o'rnatish PaintBox1->Canvas->Pen-
>Color=clRed;
do
{
X=X+h;
Y=pow(X,2)-1;
Xe=Xe0 + (int)(Mashtab*X); Ye=Ye0 -
(int)(Mashtab*Y);
if (Xe>Rect_X && Xe < PaintBox1->Width-
Rect_X && Ye>Rect_Y &&
Ye<PaintBox1->Height-Rect_Y)
PaintBox1->Canvas->LineTo(Xe,Ye); //Koordinata chegarasida chizish
}
while (X<=Pi);
}
Programma ishlashi natijasida Formadagi PaintBox1 komponenta sohasida
quyidagi chizma paydo bo'ladi.

```



7-Amaliy mashg'ulot.

Mavzu: Borland C++ Builderda satriy kattaliklar.

Satrlar va ular ustida amallar

Standart C++ tili ikki xildagi belgilar majmuasini qo'llab-quvvatlaydi. Birinchi toifaga, an'anaviy, "tor" belgilar deb nomlanuvchi 8-bitli belgilar majmuasi kiradi, ikkinchisiga 16-bitli "keng" belgilar kiradi. Til kutubxonasida har bir guruh belgilari uchun maxsus funktsiyalar to'plami aniqlangan.

C++ tilida satr uchun maxsus tur aniqlanmagan. Satr char turidagi belgilar massivi sifatida qaraladi va bu belgilar ketma-ketligi satr terminatori deb nomlanuvchi nol kodli belgi bilan tugaydi ('0'). Odatda, nol-terminator bilan tugaydigan satrlarni ASCIIZ –satrlar deyiladi. Sart konstanta deb qo'shtirnoqlar ichiga olingan belgilar ketma-ketligiga aytiladi:

“Ushbu belgilar ketma-ketligiga satr deyiladi.”

Quyidagi jadvalda C++ tilida belgi sifatida ishlatilishi mumkin bo'lgan konstantalar to'plami keltirilgan.

Belgilar sinflari	Belgi konstantalar
Katta harflar	'A' ... 'Z', 'A' ... 'Я'
Kichik harflar	'a' ... 'z', 'a' ... 'я'
Raqaamlar	'0' ... '9'
Bo'sh joy	gorizontal tabulyatsiya (ASCII kodi 9), satrni o'tkazish (ASCII kodi 10), vertikal tabulyatsiya (ASCII kodi 11), formani o'tkazish (ASCII kodi 12), karetkani qaytarish (ASCII kodi 13)

Punktuatsiya (ajratuvchilar)	belgilari	! " # \$ % & ' () * + , - . / : ; < = > ? @ [\] ^ _ { } ~
Boshqaruv belgilari		ASCII kodi 0...1Fh oralig'ida va 7Fh bo'lgan belgilar
Probel		ASCII kodi 32 bo'lgan belgi
O'n oltilik raqamlar		'0'...'9', 'A'...'F', 'a'...'f'

Satr massivi e'lon qilinishida satr oxiriga terminator qo'yilishini va natijada satrga qo'shimcha bitta bayt qo'shilishi inobatga olinishi kerak:

```
char satr[10];
```

Ushbu e'londa satr satri uchun jami 10 bayt ajratiladi – 9 satr hosil qiluvchi belgilar uchun va 1 bayt terminator uchun.

Satr o'zgaruvchilari e'lon qilinishida boshlang'ich qiymatlar qabul qilishi mumkin. Bu holda kompilyator avtomatik ravishda satr uzunligi hisoblaydi va satr oxiriga nol-terminatorni qo'shib qo'yadi:

```
char Hafta_kuni[]="Juma";
```

```
Ushbu e'lon quyidagi e'lon bilan
```

```
ekvivalent: char Hafta_kuni[] =
```

```
{'J','u','m','a','\0'};
```

Satr qiymatini o'qishda oqimli o'qish operatori ">>" o'rniga getline() funktsiyasini ishltagan ma'qul hisoblanadi, chunki oqimli o'qishda probellar inkor qilinadi (garchi ular satr belgisi hisoblansa ham) va o'qilayotgan belgilar ketma-ketligi satrdan "oshib" ketganda ham belgilarni kiritish davom etishi mumkin. Natijada satr o'ziga ajratilgan o'lchamdan ortiq belgilarni qabul qilishi mumkin. Shu sababli, getline() funktsiyasi ikkita parametrga ega bo'lib, birinchi parametr o'qish amalga oshirilayotgan satrga ko'rsatgich, ikkinchi parametrdan esa kiritilishi kerak bo'lgan belgilar soni ko'rsatiladi. Satrni getline() funktsiyasi orqali o'qishga misol ko'raylik:

```
#include <iostream.h> int main()
```

```
{
char satr[6];
cout<<"Satrni
kiriting: "<<"\n";
cin.getline(satr,6);
cout<<"Siz kiritgan
satr: "<<satr; return 0;
}
```

Programmada satr satri 5 ta belgini qabul qilishi mumkin, ortiqchalari tashlab yuboriladi. getline() funktsiyasiga murojaatda ikkinchi parametr qiymati o'qilayotgan satr uzunligidan katta bo'lmasligi kerak.

Satr bilan ishlaydigan funktsiyalarning aksariyati string.h kutubxonasida jamlangan. Nisbatan ko'p ishlatiladigan funktsiyalarning tavsifini keltiramiz.

Satr uzunligini aniqlash funktsiyalari

Satrlar bilan ishlashda, aksariyat hollarda satr uzunligini bilish zarur bo'ladi. Buning uchun string.h kutubxonasida strlen() funktsiyasi aniqlangan bo'lib, uning sintaksisi quyidagicha bo'ladi:

```
size_t strlen (const char* string)
```

Bu funktsiya uzunligi hisoblanishi kerak bo'lgan satr boshiga ko'rsatgich bo'lgan yagona parametrga ega va u ishlash natijasi sifatida ishorasiz butun sonni qaytaradi. strlen() funktsiyasi satrning real uzunligidan bitta kam qiymat qaytaradi, ya'ni nol-terminator o'rni hisobga olinmaydi. Xuddi shu maqsadda sizeof() funktsiyasidan ham foydalanish mumkin va u strlen() funktsiyasidan farqli ravishda satrning real uzunligini qaytaradi. Quyida keltirilgan misolda satr uzunligini hisoblashning har ikkita varianti

```
keltirilgan: #include <iostream.h>
```

```
#include
<string.h
> int
main()
{
char Str[]="1234567890";
cout
<<"strlen(Str)="<<strlen(Str)<<
endl;
cout<<"sizeof(Str)="<<sizeof(S
tr)<<endl; return 0;
}
```

Programma ishlashi natijasida ekranga strlen(Str)=10

sizeof(Str)=11 xabarlari chiqadi.

Odatda sizeof() funksiyasidan getline() funksiyasining ikkinchi argumenti sifati ishlatiladi va satr uzunligini yaqqol ko'rsatmaslik imkonini beradi:

```
cin.getline(Satr, sizeof(Satr));
```

Satrlarni nusxalash

Satr qiymatini biridan ikkinchisiga nusxalash mumkin. Buning uchun bir qator standart funksiyalar aniqlangan bo'lib, ularning tavsiflari quyida keltiramiz.

strcpy() funksiyasi prototipi

```
char* strcpy(char* str1, const char* str2)
```

ko'rinishga ega va bu funktsiya str2 ko'rsatib turgan satrdagi belgilarni str1 ko'rsatib turgan satrga baytma-bayt nusxalaydi. Nusxalash str2 ko'rsatib turgan satrdagi nol-terminal uchraguncha davom etadi. Shu sababli, str2 satr uzunligi str1 satr uzunligidan katta emasligiga ishonch hosil qilish kerak, aks holda berilgan sohasida (segmentida) str1 satrdan keyin joylashgan berilganlar "ustiga" str2 satrning "ortiqcha" qnomi yozilishi mumkin.

Navbatdagi programma qnomi "Satrni nusxalash!" satrini Str satrga nusxalaydi:

```
char Str[20];
strcpy(Str, "Satrni nusxalash!");
```

Zarur bo'lganda satrning qaysidir joyidan boshlab, oxirigacha nusxadash mumkin. Masalan, "Satrni nusxalash!" satrini 8 belgisidan boshlab nusxa olish zarur bo'lsa, uni quyidagicha yechish mumkin:

```
#include <iostream.h> #include
<string.h> int main()
{
char Str1[20]="Satrni
nusxalash!"; char
Str2[20];
char* kursatgich=Str1; kursatgich+=7;
strcpy(Str2, kursatgich); cout<<Str2<<endl;
return 0;
}
```

strncpy() funksiyasining strcpy() funksiyasidan farqli joyi shundaki, unda bir satrdan ikkinchisiga nusxalanadigan belgilar soni ko'rsatiladi. Uning sintaksisi quyidagi ko'rinishga ega:

```
char* strncpy(char* str1, const char* str2, size_t num)
```

Agar str1 satr uzunligi str2 satr uzunligidan kichik bo'lsa, ortiqcha belgilar "kesib" tashlanadi. strncpy() funksiyasi ishlatilishiga misol ko'raylik:

```
#include <iostream.h> #include <string.h> int
main()
{
```



```
char Uzun_str[]="01234567890123456789";
char Qisqa_str[]="ABCDEF"; strncpy(Qisqa_str,Uzun_str,4);
cout <<"Uzun_str=" <<Uzun_str<<endl;
cout<<"Qisqa_str="<<Qisqa_str<<endl; return 0;
}
```

Programmada Uzun_str satri boshidan 4 belgi Qisqa_str satriga oldingi qiymatlar ustiga nusxalanadi va natijada ekranga

```
01234567890123456789
```

```
0123EF
```

xabarlari chop etiladi.

strdup() funksiyasiga yagona parametr sifatida satr–manbaga ko‘rsatgich uzatiladi. Funktsiya, satrga mos xotiradan joy ajratadi, unga satrni nusxalaydi va yuzaga kelgan satr- nusxa adresini qaytaradi. strdup() funktsiya sintaksisi:

```
char* strdup(const char* source)
```

Quyidagi programma bo‘lagida satr1 satrining nusxasi xotiraning satr2 ko‘rsatgan joyida paydo bo‘ladi:

```
char* satr1="Satr
```

```
nusxasini olish."; char*
```

```
satr2;
```

```
satr2=strdup(satr1);
```

Satrlarni ulash

Satrlarni ulash (konkatenatsiya) amali yangi satrlarni hosil qilishda keng qo‘llaniladi. Bu maqsadda string.h kutubxonasida strcat() va strncat() funksiyalari aniqlangan.

strcat() funktsiyasi sintaksisi quyidagi ko‘rinishga ega:

```
char* strcat(char* str1, const char* str2)
```

Funktsiya ishlashi natijasida str2 ko‘rsatayotgan satr, funktsiya qaytaruvchi satr – str1 ko‘rsatayotgan satr oxiriga ulanadi. Funktsiyani chaqirishdan oldin str1 satr uzunligi, unga str2 satr ulanishi uchun yetarli bo‘lishi hisobga olingan bo‘lishi kerak.

Quyida keltirilgan amallar ketma-ketligi bajarilishi natijasida satr satriga qo‘shimcha satr ostilari ulanishi ko‘rsatilgan:

```
char satr[80];
```

```
strcpy(satr,"B
```

```
u satrga ");
```

```
strcat(satr,"satr osti ulandi.");
```

Amallar ketma-ketligini bajarilishi natijasida satr satri “Bu satrga satr osti ulandi.” qiymatiga ega bo‘ladi.

strncat() funktsiyasi strcat() funktsiyadan farqli ravishda str1 satrga str2 satrning ko‘rsatilgan uzunligidagi satr ostini ulaydi. Ulanadigan satr osti uzunligi funktsiyaning uchinchi parametri sifatida beriladi. Funktsiya sintaksisi

```
char* strncat(char* str1, const char* str2, size_t num)
```

Pastda keltirilgan programma bo‘lagida str1 satrga str2 satrning boshlang‘ich 10 ta belgidan iborat satr ostini ulaydi:

```
char satr1[80]="Programmalash tillariga
```

```
misol bu-"; char satr2[80]="C++,Pascal,
```

```
Basic"; strncpy(satr1,satr2,10);
```

```
cout<<satr1;
```

Amallar bajarilishi natijasida ekranga “Programmalash tillariga misol bu-C++,Pascal” satri chop etiladi.

Satrlarni solishtirish

Satrlarni solishtirish ularning mos o‘rindagi belgilarini solishtirish (katta yoki qichikligi) bilan aniqlanadi. Buning uchun string.h kutubxonasida standart funktsiyalar

mavjud.

`strcmp()` funktsiyasi sintaksisi

`int strcmp(const char* str1, const char* str2)`

ko'rinishiga ega bo'lib, funktsiya `str1` va `str2` solishtirish natijasi sifatida son qiymatlarni qaytaradi va ular quyidagicha izohlanadi:

➤ `<0` – agar `str1` satri `str2` satridan kichik bo'lsa;

➤ `=0` – agar `str1` satri `str2` satriga teng bo'lsa;

➤ `>0` – agar `str1` satri `str2` satridan katta bo'lsa.

Funktsiya harflarning bosh va kichikligini farqlaydi. Buni misolda

ko'rinishimiz mumkin: `char satr1[80]="Programmash tillariga bu-C++,pascal, Basic.";`

`char satr2[80]="Programmash tillariga bu-`

`C++,Pascal, Basic.";` `int i;`

`i= strcmp(satr1,satr2);`

Natijada `i` o'zgaruvchisi musbat qiymat qabul qiladi, chunki solishtirilayotgan satrlardagi "pascal" va "Pascal" satr ostilarida birinchi harflar farq qiladi. Keltirilgan misolda `i` qiymati 32 bo'ladi – farqlanuvchi harflar satrning 32 elementi hisoblanadi. Agar funktsiyaga

`i= strcmp(satr2,satr1);`

ko'rinishida murojaat qilinsa `i` qiymati `-32` bo'ladi.

Agar satrlardagi bosh yoki kichik harflarni farqlamasdan solishtirish amalini bajarish zarur bo'lsa, buning uchun `stricmp()` funktsiyasidan foydalanish mumkin. Yuqorida keltirilgan misoldagi satrlar uchun

`i=stricmp(satr2,satr1);`

amali bajarilganda `i` qiymati

`0` bo'ladi. `strncmp()`

funktsiyasi sintaksisi

`int strncmp(const char* str1, const char* str2, size_t num)`

ko'rinishida bo'lib, `str1` `str2` satrlarni boshlang'ich `num` sonidagi belgilarini solishtiradi. Funktsiya harflar registrini inobatga oladi. Yuqorida misolda aniqlangan `satr1` va `satr2` satrlar uchun

`i=strncmp(satr1,satr2,31);`

amali bajarilishida `i` qiymati `0` bo'ladi, chunki satrlar boshidagi 31 belgilar bir xil.

`strncmp()` funktsiyasi `strncmp()` funktsiyasidek amal qiladi, farqli tomoni shundaki, solishtirishda harflarning registrini hisobga olinmaydi. Xuddi shu satrlar uchun

`i=strncmp(satr1,satr2,32);`

amali bajarilishi natijasida `i` o'zgaruvchi qiymati `0` bo'ladi.

Satrdagi harflar registrini almashtirish

Berilgan satrdagi kichik harflarni bosh harflarga yoki teskari almashtirishga mos ravishda

`_strupr()` va `_strlwr()` funktsiyalar yordamida amalga oshirish mumkin.

Kompilyatorlarning ayrim variantlarida funktsiyalar nomidagi tagchiziqlik ('_')

bo'lmasligi mumkin.

`_strlwr()` funktsiyasi

sintaksisi `char*`

`_strlwr(char* str)`

ko'rinishida bo'lib, argument sifatida berilgan satrdagi bosh harflarni kichik harflarga almashtiradi va hosil bo'lgan satr adresini funktsiya natijasida qaytaradi. Quyidagi

programma bo'lagi `_strlwr()`

) funktsiyasidan foydalanishga misol

bo'ladi. `char str[]="10 TA KATTA`

`HARFLAR";`

```

    _strlwr(st
r);
cout<<str;

```

Natijada ekranga “10 ta katta harflar” satri chop etiladi.

_strupr() funksiyasi xuddi _strlwr() funksiyasidek amal qiladi, lekin

satrd

agi kichik harflarni bosh harflarga almashtiradi:

```
char str[]="10 ta katta harflar";
```

```
_strupr(str); cout<<str;
```

Natijada ekranga "10 TA KATTA HARFLAR" satri chop etiladi.

Programmalash amaliyotida belgilarni qaysidir oraliqqa tegishli ekanligini bilish zarur bo‘ladi. Buni ctype.h sarlavha faylida e‘lon qilingan funktsiyalar yordamida bilsa bo‘ladi. Quyida ularning bir qnomining tavsifi keltirilgan:

isalnum() – belgi raqam yoki harf (true) yoki yo‘qligini (false)

aniqlaydi; isalpha() – belgini harf (true) yoki yo‘qligini (false)

aniqlaydi;

isascii() – belgini kodi 0..127 oralig‘ida (true) yoki yo‘qligini (false) aniqlaydi;

isdigit() – belgini raqamlar diapazoniga tegishli (true) yoki yo‘qligini (false)

aniqlaydi. Bu funktsiyalardan foydalanishga misol keltiramiz.

```
#include <iostream.h> #include <ctype.h> #include <string.h>
```

```
int main()
```

```
{
```

```
char satr[5]; do
```

```
{
```

```
cout<<"Tug'ilgan yilingizni kiriting,
```

```
marhamat..."; cin.getline(satr,5);
```

```
if(isalpha(satr[0]))
```

```
{
```

```
cout<<"Siz harf kiritdingiz !"; continue;
```

```
}
```

```
if(iscntrl(satr[0]))
```

```
{
```

```
cout<<"Siz boshqaruv belgilarini
```

```
kiritdingiz !"; continue;
```

```
}
```

```
if(ispunct(satr[0]))
```

```
{
```

```
cout<<"Siz punctuatsiya belgilarini
```

```
kiritdingiz !"; continue;
```

```
}
```

```
for (int i=0; i<=strlen(satr); i++)
```

```
{
```

```
if (!isdigit(satr[i]))
```

```
continue; else
```

```
{
```

```
cout << "Sizni tug'ilgan
```

```
yilingiz: "<<satr; return 0;
```

```
}
```

```
}
```

```
}
```

```
while (1);
```

```
}
```

Programada foydalanuvchiga tug'ilgan yilini kiritish taklif etiladi. Kiritilgan sana satr o'zgaruvchisiga o'qiladi va agar satrning birinchi (satr[0]) belgisi harf yoki boshqaruv belgisi yoki punktuatsiya belgisi bo'lsa, shu haqda xabar beriladi va tug'ilgan yilni qayta kiritish taklif etiladi. Programma tug'ilgan yil (to'rtta raqam) to'g'ri kiritilganda "Sizni tug'ilgan yilingiz: XXXX" satrini chop qilish bilan o'z ishini tugatadi.

Satrnı teskari tartıblash

Satrnı teskari tartıblashni uchun strrev() funksiyasidan foydalanish mumkin. Bu funksiya quyidagicha prototipga ega:

```
char* strrev(char* str)
```

Satr reversini hosil

etishga misol: char

```
str[]="telefon";
```

```
cout <<strrev(str);
```

amallar bajarilishi natijasida ekranga "nofelet" satri chop etiladi.

Satrdagi belgini izlash funksiyalari

Satrlar bilan ishlashda satrdagi birorta belgini yoki satr ostini izlash masalasi nisbatan ko'p uchraydi. Bu turdagi masalalar uchun string.h kutubxonasida bir qator standart funksiyalar mavjud.

Satrdagi belgi bor yoki yo'qligini aniqlab beruvchi strchr() funksiyasining prototipi char*

```
strchr(const char* string, int c)
```

ko'rinishida bo'lib, u s belgining satr string satrida izlaydi. Agar izlash muvofaqiyatli bo'lsa, funksiya shu belgining satrdagi o'rnini (adresini) funksiya natijasi sifatida qaytaradi, aks holda, ya'ni belgi satrdagi uchramasa funksiya NULL qiymatini qaytaradi. Belgini izlash satr boshidan boshlanadi.

Quyida keltirilgan programma bo'lagi belgini satrdan izlash

```
bilan bog'liq. char satr[]="0123456789";
```

```
char* pSatr; pSatr=strchr(satr,'6');
```

Programma ishlashi natijasida pSatr ko'rsatgichi satr satrining '6' belgisi joylashgan o'rnini adresini ko'rsatadi.

strchr() funksiyasi berilgan belgini (s) berilgan satr (string) oxiridan boshlab izlaydi. Agar izlash muvofaqiyatli bo'lsa, belgini satrga oxirgi kirishining o'rnini qaytaradi, aks holda NULL.

Misol uchun

```
char satr[]="0123456789101112";
```

```
char* pSatr;
```

```
pSatr=strchr
```

```
(satr,'0');
```

amallarini bajarilishida pSatr ko'rsatgichi satr satrining '01112' satr ostining boshlanishiga ko'rsatadi.

strspn() funksiyasi ikkita satrni mos o'rindagi belgilarni solishtiradi va birinchi ustma-ust tushmagan o'rnini aniqlab beradi (registrni hisobga olgan holda). Funksiya quyidagi ko'rinishdagi prototipga ega:

```
size_t strspn(const char* string, const char* group)
```

Funksiya qaytaruvchi qiymatiga boshqacha mazmun berish mumkin – funksiya ikkita satrdagi ustma-ust tushadigan elementlar sonini beradi:

```
char satr1[]="0123456789101112"; char
```

```
satr2[]="0123456789012345678";
```

```
int mos_belgilar; mos_belgilar=strspn(satr1,satr2);
```

```
cout<<"Satrlardagi mos tushgan belgilar soni= "<<mos_belgilar;
```

amallar bajarilishi natijasida ekranga "Satrlardagi mos tushgan belgilar soni= 10" satri chop etiladi.

strspn() funksiyasi prototipi

size_t strcspn(const char* str1, const char* str2)
ko‘rinishida bo‘lib, u str1 va str2 satrlarni solishtiradi va str1satrining str2 satriga kirmaydigan satr ostining uzunligini beradi. Boshqacha aytganda, funktsiya satrlarning birinchi kesishish o‘rnini qaytaradi. Masalan

```
char  
satr[]="Birinci  
satr"; int index;  
index=strcspn(satr,"sanoq tizimi");
```

amallar bajarilgandan keyin index o‘zgaruvchisi 9 qiymatini qabul qiladi, chunki 9 joydagi belgi ikkinchi satrning birinchi belgisi bilan mos tushadi.

strpbrk() funktsiyasi prototipi
char* strpbrk(const char* str1, const char* str2)
ko‘rinishga ega bo‘lib, u str1 satrdagi str2 satrga kiruvchi birorta belgini izlaydi va agar bunday element topilsa, uning adresi funktsiya qiymati sifatida qaytariladi, aks holda funktsiya NULL qiymati qaytaradi. Quyidagi misol funktsiyani qanday ishlashini ko‘rsatadi.

```
char satr1[]="0123456789ABCDEF";  
char  
satr2[]="ZXYabcdefABC";  
char* element;  
element =  
strpbrk(satr1,satr2);  
cout<<element<<'\n';
```

Programma ishlashi natijasida ekranga str1 satrining “ABCDEF” satr ostisi chop etiladi.

Satr ostilarini izlash funktsiyalari

Satrlar bilan ishlashda bir satrda ikkinchi bir satr ostining to‘liq kirishini aniqlash bilan bog‘liq masalalar uchraydi. Masalan, matn tahrirlaridagi satrdagi birorta satr ostini ikkinchi satr osti bilan almashtirish masalasi. Standart string.h kutubxonasi bu toifadagi masalalar uchun bir nechta funktsiyalarni taklif etadi.

strstr() funktsiyasi quyidagicha
e‘lon qilinadi: char* strstr(const
char* str, const char* substr)

Bu funktsiya str satrini substr satr ostisi kirishi bo‘yicha qarab chiqadi, agar substr satr ostisi str satriga to‘liq kirishi mavjud bo‘lsa, birinchi kirishdagi birinchi belgining adresi javob tariqasida qaytariladi, aks holda funktsiya NULL qiymatini qaytaradi.

Quyidagi misol strstr() funktsiyasini ishlatishni ko‘rsatadi.

```
char satr1[]="Satrdan satr ostisi izlanmoqda, satr ostisi  
mavjud"; char satr2[]="satr ostisi";  
char* satr_osti;  
satr_osti = strstr(satr1,satr2);  
cout<<satr_osti<<'\n';
```

Programma buyruqlari bajarilishi
natijasida ekranga "satr ostisi
izlanmoqda, satr ostisi mavjud"
satri chop etiladi.

Laboratoriya ishlarnalari

1-Laboratoriya mashg'uloti.

Mavzu: Borland C++ Builder dasturlash muhitiga kirish, ishchi muhit, oynalar.

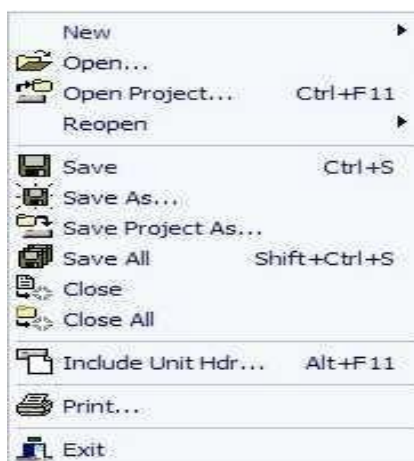
Ishning maqsadi: Talabalarda Borland C++ Builder dasturlash muhiti haqida boshlang'ich tushunchalarni hosil qilish.

Nazariya bo'yicha qisqacha ma'lumot

Borland C++ Builder 6 muhiti asosiy menyusu

buyruqlari File buyruqlari guruhi

Menyuning birinchi File (Fayl) buyrug'i guruhi 20.7-rasmda berilgan

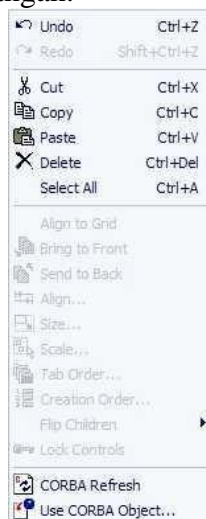


20.7-rasm. File buyrug'i guruhi

Bu menyusu buyruqlari fayllar bilan ishlashga mo'ljallangan va qo'yidagi amallarni bajaradi: **(New)** yangi fayl hosil qilish, forma (oyna) ochish, mavjud fayllarni **(Open)** ochish, fayllarni saqlash **(Save)** va **(Close)** yopish, ilova dastur kodi matnini **(Print)** chop qilish mumkin.

Edit buyruqlari guruhi

Edit (Muharrir), 20.8-rasmda ifodalangan.



20.8-rasm. Edit buyruqlari guruhi

Menyuning bu buyruqlari guruhi tahrir qilish uchun zarur bo'lgan buyruqlardir. Masalan inkor qilish (**Undo**) va (**Redo**) takrorlash, qirqib olish(**Cut**), nusxa olish (**Copy**), nusxa olinganni qo'yish (**Paste**) va o'chirish (**Delete**) kabi amallarni o'z ichiga olgan.

Search buyruqlari guruhi

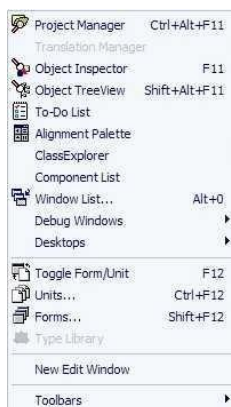
Search (Izlash)



20.9-rasm. Search (Izlash)

Menyuning bu buyruqlari guruhi fayllardan biror matnni izlash va avtomatik almashtirish, kerakli satrga o'tish va uni o'zgartirish imkonini beradi.

View buyruqlari guruhi.



20.10-rasm. View(Ko'rish) buyruqlar guruhi

Menyuning bu buyruqlari guruhidan ilova va komponentalarni boshqarishni asosiy muloqot oynalari chaqiriladi. Masalan, ilova menedjeri (**Project Manager**), komponentlar ro'yxati (**Component List**) va oynalar ro'yxati (**Window List**).

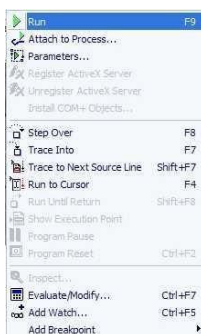
Project buyruqlari guruhi

Project (Ilova) 20.11-rasmda keltirilgan.

Menyuning bu buyruqlari guruhida ilovani boshqarish buyruqlari yig'ilgan. Bu buyruqlar yordamida fayllarni qo'shish, o'chirish, VCL komponentalar kutubxonasiga komponenta qo'shish, kompilyatsiya qilish va shunga o'xshash amallarni bajarish mumkin.

Run buyruqlari guruhi

Run (Bajarish) 20.12-rasmda ifodalangan.

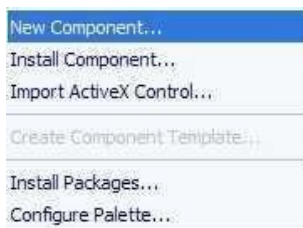


20.12-rasm. Run buyruqlar guruhi

Menyuning bu buyruqlari guruhi yordamida ilovani ishga tushirish va bekor qilish, ilovani butunlay va qadamba-qadam rejimda ishga tushirish, ko'rish uchun qo'shimcha o'zgaruvchilar kiritish, ilova bajarilishini to'xtatuvchi belgilar kiritish mumkin.

Component buyruqlari guruhi

Component (Komponent) buyruqlar guruhi 20.13-rasmda berilgan.



20.13-rasm. Component (Komponent) buyruqlar guruhi

Menyuning bu buyruqlari guruhi yordamida yangi komponentalarni qo'shish va konfiguratsiyalarini aniqlash ishlari bajariladi.

Database buyruqlari guruhi

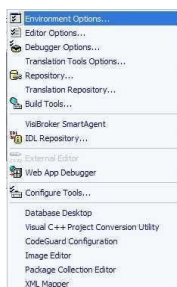
Database (Ma'lumotlar bazasi) buyruqlar guruhi (20.14-rasm) ma'lumotlar bazasi bilan ishlaydigan buyruqlarni o'z ichiga olgan.



20.14-rasm. Database (Ma'lumotlar bazasi) buyruqlar guruhi

Tools buyruqlari guruhi

Tools (Instrumentlar) buyruqlar guruhi 20.15-rasmda berilgan

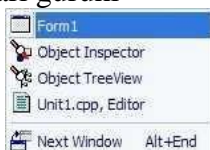


20.15-rasm. Tools buyruqlar guruhi

Menyuning bu buyruqlari guruhi yordamida ilova parametrlarini o'rnatish va yordamchi dasturlar buyruqlarini chaqirish mumkin.

Windows buyruqlari guruhi

Windows (oyna) buyruqlari guruhi

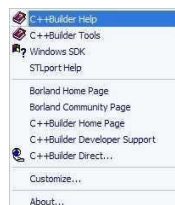


20.16-rasm. Windows(oyna) buyruqlari guruhi

Menyuning bu buyruqlari guruhi yordamida interfeysning oynalarini boshqarish mumkin.

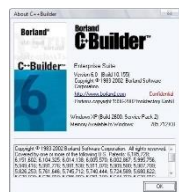
Help buyruqlari guruhi

Help (Yordam) buyruqlari guruhi



20.17-rasm. Help (Yordam) buyruqlari guruhi

Menyuning bu buyruqlari guruhi yordamida muhit, til, komponentalar va kompyuter haqidagi ma'lumotlarni olishimiz mumkin. Masalan, quyidagicha:



20.18-rasm. About buyrug'i oynasi

Tezkor tugmalar

Tezkor tugmalar asosiy menyu bilan yonma-yon joylashgan.



20.19-rasm. Tezkor tugmalar

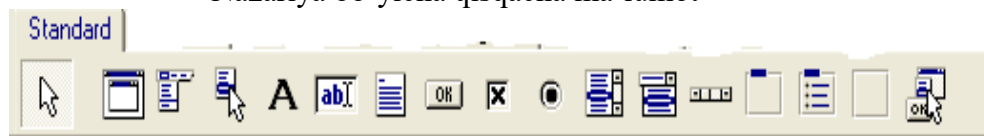
Bu tugmalar ko'p ishlatiladigan, tez-tez murojaat qilinadigan bo'lganligi uchun alohida guruhlariga ajratilgan holda kichik piktogrammalar yordamida berilgan. Bu tugmalar asosiy menyuda ham mavjud, ammo vaqtdan yutish uchun ajratib ham qo'yilgan.

2-Laboratoriya mashg'uloti.

Mavzu: Borland C++ Builderda Label, Edit, Button va Memo komponentlaridan foydalanib dastur tuzish.

Ishning maqsadi: Talabalarga Borland C++ Builderda Label, Edit, Button va Memo komponentlaridan foydalanib dastur tuzishni o'rgatish.

Nazariya bo'yicha qisqacha ma'lumot



Standard komponentalar

Komponentalar palitrasining Standard qo'shimcha ilovalari komponentalari sizning dasturingizga Windows standart interfeysli elementlarning 14 tasining ulanishini amalga oshiradi.

TMainMenu



Bosh menyu komandalari Panelini va ularga mos keladigan tushib qoladigan menyularni yaratadi. Barcha menyu komandalarining identifikatorlari menyuning har qanday konkret komandasiga kirish huquqiga ega bo'lgan Items xususiyati bilan aniqlanadi. AutoMerge xususiyati Merge va UnMerge metodlari bilan birgalikda turli shakldagi menyularning birlashish jarayonini boshqaradi.

TPopupMenu



Shakl yoki bironta boshqa komponenta uchun maxsus menyu yaratadi. E'tiborga oling, aynan shu maqsad uchun har qanday boshqa komponenta PopUpMenu xususiyatiga ega bo'lib, bu xususiyatda siz uning bilan bog'liq menyuga iqtibos qilishingiz mumkin.

Agar siz sichqonchaning o'ng tugmasini shaklga yoki berilgan komponenta mansub bo'lgan biron boshqa elementga bosish bilan maxsus menyu ekranda paydo bo'lishini xoxlasangiz, AutoPopUp xususiyatining true qiymatini o'rnatib. Voqea qayta ishlatgichi - OnPopUp yordamida bevosita maxsus menyuning paydo bo'lishi oldidan bajariladigan protsedurani aniqlashi 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'rnatib. Kalta soha ichida matnning hammasini ko'rish imkoniga ega bo'lish uchun, WordWrap xususiyatining true qiymatini bering. Transparent xususiyatining true qiymatini o'rnatib, boshqa komponentaning bir qismini to'g'ri uning ustida joylashtirilgan nom orasidan ko'rinib turadigan qilishingiz mumkin.

TEdit



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 hatti- 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 bekor qilish tugmachasi, Dialog darchasida har gal Escape klavishasi bosilganda, OnClick voqea qayta ishlatgichini ishga tushiradi.

TVutton komponentasi TButtonControl sinfining hosilasi hisoblanadi.

3-Laboratoriya mashg'uloti.

Mavzu: Borland C++ Builderda CheckBox, RadioGroup, ComboBox va ListBox komponentlaridan foydalanib dastur tuzish.

Ishning maqsadi: Talabalarga Borland C++ Builderda CheckBox, RadioGroup, ComboBox va ListBox komponentlaridan foydalanib dastur tuzishni o'rgatish.

Nazariya bo'yicha qisqacha ma'lumot

TCheckBox



Ikkita holatga hamda tavsifiy matnga ega bo'lgan kvadrat check-boxni yaratadi (bunda tavsifiy matn check-boxning vazifasini spetsifikatsiya qiladi).

Box holatini bildiruvchi «Check» biron-bir variantning tanlanishiga mos keladi (box ustidan tortilgan chiziq bilan belgilanadi). «UnCheck» 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. AllowGraed xususiyatining true qiymatini o'rnatib, boxni to'qroq rangli (masalan, kulrang) qilish mumkin. State xususiyati joriy holatni va box 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. Radio-tugma bosilganda, Checked komponentasining xususiyati ham mos ravishda o'zgaradi va OnClick voqeasi yuzaga keladi.

Odatda radio-tugmalar avvaldan shaklda o'rnatilgan konteyner ichiga joylashtiriladi. Agar bitta tugma tanlangan bo'lsa, ushbu guruhga mansub barcha boshqa tugmalarning tanlovlari avtomatik tarzda olib tashlanadi. Masalan, shakldagi ikkita radio-tugma, agar ular boshqa-boshqa konteynerlarda joylashgan bo'lsagina bir paytning o'zida tanlab olinishi mumkin. Agar radio-tugmalarning guruhlanishi ochiq-oydin berilmagan bo'lsa, bu holda ularning hammasi, yashirin holda, konteyner darchalari (TForm, TGroupBox yoki TPanel) dan birida guruhlanadi.

TRadioButton komponentasi TButtonControl sinfining hosilasidir.

TListBox



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

Agar ro'yxatdagi barcha elementlar ajratilgan sohaga sig'masa, ro'yxatni aylantirish lineykasi yordamida ko'rib chiqish mumkin. Ro'yxat elementlari Items xususiyatining ichida, dastur bajarilish vaqtida tanlab olinadigan element raqami esa ItemIndex xususiyatining ichida joylashgan bo'ladi. Ro'yxat elementlari matn muharririning darchasi Items xususiyati qiymatining grafasida tugmacha bilan ochiladi. Ro'yxat elementlarini Items obyektining Add, Append, Delete va Insert metodlari yordamida dinamik tarzda qo'shish, o'chirish, orasiga joylash va o'rnini almashtirish mumkin. Masalan:

ListBox1->Items->Add («Ro'yxatning oxirgi elementi»);

Sorted xususiyatining true qiymati ro'yxat elementlarini alifbo tartibida ajratib joylashtiradi.

TListBox komponentasi TCustomListBox sinfining hosilasi bo'lib, uning barcha

xususiyat, metod va voqealariga vorislik qiladi.

TComboBox



Tahrir sohasi hamda matn variantlarining tushib qoladigan ro'yxati kombinatsiyasini tanlash uchun ishlatiladi.

Text xususiyatining qiymati bevosita tahrir sohasiga kiritib qo'yiladi. Foydalanuvchi tanlab olishi mumkin bo'lgan ro'yxat elementlari Items xususiyatining ichida bo'ladi. Dasturning bajarilish paytida tanlab olinishi mumkin bo'lgan element raqami ItemIndex xususiyatining ichida bo'ladi. 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.

Items obyektining Add, Append, Delete va Insert metodlari yordamida ro'yxat elementlarini dinamik tarzda qo'shish, o'chirish orasiga qo'yish va o'rnini almashtirish mumkin. Masalan: ComboBox->Items->Insert(0, «Ro'yxatdagi birinchi element»);

Sorted xususiyatining true elementi ro'yxat elementlarini alifbo tartibida navlarga ajratilishini ta'minlaydi. TComboBox komponentasining turini Style xususiyatidan tanlab olish mumkin.

TComboBox komponentasi TCustomComboBox sinfining hosilasi bo'lib uning barcha xususiyatlari, metodlari va voqealariga vorislik qiladi.

TScrollBar



Darcha, shakl yoki boshqa komponenta ichidagilarini ko'rib chiqish uchun ishlatiladi. Masalan, biron-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 SmallShange 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.

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 Windows ning bir nomdagi obyektning inkapsulyatsiyalanishidan iborat.

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. RadioGroup komponentasiga tugmalarni qo'shish uchun, Items xususiyatining tahriri bajarilishi kerak. Items xususiyatining navbatdagi satriga nom berilsa, shu tugma guruhlovchi ramkada paydo bo'ladi. Ushbu daqiqada qaysi tugma tanlab olinishi kerakligini ItemIndex xususiyatining qiymati aniqlab beradi. Columns xususiyatining tegishli qiymatini

joylashtirib, siz radiotugmalarni bir necha ustunga guruhlashingiz mumkin.

TPanel



Boshqa komponentlarni 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 komponentasi TCustomPanel sinfining hosilasi bo'lib, uning barcha xususiyatlar, metodlari va voqealari to'liq vorislik qiladi.

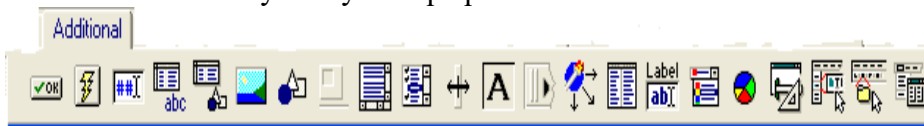
4-Laboratoriya mashg'uloti.

Mavzu: Borland C++ Builderda Additional komponentlaridan foydalanib dastur

Ishning maqsadi: Talabalarga Borland C++ Builderda Additional komponentlaridan

foydalanib dastur tuzishni o'rgatish.

Nazariya bo'yicha qisqacha ma'lumot



20.38-rasm. Additional komponentalari

Komponentalar palitrasining Additional qo'shimcha ilovalar komponentalari sizning dasturingizga Borland korporatsiyasi maxsus C++ Builder muhiti uchun ishlab chiqqan 9 ta boshqarish elementini kiritadi.

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 uchun xususiyatlarga ega bo'ladi. Siz C++ Builder qurilmasiga kirgan alohida tasvirlar katalogidagi grafik tugmalarning tayyor stillaridan foydalanishingiz ham, yoki bo'lmasa, tasvirlarni tahrir qilish tizimlarining biri tomonidan yaratilgan suratlardan foydalanishingiz ham mumkin.

Tugmaning turli holatlariga (masalan «bosilgan», «qo'yib yuborilgan», «ta'qiqlangan» va h.k.) turli bit obrazlari mos kelishi mumkin.

Tasvirlar Fayllari Muharririning bmp kengaytmali darchasi (20.39-rasm) Glyph xususiyatining qiymatlari tugmasi bilan ochiladi. King xususiyati sizga yozuvlar va tegishli grafika (OK, Cancel, Hello va boshqalar) bilan ta'minlangan standartlashtirilgan tugmalarni yaratish imkonini beradi.



TSpeedButton



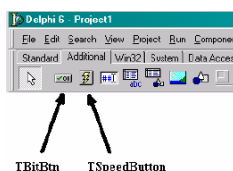
Odatda ma'lum menyu komandalarini tez chaqirish yoki rejimlarni o'rnatish Paneli (TPanel) da joylashtiriladigan grafik tugmani yaratadi.

Tezkor tugmaning turli holatlariga (masalan «bosilgan», «qo'yib yuborilgan»,

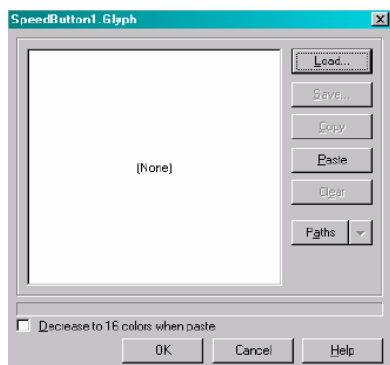
«ta'qiqlangan» va h.k.) turli bit obrazlari mos kelishi mumkin. Bir-birining o'rnini bosadigan tasvirlar va yozuv matnini tanlash uchun xususiyatlar mavjud. Kengayishli tasvirlar fayllari muharririning darchasi Glyph xususiyati qiymtlarining grafasidagi tugma bilan ochiladi. Tez tugmalarning boshqa xususiyatlari ularning biron-bir guruhdagi ishini tashkil etadi.

TSpeedButton va TBitBtn tugmalari.

Bu tugmalar **TButton** vazifalarini bajaradi. Yagona farqi matndan tashqari racmlarni ham aks ettiradi. **TSpeedButton** tugmasi fokus olmaydi. Bu shuni bildiradiki, agar matn qatorida satr terib, bu tugma bosilsa, shu hodisa qayta ishlangandan so'ng fokus yana matn qatoriga qaytib keladi. TAB tugmasi bilan bu tugmani ajratib bo'lmaydi.



Tugmaga rasm o'rnatish uchun ikki marta *Glyph* xossasi qatorida chertish lozim. Natijada rasm paydo bo'lgan yuklash oynasida *Load* tugmasini bosish lozim. Ko'p rasmlar Program Files\Common Files\Borland Shared\Images\Buttons katalogida joylashgandir.



TBitBtn va **TSpeedButton** tugmalari deyarli bir xil xossalarga egadir. Ular uchun umumiy *Layout*, xossasi rasm va matn o'zaro joylashuvini o'zgartirishga imkon beradi. Quydagi rasmda har xil qiymatlar mos variantlari ko'rsatilgan

TBitBtn tugmasining yana bir xossasi *Kind* bo'lib oldindan tayyorlangan standart tugmalarni tanlash imkonini beradi. Quyidagi rasmda standart tugmalar va ularga mos qiymatlarni ko'rish mumkin.

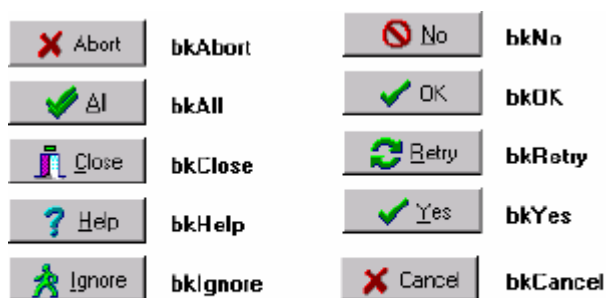


Рис 11.1.7 Различные значения свойства Kind.

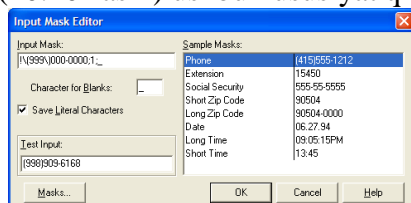
Yana bir xossa ModalResult – dialog oynasi uchun tugma qaytaradigan natijani tanlashga imkon beradi.

TSpeedButton tugmasining *GroupIndex* xossasi tugmalarni guruhlashga imkon beradi. Buning uchun bir guruhga tegishli tugmalarning *GroupIndex* xossasi bir xil qiymatga masalan 1 ga teng bo'lishi kerak. Guruhlangan tugmalarning biri bosilsa, qolganlaridan ajralib qoladi. Buning uchun *Down* xossasi qiymati *true* ga teng bo'lishi kerak.

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 (20.40-rasm) ushbu xususiyat qiymatlari grafasida tugma bilan ochiladi.



20.40-rasm. Telefon raqamlarini kiritish uchun niqobni yaratish

TMaskEdit komponentasi TCustomMaskEdit sinfining to'g'ridan-to'g'ri hosilasidir. U satrlar yoki ustunlar bo'yicha belgili ketma-ketliklarni aks ettirish uchun mo'ljallangan muntazam (regulyar) to'rnini yaratadi.

TStringGrid



Ushbu 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 to'tish imkonini beradi. Setkaning barcha

simvolli ketma-ketliklari setkaning kerakli uyasini adreslaydigan (manzillaydigan) Cells xususiyatida mujassamdir.

TDrawGrid



To'zilma holiga keltirilgan grafik ma'lumotlarni satrlar yoki ustunlar bo'yicha aks ettirish uchun muntazam setka yaratadi. RowCount va ColCount xususiyatlari vertikal bo'yicha va gorizontal bo'yicha setka uylarining 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 xususiyatli tomonidan belgilanadi, aylantirish chiziqlari esa ScrollBars xususiyati tomonidan qo'shiladi. FixedCol va FixedRows xususiyatlari ustunlar va satrlarning aylantirilishini ta'qiqlab qo'yish imkonini beradi, Fixed Color xususiyati esa barcha usutn va satrlarga ma'lum rang beradi.

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

Dasturning ishlash paytida siz CellRest metodi yordamida biron-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.

Dastur bajarilish paytida qaysi satr setkaning ustki satri bo'lishini aniqlash yoki TopRow xususiyati yordamida ko'rsatilgan satrni ustki holatga qo'yib qo'yish mumkin. Qaysi ustun setkaning ko'rinib turadigan usutni bo'lishini aniqlash uchun, LeftCol xususiyatidan foydalaning. VisibleColCount va VisibleRowCount xususiyatlarining qiymatlari setkaning ko'rinib turgan ustunlari va satrlarining umumiy sonini spetsifikatsiyalaydi.

TImage



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

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

Tasvirlar fayllarining dinamik yuklanishi va saqlanishi uchun, Picture obyekt xususiyatining LoadFromFile va SaveToFile metodlaridan qo'yidagi turlar yordamida foydalaning:

Image->Picture-

>LoadFromFile(«<fayl nomi>»);

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



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

Tanlab olingan geometrik shaklning turini Shape xususiyati, rang va bo'yash usulini Brush

komponentasiga joylangan ikkita Color va Style xususiyatlari aniqlaydi. Shakllarning o'lchamlarini ham tegishli xususiyatlar aniqlaydi.

TBevel



Xuddi uskuna bilan o'yilgandek xajmli ko'rinadigan chiziqlar, box lar yoki ramkalarni yaratadi.

Komponenta chizayotgan obyektни Shape xususiyati aniqlaydi, Style xususiyatining qiymati esa obyekt ko'rinishini o'zgartirib, uni bo'rtiq yoki botiq holga keltiradi. Foydalanuvchi shakl o'lchamlarini o'zgartirganda ham obyektning nisbiy holatini o'zgarms qoldirish uchun, Align xususiyatining true qiymatini o'rnatish.

TScrollBar



Darchada o'lchamlari o'zgaruvchan box ni yaratadi, bu box 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 boksiini 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 biron-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.

5-Laboratoriya mashg'uloti.

Mavzu: Fayllar bilan ishlash. Dialog oynalari. Panel va menyu yaratuvchi komponentlar


Ishning maqsadi: Talabalarga Fayllar bilan ishlash, Dialog oynalari, Panel va menyu yaratuvchi komponentlardan foydalanib dasturlar tuzishni o'rgatish

Nazariya bo'yicha qisqacha ma'lumot

Dialogs sahifasi komponentalari Windowsning odatiy dasturlari muloqot oynalarini hosil qilishda foydalaniladi. Dialogs komponentalari fayllarni ko'rsatish yoki sozlashlarni o'rnatish uchun ishlatiladi. C++ Builderning Dialogs komponentalaridan foydalanish ilovalarni yaratishda vaqtini tejaydi va Windowsning odatdagi muloqot oynalaridan foydalanish imkonini yaratadi.

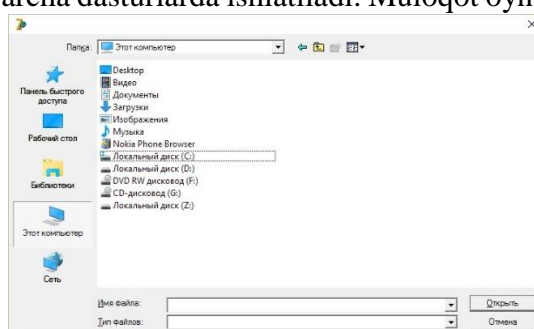
Dialogs



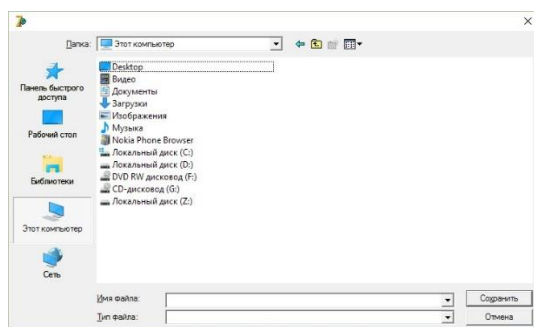
№	Komponenta	Tavsifi
23.	 OpenDialog	«Открыть файл» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.

24.	 SaveDialog	«Сохранить файл как...» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
25.	 OpenPictureDialog	«Открыть рисунок» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
26.	 SavePictureDialog	«Сохранить рисунок как...» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
27.	 FontDialog	«Шрифты» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
28.	 ColorDialog	«Цвет» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
29.	 PrintDialog	«Печать» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
30.	 PrinterSetupDialog	«Установка принтера» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
31.	 FindDialog	«Найти» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
32.	 ReplaceDialog	«Заменить» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.
33.	 PageSetupDialog	«Параметры страницы» muloqot oynasini yaratish uchun mo'ljallangan. Komponenta vizual ko'rinishga ega emas.

OpenDialog – faylni ochish muloqot oynasi va **SaveDialog** faylni saqlash muloqot oynasi komponentalari deyarli barcha dasturlarda ishlatiladi. Muloqot oynalari quyidagi ko'rinishda bo'ladi.

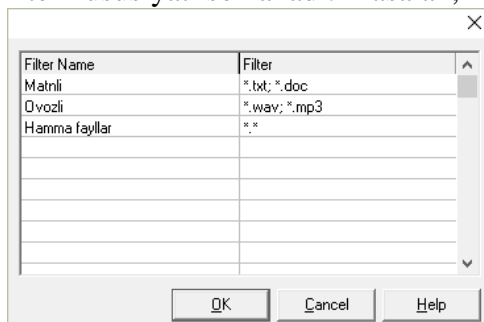


Faylni ochish muloqot oynasi.



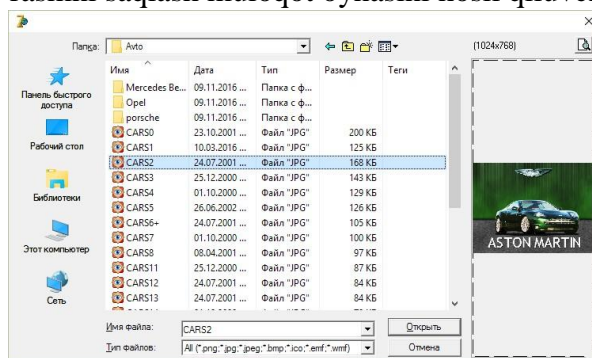
Faylni saqlash muloqot oynasi.

Ushbu komponentlarning barcha xususiyatlari bir xil, yagona farq ularning biri faylni ochishda soʻralsa, ikkinchisi faylni yopishda soʻraladi. Bu komponentlarning eng asosiy xususiyati FileName (tanlangan faylning nomi) dir. Xususiyatning qiymatini muloqot oynasi ochilmasdan oldin ham koʻrsatish mumkin. Xususiyatga kiritilgan fayl nomi yuqoridagi muloqot oynalarining Имя файла bandidan paydo boʻladi. Muloqot oynalarining Тип файла boʻlimida fayllarning turini koʻrsatish uchun komponentlarning Filter xususiyati sozlanadi. Masalan,

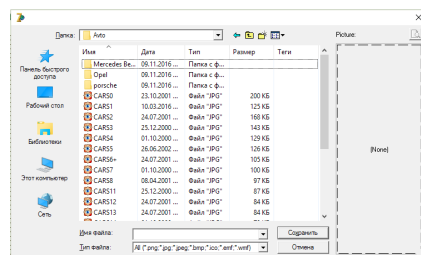


Title xususiyati orqali muloqot oynasiga nom berish mumkin.

C++ Builderning kutubxonasida maxsus **OpenPictureDialog** – rasm ochish muloqot oynasi va **SavePictureDialog** – rasmni saqlash muloqot oynasini hosil qiluvchi komponentlar mavjud.

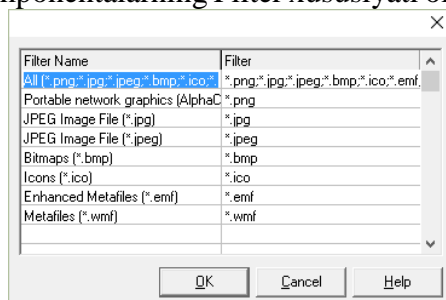


Rasm ochish muloqot oynasi



Rasmni saqlash muloqot oynasi

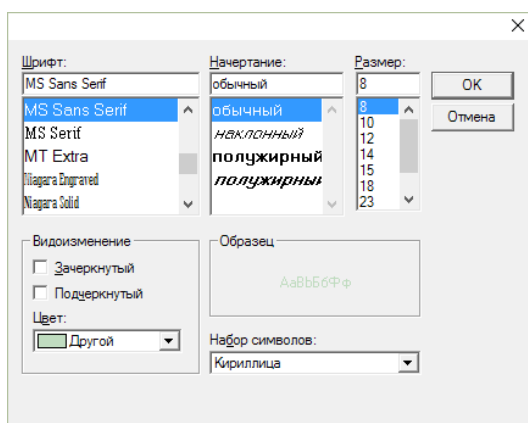
OpenPictureDialog va SavePictureDialog komponentlarning xususiyatlari OpenFileDialog va SaveDialog komponentlarining xususiyatlari bilan bir xil. Yagona farq shundaki OpenPictureDialog va SavePictureDialog komponentlarning Filter xususiyati oldindan sozlangan boʻladi. Yaʼni,



Filter oynasi

Ushbu feltirda barcha grafik fayl turlari kiritilgan. Foydalanuvchi hohlasa ularni oʻchirishi yoki qoʻshimcha kiritishi mumkin.

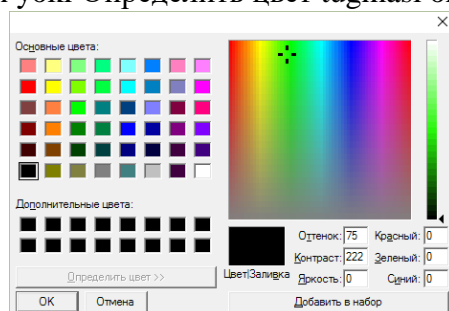
FontDialog komponentasi shrift xususiyatlarini sozlash muloqot oynasini chaqiradi. Ushbu oynada shrift nomi, yozilish shakli, o'lchami va boshqa xususiyatlarni sozlash mumkin.



Шрифт xususiyatini sozlash oynasi

Komponentning asosiy xususiyati – Font. **MaxFontSize** va **MinFontSize** xususiyatlari shrift o'lchamining eng katta va eng kichik qiymatini o'rnatadi. Agar xususiyatlarning qiymati 0 (nol) ga teng bo'lsa (по умолчанию), o'lchamni kiritishda hech qanday cheklashlar kuzatilmaydi. Agarda xususiyatlarga qiymat kiritilsa, muloqot oynasining **Размер** bo'limida kiritilgan diapazondagi qiymatlar hosil bo'ladi. Agar diapazondan tashqaridagi qiymat kiritilsa, «Размер должен лежать в интервале ...» ogohlantirish oynasi hosil bo'ladi.

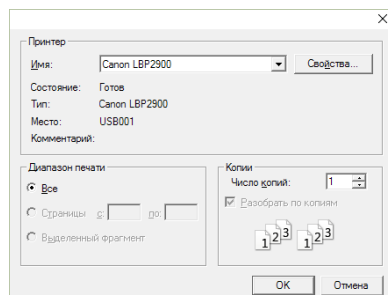
ColorDialog komponentasi rang tanlash muloqot oynasini chaqiradi. Bundan foydalanuvchi bazadagi ranglar palitrasidan yoki Определить цвет tugmasi orqali kerakli rangni tanlashi mumkin.



Rang tanlash oynasi

ColorDialog komponentasining asosiy xususiyati – Color. Bu xususiyat foydalanuvchi oynadan tanlagan rangni bildiradi.

PrintDialog komponentasi chop qilish muloqot oynasini chaqiradi. Foydalanuvchi bunda printerni tanlashi va uning xususiyatini sozlashi, nusxalash sonini ko'rsatishi, sahifani yoki belgilangan fragmentni chop qilishi mumkin.



Chop qilish oynasi

PrintDialog komponentasi chop qilishni amalga oshirmaydi. U faqat chop qilish xususiyatlarini sozlaydi. Chop qilish uchun esa **Printer** komponentasidan foydalaniladi.

Ushbu dasturlar yatlari:

1 **FindText** – foydalanuvchi tomonidan kiritilgan qidiriladigan yoki almashtiriladigan satr. **ReplaceText** – **ReplaceDialog** komponentasiga tegishli. **FindText** ga almashtirilishi kerak boʻlgan satr.

2 **Options** – koʻplab xususiyatlar.

3 **FindDialog** va **ReplaceDialog** komponentalari oʻzlari aqidiruv yoki almashtirishni amalga oshirmaydi. Ular faqat . foydalanuvchi interfeysini sozlaydi. Qidirish yoki almashtirishni amalga oshirish uchun kerakli buyruqlarni kiritish talab qilinadi.

4 Muloqot oynasidagi Найти далее tugmasi ishlashi uchun **OnFind** hodisasiga, Заменить yoki Заменить все tugmalari uchun esa **OnReplace** hodisasiga foydalanuvchi tegishli buyruqlarni kiritishi lozim.

5

6 **Laboratoriya mashgʻuloti.**

7 **Mavzu: Borland C++ Builderda bir oʻlchovli massivlarga doir dastur tuzish.**

8 **ReplaceDialog** komponentasida satrni almashtirish mavjud.

9 Ishning maqsadi: Talabalarga Borland C++ Builderda bir oʻlchovli massivlarga doir dastur tuzishni oʻrgatish

10 Nazariya boʻyicha qisqacha maʼlumot

11 Koʻp hollarda jadval yoki matritsalar koʻrinishidagi maʼlumotlar bilan ish yuritish kerak boʻladi. Jadvalda maʼlumotlar juda koʻp boʻlgani sabab, ularning har bir yacheykasidagi sonni mos ravishda bitta oʻzgaruvchiga qiymat qilib berilsa ular ustida ish bajarish ancha noqulayliklarga olib keladi. Shu sabab dasturlashda bunday muammolar massivlarni ishlatish yordamida hal qilinadi.

12 Massiv - bu bir nom bilan belgilangan qiymatlar guruhi yoki jadvaldir. Massivning har bir elementi massiv nomidan soʻng kvadrat qavs ichiga olingan raqam va arifmetik ifoda yozish bilan belgilanadi. Qavs ichidagi raqam massiv indeksini belgilaydi. Vektorni bir oʻlchovli massiv, matritsani ikki oʻlchovli massiv deb qarash mumkin.

13 Bir oʻlchovli massivda uning har bir elementi oʻzining joylashgan oʻrin nomeri bilan aniqlanadi va nomeri qavs ichida indeks bilan yoziladi. Ikki oʻlchovli massiv elementi oʻzi joylashgan i satr va ustun nomerlari yordamida aniqlanadi. Shu sabab ikki oʻlchovli massiv elementi ikkita indeks orqali yoziladi. Masalan: $A[i,j]$ bu yerda i-satr nomeri j-ustun nomerini bildiradi.

14 Massivni eʼlon qilish dasturning bosh qismida berilib, uning yozilishi umumiy holda quyidagicha boʻladi:

15 **<Massiv nomi>:Array[oʻlcham] of <element turi>;**

16 Masalan: **A,B:array[1..10] of real; C,A1,D1:Array[1..10,1..20] of real;** bu yerda A va B massivlari 10 tadan elementga ega. C,A1,D1 massivlari esa $10 \times 20 = 200$ tadan elementga ega.

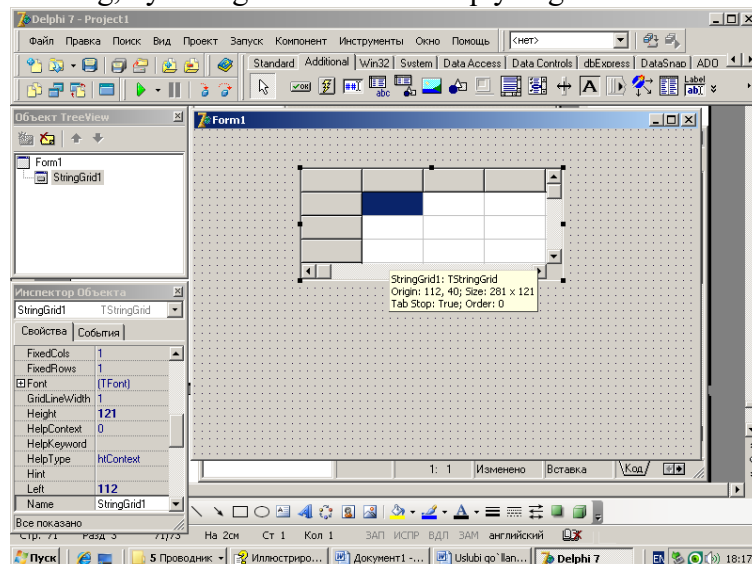
1-bosqich.

1. Komponentalar ro'yxatini Additional bandidan StringGrid1(abc) tugmani tanlaymiz, ushbu tanlash (3.5.2-rasmda) ko'rsatilgan.



3.5.2.-rasm. Additional panilining ko'rinishi

Tugma tanlangandan so'ng, oynaning ko'rinish holati quyidagicha bo'ladi:



3.5.3-rasm. StringGrid1(abc) tugmasidan foydalanish

Hosil bo'lgan jadvaldan 5 ta ustun, 1 ta satr yarating. Bu jadvalni yaratish uchun parametrlar va xossalar bo'limidan **ColCount** hodisalar bandiga 5 raqamini kiritamiz. Satr bo'yicha o'zgartirish kiritish uchun **RowCount** bandiga 1 raqamini kiritamiz.

Umumiy holda oynani ishga tayyorlashimiz uchun parametrlar va xossalar bo'limidan jadvalda ko'rsatilgan buyruqlarning tarkibi quyidagicha:

3.5.1-jadval

Parametrlar va xossalar bo'limi	Hodisalar bo'limi
ColCount	5
FixedCols	0
RowCount	1
DefaultRowHeight	24
Height	24
DefaultColWidth	64
Width	328
Options . goEditing	True
Options . AlwaysShowEditing	True
Options .goTabs	True

Jadvalda ko'rsatilgan 3 ta **Options.goEditing**, **Options. AlwaysShowEditing**, **Options.goTabs**

buyruqlarni hosil qilishimiz uchun parametrlar va xossalar bo'limidan "Options" parametirining

"+" ishorasini "-" ishoraga o'tkazamiz (jadval ustiga sichqoncha tugmasini bir marta bosganimizdan so'ng, aytilgan topshiriqni bajarishimiz mumkin).

7-Laboratoriya mashg'uloti.

Mavzu: Borland C++ Builderda ikki o'lchovli massivlarga doir dastur tuzish.

Ishning maqsadi: Talabalarga Borland C++ Builderda ikki o'lchovli massivlarga doir dastur tuzishni o'rgatish.

Nazariya bo'yicha qisqacha ma'lumot

Massiv elementlari yig'indisini va o'rta arifmetigini hisoblash dasturi

Izoh : Hosil bo'lgan ko'rinishdagi **edit** qatoriga qiymat kiritib kiritish tugmasini bossak, **ListBox** maydonida paydo bo'ladi. Uning qiymatini hisoblash uchun esa, hisoblash tugmasini tanlash kifoya.

2- misol. massiv
tuzamiz. element
larini
 $A(5,5)$ yig'indi
si va
o'rta
arifmeti
gini
topish
dasturi

Ikki o'ldirli massiv elementlari yig'indisi va o'rta arifmetigini hisoblash uchun **button**, **memo** va **StringGrid** tugmalaridan foydalanamiz.

Dastur kodi va oynaning umumiy ko'rinishini keltiramiz:

unit Unit1; interface uses

**Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs, StdCtrls, Grids;**

type

TForm1 =

class(TForm)

StringGrid1:

TStringGrid;

Button1:

TButton;

Memo1:

TMemo;

**procedure Button1Click(Sender:
TObject); private**

{ Private

declarations }

public

{ Public

declarations }

end;

var

Form1:

TForm1;

impleme

ntation

{ \$R *.dfm }

**procedure TForm1.Button1Click(Sender:
TObject); Var i,j,cod:integer;**

A:array[1..5,1..

5] of Real;

S:real;

s1:String;

begin

For i:=1 to 5 do For j:=1 to 5 do

Val(StringGrid1.cells[i-1,j-

1],a[i,j],cod); S:=0;

For i:=1 to 5 do For j:=1 to 5 do

s:=s+a[i,j];

Str(s:7:2,s1);

Memo1.Clear;

Memo1.Lines.add('Yig'in

di =' +s1); s:=s/25;

Str(s:7:2,s1);

Memo1.Lines.add('O'rtac

ha=' +s1); end; end.

3.5.10-rasm. Massiv elementlari yig'indisini va o'rta arifmetigini hisoblash dasturi

Izoh: hosil bo'lgan dastur yordamida 5 ta ustun va 5 ta satrdan tashkil topgan massiv

elementlarining yig'indisini va o'rta arifmetigini hisoblaydi.

Misol. $A(5)$ massiv elementlarining eng kichigi va element turgan o'rnini topish dasturini tuzamiz.

Dastur kodi va oynaning umumiy ko‘rinishini keltiramiz:

U
n
i
t
1
;
i
n
t
e
r
f
a
c
e
u
s
e
s

**Windows, Messages, SysUtils, Variants, Classes, Graphics,
Controls, Forms, Dialogs, StdCtrls, Grids;**

type

**TForm1 =
class(TForm)
StringGrid1:
TStringGrid;
Label1: TLabel;
Label2:
TLabel;
Button1:
TButton;
procedure Button1Click(Sender:
TObject); procedure
StringGrid1Click(Sender: TObject);
private
{ Private
declarations }
public
{ Public
declarations }
end;**

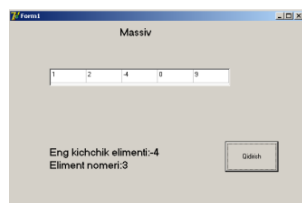
var

**Form1:
TForm1;
impleme
ntation
{ \$R *.dfm }
procedure TForm1.Button1Click(Sender:
TObject); const**

```

k=5;
var
a:array[1..k]
of integer;
min:integer;
i
:
i
n
t
e
g
e
r
;
b
e
g
i
n
  for i:=1 to k do
a[i]:=StrToInt(StringGrid1.Cells[i-1,0]); min:=1;
  for i:=2 to k do
  if a[i]< a[min]then min:=i;
  label2.caption:='Eng kichik
Elementi:'
+IntToStr(a[min])+#13+'Element nomeri:'
+IntToStr(min); end;
  procedure TForm1.StringGrid1Click(Sender:
TObject); var Key: Char;
  begin
    case Key of #8,'0'..'9': ;
  #13:
    if      StringGrid1.Col      <
StringGrid1.ColCount-1      then
StringGrid1.Col:= StringGrid1.Col
+ 1; else key := Chr(0);
  end; end; end.

```



3.5.11-rasm. Massiv elementlari ustida amallari

Izoh : Hosil bo'lgan dastur yordamida 5 ta elementdan tashkil topgan vektor elementlarining eng kichchigi va element nomerini topish mumkin. Buning uchun hosil qilingan dastur maydoniga yangi qiymat kiritish orqali natijani olishimiz mumkin.

5-Amaliy mashg'ulot.

Mavzu: Borland C++ Builderda funksiyalarga doir dastur tuzish.

Foydalanuvchi Funktsiyalari.

Funktsiyalarni ta'riflash va ularga murojaat qilish. Funktsiya ta'rifida funksiya nomi, tipi va formal parametrlar ro'yhati ko'rsatiladi. Formal parametrlar nomlaridan tashqari tiplari ham ko'rsatilishi shart. Formal parametrlar ro'yhati funksiya signaturasi deb ham ataladi. Funktsiya ta'rifi umumiy ko'rinishi quyidagichadir:

Funktsiya tipi funksiya nomi(formal_parametrlar_ta'rifi)

Formal parametrlarga ta'rif berilganda ularning boshlang'ich qiymatlari ham ko'rsatilishi mumkin. Funktsiya qaytaruvchi ifoda qiymati funksiya tanasida return <ifoda> ; operatori orqali ko'rsatiladi.

Misol:

```
Float min(float, float b)
{ if (a<b) return a; return b;
}
```

Funktsiyaga murojaat qilish quyidagicha amalga oshiriladi: Funktsiya nomi (haqiqiy parametrlar ruyhati)

Haqiqiy parametr ifoda ham bo'lishi mumkin. Haqiqiy parametrlar qiymati hisoblanib mos formal parametrlar o'rnida ishlatiladi.

Misol uchun yuqoridagi funksiya qo'yidagicha murojaat qilish mumkin: Int x=5,y=6,z; z=min(x,y) eki int z=Min(5,6) eki int x=5; int z=min(x,6)

Funktsiya ta'rifida formal parametrlar initsializatsiya qilinishi, ya'ni boshlang'ich qiymatlar ko'rsatilishi mumkin. Funktsiyaga murojaat qilinganda biror haqiqiy parametr ko'rsatilmasa, uning urniga mos formal parametr ta'rifida ko'rsatilgan boshlang'ich qiymat ishlatiladi.

Misol uchun:

```
Float min(float a=0.0, float b)
{ if (a<b) return a; return b;
}
```

Bu funksiya yuqorida ko'rsatilgan murojaat usullaridan tashqari quyidagicha murojaat qilish mumkin:

```
Int y=6,z; z=min(,y) eki int z=Min(,6);
```

Agar funksiya hech qanday qiymat qaytarmasa uning tipi void deb ko'rsatiladi. Misol uchun:

```
Void print;
{ Cout<<("\n Salom!");
};
```

Bu funksiya Print; shaklida murojlat qilish ekranga Salom! Yozilishiga olib keladi. Qiymat qaytarmaydigan funksiya formal parametrlarga ega bo'lishi mumkin:

```
Void Pint_Baho(Int baho);
{
```

```
Switch(baho)
{case 2:Cout<<("\n emon"); break; case
3:Cout<<("\n urta");break; case 4:Cout<<("\n
yahshi");break; case 5:Cout<<("\n a'lo");break;
default: Cout<<("\n baho notugri kiritilgan");
};
```

Bu funksiya Print_Baho(5) shaklida murojaat qilish ekranga a'lo so'zi yozilishiga olib keladi.

Agar programmada funksiya ta'rifi murojaatdan keyin berilsa, yoki funksiya boshqa faylda

joylashgan bo'lsa, murojlatdan oldin shu funksiyaning prototipi joylashgan bulishi kerak. 137

Prototip

funksiya nomi va formal parametrlar tiplaridan iborat bo'ladi. Formal parametrlar nomlarini berish shart emas.

Misol uchun $y = \min(a,b) + 2 * \max(c,d)$ ifodani hisoblashni

kuramaz: #Include <iostream.h>

```
int max(int a,int b)
```

```
{ if (a<b) return a;else
```

```
return b}; void main()
```

```
{ int a,b,c,d,y; int min(int ,int);
```

```
  Cin>>("n
```

```
%f%f%f%f',&a,&b,&c,&d);
```

```
y=min(a,b)+2*max(c,d);
```

```
Cout<<("n %f',y);
```

```
};
```

```
int min(int a,int b)
```

```
{ if (a<b) return b;else return a};
```

Funksiyaga parametrlar uzatish. Funksiyaga parametrlar qiymat buyicha uzatiladi va quyidagi bosqichlardan iborat bo'ladi:

1. Funksiya bajarishga tayyorlanganda formal parametrlar uchun hotiradan joy ajratiladi, ya'ni formal parametrlar funksiyalarning ichki parametrlariga aylantiriladi. Agar parametr tipi float bo'lsa double tipidagi ob'ektlar hosil buladi, char va shortint bulsa int tipidagi ob'ektlar yaratiladi.

2. Haqiqiy parametrlar sifatida ishlatilgan ifodalar qiymatlari hisoblanadi.

3. Haqiqiy parametrlar ifodalar qiymatlari formal parametrlar uchun ajratilgan hotira qismlariga yoziladi. Bu jarayonda float tipi double tipiga, char va shortint tiplari int tipiga keltiriladi.

4. Funksiya tanasi ichki ob'ektlar – parametrlar yordamida bajariladi va qiymat chaqirilgan joyga qaytariladi.

5. Haqiqiy parametrlar qiymatlariga funksiya hech qanday ta'sir o'tkazmaydi.

6. Funksiyadan chiqishda formal parametrlar uchun ajratilgan hotira qismlari bo'shatiladi.

C++ tilida chaqirilgan funksiya chaqiruvchi funksiyadagi o'zgaruvchi qiymatini uzgartira olmaydi. U faqat o'zining vaqtinchalik nushasini o'zgartirishi mumkin holos.

Qiymat bo'yicha chaqirish qulaylik tug'diradi. Chunki funksiyalarda kamroq o'zgaruvchilarni ishlatishga imkon beradi. Misol uchun shu xususiyatni aks ettiruvchi POWER funksiyasi variantini keltiramiz:

```
power(x,n) /* raise x n-th power; n >
```

```
0; version 2 */
```

```
int
```

```
x,n;
```

```
int p;
```

```
for (p = 1; n > 0; --
```

```
n) p = p * x;
```

```
return (p);
```

8-Laboratoriya mashg'uloti.

Mavzu: Borland C++ Builderda funksiyalarga doir dastur tuzish.

Ishning maqsadi: Talabalarga Borland C++ Builderda funksiyalarga doir dastur tuzishni o'rgatish.

Nazariya bo'yicha qisqacha ma'lumot

9-Laboratoriya mashg'uloti.

Mavzu: Borland C++ Builderda multimedia va animatsiyalarga doir dastur tuzish

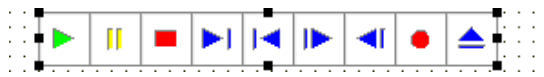
Ishning maqsadi: Talabalarga Borland C++ Builderda multimedia va animatsiyalarga doir dastur tuzishni o'rgatish.

Nazariya bo'yicha qisqacha ma'lumot

C++ Builder dasturlash tilida Media Player dasturini ishlab chiqish uchun biz yangi oynani ochamiz va bir necha amallar ketma-ketligini bajaramiz.

- a) C++ Builder oynasining **System** komponentalar palitrasidan **Media Player**

piktogrammasini tanlaymiz. Natijada ekranda quyidagi oyna hosil bo'ladi.



Biz ushbu hosil bo'lgan oyna ustiga sichqonchaning chap tugmasini 2 marta chertamiz va natijada ushbu piktogrammaning dasturlash maydoni ochiladi. Biz shu dasturlash maydoniga quyidagi kodni kiritamiz

```
voidfastcall TForm1::MediaPlayer1Click(TObject  
*Sender, TMPBtnType Button, bool &DoDefault)
```

- b) Dasturni kiritib bo'lganimizdan so'ng keying amallarni bajarishga o'tamiz. b) biz **Standart** komponentalar palitrasidan 2 marotaba **Scroll Bar** piktogrammasini tanlaymiz. Natijada ekranda quyidagi oyna hosil bo'ladi.



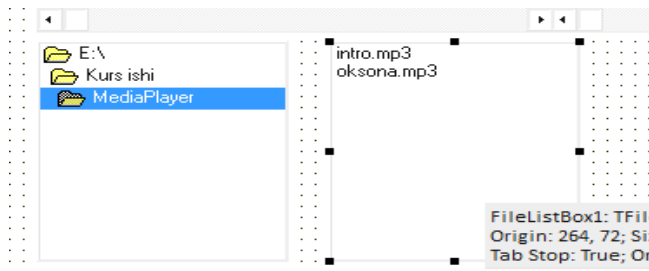
Biz 1-**Scroll Bar** piktogrammasi dasturlash maydoniga quyidagi kodlarni kiritamiz.

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

2-**Scroll Bar** piktogrammasi dasturlash maydoniga quyidagi kodlarni kiritamiz

```
TForm1::ScrollBar2Scroll(TObject  
*Sender, TScrollCode  
ScrollCode, int &ScrollPos)  
{ MediaPlayer1->volume = ScrollBar2->Position;}
```

- c) Endi **Win3.1** komponentalar palitrasiga o'tib **Directory List Box** va **File List Box** piktogrammasini tanlaymiz.



Dastlab **Directory List Box**

piktogrammasining

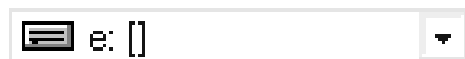
agi kodlarni kiritamiz:

```
voidfastcall TForm1::DirectoryListBox1Change(TObject *Sender){}
```

File List Box piktogrammasining dasturlash maydoniga quyidagi kodlarni kiritamiz.

```
TForm1::FileListBox1DbClick(TObject *Sender){ AnsiString s;  
s = DirectoryListBox1->Directory; if  
(s[s.Length()] != '\\')  
s += '\\';  
s += FileListBox1->Items->Strings[FileListBox1->ItemIndex]; MediaPlayer1->FileName = s;  
MediaPlayer1->DeviceType = dtAutoSelect; MediaPlayer1->Open();  
MediaPlayer1->Play();  
ScrollBar1->Max = MediaPlayer1->Length;  
ScrollBar1->Position = 0;
```

- d) Keyingi amallar ketma-ketligi quyidagicha amalga oshiriladi. Ya'ni **Win3.1** komponentalar palitrasiga o'tib **Drive Combo Box** piktogrammasini



tanlaymiz.

Drive Combo Box piktogrammasining dasturlash maydoniga quyidagi kodlarni kiritamiz.

```
TForm1::DriveComboBox1Change(TObject *Sender) { }
```

- e) Keyingi amallar ketma-ketligi quyidagicha amalga oshiriladi. Ya'ni **Win3.1** komponentalar palitrasiga o'tib **Filter Combo Box** piktogrammasini



tanlaymiz.

Filter Combo Box piktogrammasining dasturlash maydonchasiga quyidagi kodlarni kiritamiz.

```
TForm1::FilterComboBox1Change(TObject *Sender) { }
```

- f) Keyingi amallar ketma-ketligi quyidagicha amalga oshiriladi. Ya'ni **Win 32** komponentalar palitrasiga o'tib **Progress Bar** piktogrammasini tanlaymiz.

- g) Keyingi amallar ketma-ketligi quyidagicha amalga oshiriladi. Ya'ni **Win 32** komponentalar palitrasiga o'tib **Track Bar** piktogrammasini tanlaymiz.



Track Bar piktogrammasining dasturlash maydonchasiga quyidagi kodlarni kiritamiz.

```
TForm1::TrackBar1Change(TObject *Sender){ }
```

h) Keyingi amallar ketma-ketligi quyidagicha amalga oshiriladi. Ya'ni **System** komponentalar palitrasiga o'tib **Timer** piktogrammasini tanlaymiz.

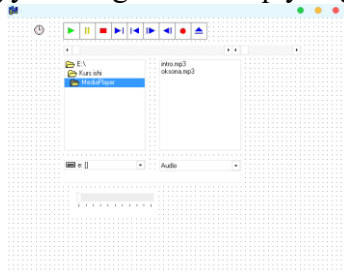


Timer piktogrammasining dasturlash maydonchasiga quyidagi kodlarni kiritamiz.

```
TForm1::Timer1Timer(TObject *Sender)
```

```
{ ScrollBar1->Position = MediaPlayer1->Position; }
```

Barcha amallarni bajarib bo'lganimizdan keyin dasturni to'liq saqlab bo'lib, uni ishga tushiramiz. Tayyor bo'lgan dastur quyidagicha ko'rinishga ega



Bu **Multimediali dastur** ilovasi **.mp3** kengaytmali fayllarni o'qishga mo'ljallangan.

10-Laboratoriya mashg'uloti.

Mavzu: Borland C++ Builderda grafikaga doir dastur tuzish

Ishning maqsadi: Talabalarga Borland C++ Builderda grafikaga doir dastur tuzishni o'rgatish.

Nazariya bo'yicha qisqacha ma'lumot

C++ Builder muhitida chizish sirti - TCanvas sinfi programma ishlash paytida rasm chizish imkonini beradi. Bu sinf obekti sirt bo'yicha ko'chish, grafik primitivlar chizish, rasmlarni va sirtning biror qismini nusxalash, hamda matnni chop qilish imkonini beruvchi xossa va metodlarni o'z ichiga oladi.

Har bir Canvas xossasiga ega komponenta o'z navbatida qalam, kist va shrift obektlarini tarkibiga oladi va mos ravishda Pen, Brush va Font xossalariga ega bo'ladi.

Pen xossasi rangga (Canvas->Pen-> Color), chizishning piksellerdagi qalinlikka (Canvas->

>Pen-> Width), chizilayotgan chiziq toifasiga (Canvas->Pen->Style) ega. Chiziq toifasi quyidagi qiymatlarni qabul qilishi mumkin:

psSolid – uzluksiz chiziq (kelishuv bo'yicha);

psDash – tire belgilaridan hosil bo'lgan chiziq;

psDot – nuqtalardan tashkil topgan chiziq;

psDashDot – nuqta va tire ketma-ketliklaridan iborat chiziq;

psDashDotDot – tire va nuqtalardan ketma-ketliklaridan iborat chiziq;

psClear – ko'rinmas chiziq;

psInsideFrame – chizish sirtini chegaralovchi to'g'ri to'rtburchak ichidagi chiziq.

Brush xossasi geometrik shakllar, masalan, to'g'ri to'rtburchak va ellips ichini to'ldirish naqshini aniqlaydi. U quyidagi xossalarga ega:

Canvas->Brush->Color – kist ranggi;

Canvas->Brush->Style – kist toifasini aniqlaydi va u quyidagi qiymatlarni qabul qilishi mumkin:

bsSolid – berilgan rang bilan shakl yuzasini to'liq bo'yaladi;

bsClear -shakl yuzasi bo'yalmaydi;

bsHorizontal - shakl yuzasi parallel chiziqlar bilan to'ldiriladi; bsVertical

- shakl yuzasi vertical chiziqlar bilan to'ldiriladi; bsFDiagonal - shakl

yuzasi yuqoriga qaragan chiziqlar bilan to'ldiriladi; bsFDiagonal - shakl

yuzasi yuqoriga pastga chiziqlar bilan to'ldiriladi; bsCross - shakl yuzasi

to'r bilan to'ldiriladi;

bsDiagCross - shakl yuzasi egri chiziqlardan hosil bo'lgan to'r bilan to'ldiriladi.

Canvas obekting muhim xossalariidan biri

Canvas->Pixels[x][y] xossasi bo'lib, u ko'rsatilgan koordinatadagi piksel rangini aniqlaydi.

Bu xossa qiymatini o'qish va unga qiymat yozish mumkin.

Geometric shakllar chizish uchun quyida keltirilgan funksiyalardan foydalanish mumkin:

Arc(int X1, int Y1, int X2, int Y2, int X3, int Y3, int X4, int Y4) – yoy chizish. Bu erda (X1,Y1) va (X2,Y2) – mos ravishda yoy chiziladigan to'rtburchak sohaning chap yuqori va o'ng past uchlari koordinatasi. (X3,Y3) va (X4,Y4) nuqtalar mos holda yoy boshlanishi va oxiri koordinatasi.

Chord(int X1, int Y1, int X2, int Y2, int X3, int Y3, int X4, int Y4) – ellips vatarini chizish. Bu erda (X1,Y1) va (X2,Y2) – mos ravishda yoy chiziladigan to'rtburchak sohaning chap yuqori va o'ng past uchlari koordinatasi. (X3,Y3) va (X4,Y4) nuqtalar mos holda vatar boshlanishi va oxiri koordinatasi.

Ellipse(int X1, int Y1, int X2, int Y2) – rang bilan to'ldirilgan ellipsni chizish. Bu erda (X1,Y1) va (X2,Y2) – mos ravishda yoy chiziladigan to'rtburchak sohaning chap yuqori va o'ng past uchlari koordinatasi.

Rectangle(int X1, int Y1, int X2, int Y2) – rang bilan to'ldirilgan to'g'ri to'rtburchakni chizish. Bu erda (X1,Y1) va (X2,Y2) – mos ravishda yoy chiziladigan to'rtburchak sohaning chap yuqori va o'ng past uchlari koordinatasi.

Canvas xossalariidan foydalangan holda uy rasmini chizishda ellips, to'g'ri to'rtburchak, ko'pburchak shakllari ishlatilgan. Rangni boshqarish, shakl yuzasini to'ldirish uchun qalam va kist xossalari ishlatilgan. Rasm olchamlari forma o'lchamiga mos ravishda o'zgaradi.

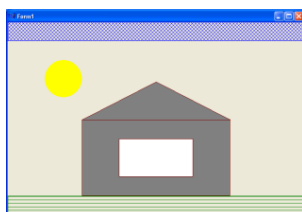
```
#include <vcl.h> #pragma
hdrstop #include "Unit1.h"
//-----
#pragma
package(smart_init
) #pragma resource
"*.dfm"

TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
: TForm(Owner)
{
}
//-----
void __fastcall TForm1::FormResize(TObject *Sender)
{
int w, h, wm, hm; Form1->Refresh();
```

```

wm=Form1->ClientWidth; w=wm/8; hm=Form1->ClientHeight; h=hm/10;
// Osmon
Form1->Canvas->Brush->Color=clBlue; Form1->Canvas->Brush-
>Style=bsDiagCross; Form1->Canvas->Pen->Color=clBlue;
Form1->Canvas->Rectangle(0,0,wm,h);
// Maysalar
Form1->Canvas->Brush->Color=clGreen; Form1->Canvas->Brush-
>Style=bsHorizontal; Form1->Canvas->Pen->Color=clGreen; Form1->Canvas-
>Rectangle(0,hm-h,wm,hm);
// Quyo'sh
Form1->Canvas->Brush->Color=clYellow; Form1->Canvas->Brush-
>Style=bsSolid; Form1->Canvas->Pen->Color=clYellow; Form1->Canvas-
>Ellipse(w,2*h,2*w,2*h+w);
// Uycha
Form1->Canvas->Brush->Color=clGray; Form1->Canvas->Brush->Style=bsSolid;
Form1->Canvas->Pen->Color=clMaroon;
Form1->Canvas->Rectangle(2*w, hm-5*h, 6*w, hm-h); TPoint point[4];
point[0].x=2*w; point[0].y=hm-5*h; point[1].x=4*w; point[1].y=hm-7*h;
point[2].x=6*w; point[2].y=hm-5*h; point[3].x=2*w; point[3].y=hm-5*h; Form1-
>Canvas->Polygon(point,3); Form1->Canvas->Brush->Color=clWhite; Form1-
>Canvas->Brush->Style=bsSolid; Form1->Canvas->Pen->Color=clMaroon;
Form1->Canvas->Rectangle(3*w, hm-4*h, 5*w, hm-2*h);
}

```



Programma ishga tushishi bilan Formmaning OnResize hodisasi ro'y beradi va ekranda uycha rasmi paydo bo'ladi.

11-Laboratoriya mashg'uloti.

Mavzu: Borland C++ Builderda satriy kattaliklar

Ishning maqsadi: Talabalarga Borland C++ Builderda satriy kattaliklardan foydalanib dasturlar tuzishni o'rgatish.

Nazariya bo'yicha qisqacha ma'lumot

Standart C++ tili ikki xildagi belgilar majmuasini qo'llab-quvvatlaydi. Birinchi toifaga, an'anaviy, "tor" belgilar deb nomlanuvchi 8-bitli belgilar majmuasi kiradi, ikkinchisiga 16-bitli "keng" belgilar kiradi. Til kutubxonasida har bir guruh belgilari uchun maxsus funktsiyalar to'plami aniqlangan.

C++ tilida satr uchun maxsus tur aniqlanmagan. Satr char turidagi belgilar massivi sifatida qaraladi va bu belgilar ketma-ketligi satr terminatori deb nomlanuvchi nol kodli belgi bilan tugaydi ('\0'). Odatda, nol-terminator bilan tugaydigan satrlarni ASCIIZ –satrlar deyiladi. Sart konstanta deb qo'shtirnoqlar ichiga olingan belgilar ketma-ketligiga aytiladi:

“Ushbu belgilar ketma-ketligiga satr deyiladi.”

Quyidagi jadvalda C++ tilida belgi sifatida ishlatilishi mumkin bo'lgan konstantalar to'plami keltirilgan.

Belgilar sinflari	Belgi konstantalar
-------------------	--------------------

Katta harflar	‘A’ ... ‘Z’, ‘А’ ... ‘Я’
Kichik harflar	‘a’ ... ‘z’, ‘а’ ... ‘я’
Raqamlar	‘0’ ... ‘9’
Bo‘sh joy	gorizontal tabulyatsiya (ASCII kodi 9), satrni o‘tkazish (ASCII kodi 10), vertikal tabulyatsiya (ASCII kodi 11), formani o‘tkazish (ASCII kodi 12), karetkani qaytarish (ASCII kodi 13)
Punktuatsiya belgilari (ajratuvchilar)	! ” # \$ % & ‘ () * + - . / : ; < = > ? @ [\] ^ _ { } ~
Boshqaruv belgilari	ASCII kodi 0...1Fh oralig‘ida va 7Fh bo‘lgan belgilar
Probel	ASCII kodi 32 bo‘lgan belgi
O‘n oltilik raqamlar	‘0’ ... ‘9’, ‘A’ ... ‘F’, ‘a’ ... ‘f’

Satr massivi e‘lon qilinishida satr oxiriga terminator qo‘yilishini va natijada satrga qo‘shimcha bitta bayt qo‘shilishi inobatga olinishi kerak:

```
char satr[10];
Ushbu e‘londa satr satri uchun jami 10 bayt ajratiladi – 9 satr hosil qiluvchi
belgilar uchun va 1 bayt terminator uchun.
Satr o‘zgaruvchilari e‘lon qilinishida boshlang‘ich qiymatlar qabul qilishi
mumkin. Bu holda kompilyator avtomatik ravishda satr uzunligi hisoblaydi va satr
oxiriga nol–terminatorni qo‘shib qo‘yadi:
char Hafta_kuni[]="Juma";
Ushbu e‘lon quyidagi e‘lon bilan
ekvivalent: char Hafta_kuni[]=
{‘J’,‘u’,‘m’,‘a’,‘\0’};
```

Satr qiymatini o‘qishda oqimli o‘qish operatori ">>" o‘rniga getline() funksiyasini ishltagan ma‘qul hisoblanadi, chunki oqimli o‘qishda probellar inkor qilinadi (garchi ular satr belgisi hisoblansa ham) va o‘qilayotgan belgilar ketma–ketligi satrdan "oshib" ketganda ham belgilarni kiritish davom etishi mumkin. Natijada satr o‘ziga ajratilgan o‘lchamdan ortiq belgilarni qabul qilishi mumkin. Shu sababli, getline() funksiyasi ikkita parametrga ega bo‘lib, birinchi parametr o‘qish amalga oshirilayotgan satrga ko‘rsatgich, ikkinchi parametrdan esa kiritilishi kerak bo‘lgan belgilar soni ko‘rsatiladi. Satrni getline() funksiyasi orqali o‘qishga misol ko‘raylik:

```
#include
<iostream.h
> int main()
{
char satr[6];
cout<<"Satrni
kiriting: "<<"\n";
cin.getline(satr,6);
cout<<"Siz kiritgan
satr: "<<satr; return 0;
}
```

Programmada satr satri 5 ta belgini qabul qilishi mumkin, ortiqchalari tashlab yuboriladi. getline() funksiyasiga murojaatda ikkinchi parametr qiymati o‘qilayotgan satr uzunligidan katta bo‘lmasligi kerak.

Satr bilan ishlaydigan funktsiyalarning aksariyati string.h kutubxonasida jamlangan. Nisbatan ko‘p ishlatiladigan funktsiyalarning tavsifini keltiramiz.

Satr uzunligini aniqlash funktsiyalari

Satrlar bilan ishlashda, aksariyat hollarda satr uzunligini bilish zarur bo‘ladi. Buning uchun string.h kutubxonasida strlen() funksiyasi aniqlangan bo‘lib, uning sintaksisi quyidagicha bo‘ladi:

size_t strlen (const char* string)

Bu funktsiya uzunligi hisoblanishi kerak bo'lgan satr boshiga ko'rsatgich bo'lgan yagona parametrga ega va u ishlash natijasi sifatida ishorasiz butun sonni qaytaradi. strlen() funktsiyasi satrning real uzunligidan bitta kam qiymat qaytaradi, ya'ni nol-terminator o'rni hisobga olinmaydi. Xuddi shu maqsadda sizeof() funktsiyasidan ham foydalanish mumkin va u strlen() funktsiyasidan farqli ravishda satrning real uzunligini qaytaradi. Quyida keltirilgan misolda satr

uzunligini hisoblashning har ikkita varianti keltirilgan: #include <iostream.h>

```
#include
<string.h>
using namespace std;
int
main()
{
char Str[]="1234567890";
cout
<<"strlen(Str)="<<strlen(Str)<<
endl;
cout<<"sizeof(Str)="<<sizeof(S
tr)<<endl; return 0;
}
```

Programma ishlashi natijasida ekranga

```
strlen(Str)=10 sizeof(Str)=11
xabarlari chiqadi.
```

Odatda sizeof() funktsiyasidan getline() funktsiyasining ikkinchi argumenti sifati ishlatiladi va satr uzunligini yaqqol ko'rsatmaslik imkonini beradi: cin.getline(Satr, sizeof(Satr));

Satrlarni nusxalash

Satr qiymatini biridan ikkinchisiga nusxalash mumkin. Buning uchun bir qator standart funktsiyalar aniqlangan bo'lib, ularning tavsiflari quyida keltiramiz.

strcpy() funktsiyasi prototipi

char* strcpy(char* str1, const char* str2)

ko'rinishga ega va bu funktsiya str2 ko'rsatib turgan satrdagi belgilarni str1 ko'rsatib turgan satrga baytma-bayt nusxalaydi. Nusxalash str2 ko'rsatib turgan satrdagi nol-terminal uchraguncha davom etadi. Shu sababli, str2 satr uzunligi str1 satr uzunligidan katta emasligiga ishonch hosil qilish kerak, aks holda berilgan sohasida (segmentida) str1 satrdan keyin joylashgan berilganlar "ustiga" str2 satrning "ortiqcha" qnomi yozilishi mumkin.

Navbatdagi programma qnomi "Satrni nusxalash!" satrini Str satrga nusxalaydi:

```
char Str[20];
strcpy(Str, "Satrni nusxalash!");
```

Zarur bo'lganda satrning qaysidir joyidan boshlab, oxirigacha nusxadash mumkin. Masalan, "Satrni nusxalash!" satrini 8 belgisidan boshlab nusxa olish zarur bo'lsa, uni quyidagicha yechish mumkin:

```
#include
<iostream.h>
#include <string.h>
int main()
{
char Str1[20]="Satrni nusxalash!"; char Str2[20];
char* kursatgich=Str1; kursatgich+=7; strcpy(Str2, kursatgich);
cout<<Str2<<endl; return 0;
}
```

strcpy() funktsiyasining strcpy() funktsiyasidan farqli joyi shundaki, unda bir satrdan ikkinchisiga nusxalanadigan belgilar soni ko'rsatiladi. Uning sintaksisi quyidagi ko'rinishda

ega:

```
char* strncpy(char* str1, const char* str2, size_t num)
```

Agar str1 satr uzunligi str2 satr uzunligidan kichik bo'lsa, ortiqcha belgilar "kesib" tashlanadi. strncpy() funktsiyasi ishlatilishiga misol ko'raylik:

```
#include <iostream.h> #include <string.h> int main()
{
    char Uzun_str[]="01234567890123456789";
    char
    Qisqa_str[]="ABCD
    EF";
    strncpy(Qisqa_str,U
    zun_str,4);
    cout          <<"Uzun_str="          "<<Uzun_str<<endl;
    cout<<"Qisqa_str="<<Qisqa_str<<endl; return 0;
}
```

Programmada Uzun_str satri boshidan 4 belgi Qisqa_str satriga oldingi qiymatlar ustiga nusxalanadi va natijada ekranga

```
01234567890123456789
```

```
0123EF
```

xabarlari chop etiladi.

strdup() funktsiyasiga yagona parametr sifatida satr-manbaga ko'rsatgich uzatiladi. Funktsiya, satrga mos xotiradan joy ajratadi, unga satrni nusxalaydi va yuzaga kelgan satr- nusxa adresini qaytaradi. strdup() funktsiya sintaksisi:

```
char* strdup(const char* source)
```

Quyidagi programma bo'lagida satr1 satrining nusxasi xotiraning satr2 ko'rsatgan joyida paydo bo'ladi:

```
char* satr1="Satr
nuxsasini olish."; char*
satr2;
satr2=strdup(satr1);
```

12-Laboratoriya mashg'uloti.

Mavzu: Borland C++ Builderda ko'p formalı loyihalar yaratish.

Ishning maqsadi: Talabalarga Borland C++ Builderda ko'p formalı loyihalar yaratishni o'rgatish.

Nazariya bo'yicha qisqacha ma'lumot

Asosiy va qo‘shimcha o‘quv adabiyotlar hamda axborot manbaalari

Asosiy adabiyotlar

1. Peter Gottschling. Discovering Modern C++. An Intensive Course for Scientists, Engineers, and Programmers. “Addison-Wesley”, 2015 y.
2. M.Ashurov, N.Mirzahmedova, N.Xaytullayeva. Algoritmash va dasturlash asoslari. Uslubiy qo‘llanma. T. : “Bayoz”, 2016 y.
3. A. R. Azamatov, B. Boltayev. Algoritmash va dasturlash asoslari. O‘quv qo‘llanma. T. : “Cho‘lpon”, 2010 y.
4. A. R. Azamatov, B. Boltayev. Algoritmash va dasturlash asoslari. O‘quv qo‘llanma. T. : “Cho‘lpon”, 2013 y.
5. Sh. I. Razzoqov, M. J. Yunusova. Dasturlash: Kasb-hunar kollejlari uchun o‘quv qo‘llanma. T. : “Ilm Ziyot”, 2011 y.
6. М. Ашуров, М. Мирмахмудов, Ш. Сапаев. Замонавий дастурлаш тиллари фанидан лаборатория ишлари. Т. : ТДПУ, 2008 й.
7. Меняев Михаил Федорович. Информационные технология управления. Москва, «Издательский ОмегаЛ», 2003 г.

Qo‘shimcha adabiyotlar

1. Мирзиёев Шавкат Миромонович. Эркин ва фаровон, демократик Ўзбекистон давлатини биргаликда барпо этамиз. Ўзбекистон Республикаси Президенти лавозимига киришиш тантанали маросимига бағишланган Олий Мажлис палаталарининг қўшма мажлисидаги нутқ / Ш.М. Мирзиёев. – Тошкент : Ўзбекистон, 2016. - 56 б.
2. Мирзиёев Шавкат Миромонович. Танқидий таҳлил, қатъий тартиб-интизом ва шахсий жавобгарлик – ҳар бир раҳбар фаолиятининг кундалик қондаси бўлиши керак. Мамлакатимизни 2016 йилда ижтимоий-иқтисодий ривожлантиришнинг асосий якуналари ва 2017 йилга мўлжалланган иқтисодий дастурнинг энг муҳим устувор йўналишларига бағишланган Вазирлар Маҳкамасининг кенгайтирилган мажлисидаги маъруза, 2017 йил 14 январ / Ш.М. Мирзиёев. – Тошкент : Ўзбекистон, 2017. – 104 б.
3. Мирзиёев Шавкат Миромонович. Қонун устуворлиги ва инсон манфаатларини таъминлаш – юрт тараққиёти ва халқ фаровонлигининг гарови. Ўзбекистон Республикаси Конституцияси қабул қилинганининг 24 йиллигига бағишланган тантанали маросимдаги маъруза. 2016 йил 7 декабр /Ш.М.Мирзиёев. – Тошкент: “Ўзбекистон”, 2017. – 48 б.
4. Мирзиёев Шавкат Миромонович. Буюк келажакимизни мард ва олижаноб халқимиз билан бирга қурамиз. Мазкур китобдан Ўзбекистон Республикаси Президенти Шавкат Мирзиёевнинг 2016 йил 1 ноябрдан 24 ноябрга қадар Қорақалпоғистон Республикаси, вилоятлар ва Тошкент шаҳри сайловчилари вакиллари билан ўтказилган сайловолди учрашувларида сўзлаган нутқлари ўрин олган. /Ш.М.Мирзиёев. – Тошкент: : “Ўзбекистон”, 2017. – 488 б
5. Ўзбекистон Республикаси Президентининг Фармони. Ўзбекистон республикасини янада ривожлантириш бўйича ҳаракатлар стратегияси тўғрисида. (*Ўзбекистон Республикаси қонун ҳужжатлари тўплами, 2017 й., 6-сон, 70-модда*)
6. Ўзбекистон Республикаси Конституцияси. Т.: Ўзбекистон. 2014. -46 б.
7. П. Дарахвелидзе, Э. Марков. Программирование в Delphi7. Учебник. Санкт-Петербург, “БХВ-Петербург” 2003 г.
8. В. М. Пестиков, А. Н. Маслובоев. Turbo PASCAL 7. 0. Изучаем на примерах. Санкт-Петербург. : “БХВ-Петербург”, 2004 г.
9. Фаронов В. В. Программирование на языке высокого уровня Delphi. Учебник. М. : “Питер”, 2003 г.
10. В.Т.Безручко. Практикум по курсу информатики. М. : «Финансы и статистика», 2004 г.

11. Nazirov Sh.A., Qobulov R.V., Bobojanov M.R., Raxmanov Q.S. C va C++ tili. “Vorish-nashriyot” MCHJ, Toshkent 2013. 488 b.
12. П. Дарахвелидзе, Э. Марков. Программирование в Delphi7. Учебник. Санкт-Петербург, “БХВ-Петербург” 2003 г.
13. Фаронов В. В. Программирование на языке высокого уровня Delphi. Учебник. М. : “Питер”, 2003 г.
14. В. Т. Безручко. Практикум по курсу информатики. М. : «Финансы и статистика», 2004 г.

Internet saytlari

1. www.zivonet.uz – Axborot ta’lim portali
2. www.edu.uz – Oliy va o’rta maxsus ta’lim vazirligi portali
3. www.tdpu.uz – **Nizomiy nomidagi TDPU rasmiy sayti**
4. <http://acm.tuit.uz/> - **dasturiy yechim to’g’riligini avtomatik testlovchi tizim.**
5. <http://acm.tuit.uz/forum/>, <http://acm.timus.ru/> – **dasturlarni testlovchi tizim.**

O'ZBEKISTON RESPUBLIKASI
OLIY VA O'RTA MAXSUS TA'LIM VAZIRLIGI
TOSHKENT DAVLAT PEDAGOGIKA UNIVERSITETI



Ro'yhatga olindi: BD -5110700-2.05

2019 - yil "17" 08

DASTURLASH TILLARI
FAN DASTURI

Bilim sohasi:	100000 – Gumanitar
Ta'lim sohasi:	110000 – Pedagogika
Ta'lim yo'nalishi:	5110700 – Informatika o'qitish metodikasi

TOSHKENT – 2019

O'zbekiston Respublikasi Oliy va o'rta maxsus ta'lim vazirligining 2019-yil "4" 10 dagi 892-sonli buyrug'i bilan ma'qullangan fan dasturlarini tayanch oliy ta'lim muassasasi tomonidan tasdiqlashga rozilik berilgan.

Fan dasturi Oliy va o'rta maxsus, kasb-hunar ta'limi yo'nalishlari bo'yicha O'quv-uslubiy birlashmalar faoliyatini Muvofiqlashtiruvchi Kengashning 2019-yil "17" 08 dagi 4-sonli bayonnomasi bilan ma'qullangan.

Fan dasturi Nizomiy nomidagi Toshkent Davlat pedagogika universitetid ishlab chiqildi.

Tuzuvchilar:

- M.O'.Ashurov – "Informatika va uni o'qitish metodikasi" kafedrası katta o'qituvchisi
- N.S.Xaytullayeva – "Informatika va uni o'qitish metodikasi" kafedrası katta o'qituvchisi, pedagogika fanlari bo'yicha falsafa doktori (PhD).

Taqrizchilar:

- Ziyadullayev D.SH. – Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti «Axborot ta'lim texnologiyalari» kafedrası mudiri, texnika fanlari nomzodi
- Yuldasheva U.T. – TTESI qoshidagi akademik litsey direktori, texnika fanlari nomzodi

Fan dasturi Nizomiy nomidagi Toshkent davlat pedagogika universiteti O'q -uslubiy Kengashida ko'rib chiqilgan va tavsiya qilingan (2019- yil "29" 06 dagi 11-sonli bayonnoma).

I. O'quv fanining dolzarbligi va oliy kasbiy ta'limdagi o'rni

Mustaqil Respublikamizda yuz berayotgan siyosiy, iqtisodiy, ilmiy-texnikaviy va madaniy o'zgarishlar Oliy ta'lim tizimida ham o'z aksini topmoqda. O'zbekistonda uzluksiz ta'lim-tarbiya tizimini yaratish, shu asosida ta'lim sifatini jaxon andozalari darajasiga etkazish ta'lim sistemasining eng dolzarb vazifasiga aylandi. Bu esa barcha mutaxassisliklar qatori Informatika va dasturlash bo'yicha kadrlar tayyorlash sifatini oshirishni ham taqozo etadi. Bu maqsad vazifalar ushbu fan dasturi mazmunini ham belgilaydi. Algoritmlar konsepsiyasining vujudga kelishi bilan algebra, sonlar nazariyasi, geometriya va matematikaning boshqa sohalariga tegishli bir qator muammolarning echimli yoki echimli emasligini aniqlashtirish imkonini berdi. Algoritmlik nazariyasi faoliyat sohasi EHMlar vujudga kelishi bilan yanada kengaydi. Yuqoridagi fikrlar "Dasturlash tillari" fanining asosiy mazmunini belgilashga yordam beradi.

"Dasturlash tillari" fani umumkasbiy fanlar blokiga kiritilgan kurs hisoblanib, 2- va 3-kurslarda o'qitilishi maqsadga muvofiq. "Dasturlash tillari" fani "Informatika o'qitish metodikasi" ta'lim yo'nalishida o'qitiladi. Mazkur fan Algoritmlik fanining nazariy va uslubiy asosini tashkil qilib, o'z rivojida aniq va tabiiy fanlar uchun zamin bo'lib xizmat qiladi.

II. O'quv fanining maqsadi va vazifasi

"Dasturlash tillari" fanini o'qitishdan maqsad – talabalarga dasturlashning ilmiy-nazariy asoslarini, informatika o'qituvchisining kasbiy sohasida egallashi lozim bo'lgan bilimlar, amalda qo'llash uchun ko'nikma va malakalarni shakllantirish hamda rivojlantirishdan iborat.

Ushbu maqsadga erishish uchun fan talabalarni ob'ektga yo'naltirilgan dasturlash tillarida ishlash, amaliy masalalarga dasturlar tuzishga oid nazariy bilimlar, amaliy ko'nikma va malakalarini shakllantirish vazifalarini bajaradi.

Fan bo'yicha talabalarning bilim, ko'nikma va malakalariga quyidagi talablar qo'yiladi. *Talaba:*

- ob'ektga yo'naltirilgan dasturlash tillarining nazariy asoslari, ob'yektlarni loyihalash, matematik va interfeys ob'yektlari, voqealar va xabarlar, ob'ektga yo'naltirilgan muhitlarda xabarlarni uzatish, ularga ishlov berish mexanizmlari, ob'yektlar iyerarxiyasi asosida dasturlarni loyihalash, muayyan ob'ektga yo'naltirilgan dasturlash tillari to'g'risida *tasavvurga ega bo'lishi*;
- ob'ektga yo'naltirilgan dasturlash tillarida chiziqli, tarmoqlanuvchi va takrorlanuvchi va modulli dasturlar tuzatishni, dasturlashning ob'ektga yo'naltirilgan paradigmasini, ob'ektga yo'naltirilgan muhitlarda dasturlarni loyihalashni *bilishi va ulardan foydalana olishi*;

I. O'quv fanining dolzarbligi va oliy kasbiy ta'limdagi o'rni

Mustaqil Respublikamizda yuz berayotgan siyosiy, iqtisodiy, ilmiy-texnikaviy va madaniy o'zgarishlar Oliy ta'lim tizimida ham o'z aksini topmoqda. O'zbekistonda uzluksiz ta'lim-tarbiya tizimini yaratish, shu asosida ta'lim sifatini jaxon andozalari darajasiga etkazish ta'lim sistemasining eng dolzarb vazifasiga aylandi. Bu esa barcha mutaxassisliklar qatori Informatika va dasturlash bo'yicha kadrlar tayyorlash sifatini oshirishni ham taqozo etadi. Bu maqsad vazifalar ushbu fan dasturi mazmunini ham belgilaydi. Algoritm konsepsiyasining vujudga kelishi bilan algebra, sonlar nazariyasi, geometriya va matematikaning boshqa sohalariga tegishli bir qator muammolarning echimli yoki echimli emasligini aniqlashtirish imkonini berdi. Algoritmik nazariyasi faoliyat sohasi EHMlar vujudga kelishi bilan yanada kengaydi. Yuqoridagi fikrlar "Dasturlash tillari" fanining asosiy mazmunini belgilashga yordam beradi.

"Dasturlash tillari" fani umumkasbiy fanlar blokiga kiritilgan kurs hisoblanib, 2- va 3-kurslarda o'qitilishi maqsadga muvofiq. "Dasturlash tillari" fani "Informatika o'qitish metodikasi" ta'lim yo'nalishida o'qitiladi. Mazkur fan Algoritmik fanining nazariy va uslubiy asosini tashkil qilib, o'z rivojida aniq va tabiiy fanlar uchun zamin bo'lib xizmat qiladi.

II. O'quv fanining maqsadi va vazifasi

"Dasturlash tillari" fanini o'qitishdan maqsad – talabalarga dasturlashning ilmiy-nazariy asoslarini, informatika o'qituvchisining kasbiy sohasida egallashi lozim bo'lgan bilimlar, amalda qo'llash uchun ko'nikma va makalalarni shakllantirish hamda rivojlantirishdan iborat.

Ushbu maqsadga erishish uchun fan talabalarni ob'ektga yo'naltirilgan dasturlash tillarida ishlash, amaliy masalalarga dasturlar tuzishga oid nazariy bilimlar, amaliy ko'nikma va malakalarini shakllantirish vazifalarini bajaradi.

Fan bo'yicha talabalarning bilim, ko'nikma va malakalariga quyidagi talablar qo'yiladi. *Talaba:*

- ob'ektga yo'naltirilgan dasturlash tillarining nazariy asoslari, ob'yektlarni loyihalash, matematik va interfeys ob'yektlari, voqealar va xabarlar, ob'ektga yo'naltirilgan muhitlarda xabarlarni uzatish, ularga ishlov berish mexanizmlari, ob'yektlar iyerarxiyasi asosida dasturlarni loyihalash, muayyan ob'ektga yo'naltirilgan dasturlash tillari to'g'risida *tasavvurga ega bo'lishi*;
- ob'ektga yo'naltirilgan dasturlash tillarida chiziqli, tarmoqlanuvchi va takrorlanuvchi va modulli dasturlar tuza olishni, dasturlashning ob'ektga yo'naltirilgan paradigmasini, ob'ektga yo'naltirilgan muhitlarda dasturlarni loyihalashni *bilishi va ulardan foydalana olishi*;

- ob'ektga yo'naltirilgan dasturlash tillari muhitida ishlash, masalalarni tahlil qila olish, muayyan dasturlash tillari yordamida masalalarning dasturini tuzish va natijalarni taqqoslay olish *ko'nikmalariga ega bo'lishi lozim*.

III. Asosiy nazariy qism (ma'ruza mashg'ulotlari)

Fanning nazariy mashg'ulotlari mazmuni

1-MODUL. OB'EKTGA YO'NALTIRILGAN DASTURLASH TILLARI

1-mavzu. "Dasturlash tillari" faniga kirish

Dasturlash tillari va ularning klassifikatsiyasi. Mashinaga mo'ljallangan va proseduraga mo'ljallangan dasturlash tillari. Yuqori darjali dasturlash tillari. Interpretatorlar va kompilyatorlar. Dasturlarni translyatsiyalash. Muayyan dasturlash tilining alifbosi, buruqlar tizimi va operatorlari.

2-mavzu. Ob'ektga yo'naltirilgan dasturlash tillari

Ob'ektga yo'naltirilgan dasturlash tillari. Dasturlashning ob'ektga yo'naltirilgan paradigmasi.

3-mavzu. Ob'ektga yo'naltirilgan loyihalash.

Ob'ektlarni loyihalash: satrlar, steklar, ro'yxatlar, navbatlar, daraxtlar. Matematik ob'ektlar: rasional va kompleks sonlar, vektorlar, matrisalar. Ob'ektlar kutubxonasi. Interfeys ob'ektlari: boshqarish elementlari, oynalar, dialoglar.

4-mavzu: Voqealar va habarlar.

Voqealar va habarlar. Ob'ektga yo'naltirilgan muhitlarda habarlarni uzatish va ularga ishlov berish mexanizmlari. Ob'ektlar ierarxiyasi asosida dasturlarni loyihalash. Muayyan ob'ektga yo'naltirilgan dasturlash tili va unda dastur tuzish asoslari

2-MODUL. DELPHI DASTURLASH TILI

5-mavzu. Delphi dasturlash tili ishchi muhiti

Delphi dasturlash tilining ishchi muhiti, undagi oynalar (Ob'ektlarning daraxtsimon ko'rinish oynasi, ob'ektlar inspektori oynasi, kod brauzeri oynasi, asosiy oyna, forma oynasi, dastur kodi oynasi), u o'rnatilishi zarur bo'lgan kompyuterga qo'yiladigan texnik talablar va instrumental tugmalar. Komponentlar palitrasi. Palitra bo'limlari va ayrim komponentlar xossalari bilan tanishish.

6-mavzu. Standard komponentlar palitrasi

Frame komponenti va uning xossalari. MainMenu, PopupMenu komponentlari. Label, Edit, Button, Memo, Panel komponentlari va ularning xossalari. CheckBox, ListBox, ComboBox, ListBox, RadioGroup, RadioButton, ScrollBar komponentlari.

7-mavzu. Additional komponentlar palitrasi

BitBtn, MaskEdit, StringGrid, MaskEdit, CheckListBox, DrawGrid, Image, Shape va boshqa komponentlaridan foydalanish. Komponentlar xossalari.

3-MODUL. DELPHIDA DASTURLASH

8-mavzu. Delphi dasturlari strukturasi, Loyiha va modul.

Bo'sh forma va uning modifikatsiyasi. Delphida nomlanishlar, forma xossalarini o'zgartirish. Formaga yangi komponent joylashtirish va unda komponent xossalaridan foydalanish, Xodisa tushunchasi. Loyiha va modul strukturasi. Dastur elementlari (alfavit, identifikatorlar, doimiyliklar, ifodalar va amallar).

9-mavzu. Delphida tiplar, o'zgarmaslar, o'zgaruvchilar va standart funksiyalar.

Delphida tiplar, ularning ahamiyati. Butun tiplar: sodda (tartib va haqiqiy) tiplar, mantiqiy va simvolli tiplar, tip-diaazon, vaqt-sana tipi. Delphida simvolli va satriy tiplar. Simvolli va satriy tiplarning berilishi, ular bilan bajariladigan amallar. Simvolli va satriy kattaliklar.

Delphi dasturlash tilida o'zgarmaslar, o'zgaruvchilar va standart funksiyalar

4-MODUL. DELPHI DASTURLASH TILI OPERATORLARI.

10-mavzu. Delphi dasturlash muxitida tarmoq operatorlari.

Tarkibiy va bo'sh operatorlar. Shart va mantiqiy ifodalar. If...Then...else shartli operatori. Tanlash (case) operatori. Goto o'tish operatori. Label (belgilar) xizmatchi so'zidan foydalanish qoidalari bilan tanishish.

11-mavzu. Delphi dasturlash muxitida siklik operatorlar.

Delphi dasturlash tilida sikllar. For sikli, While sikli, Repeat sikli. Murakkab sikllar.

12-mavzu. Delphida massivlar.

Delphi dasturlash tilida massivlar. Massivlarni tavsiflash, e'lon qilish. Massivlarni berilish usullari. Massiv elementlarini kiritish va chiqarish. Random (max) funksiyasi bilan tanishtirish. Ko'p o'lchovli massivlar. Massiv elementlarini saralash va tartiblash.

13-mavzu. Prosedura va funksiyalar

Prosedura va funksiyalar Delphi dasturlash tilining muhim instrumenti sifatida. Prosedura tarifi, uning nomi, undan foydalanish yo'llari. Funksiya tarifi, uning nomlanishi, undan dasturda foydalanish va uning proseduradan farqi.

14-mavzu. Delphi dasturlash tilining grafik vositalari.

Delphi dasturlash tilining grafik imkoniyatlari. Delphidagi maxsus TCanvas, TFont, TPen, TBrush klasslari. TFont klassi xossalari: Color, Name, Size, Style. TPen klassi xossalari: Color, Mode, Width, Style. TBrush klassi xossalari: Bitmap, Color, Style.

5-MODUL. C++ DASTURLASH TILIGA KIRISH

15-mavzu. C++ tilining leksik asoslari

C++ dasturlash tiliga kirish. C++ dasturlash tili alifbosi va xizmatchi so'zlari. Amallar. Izohlar satrini tavsiflash. C++ tilida operatorlar. Standart funksiyalar va ularning yozilishi. Konsol orqali muloqot qilish. Chiqarish operatori. Kiritish operatori

16-mavzu. O'zgaruvchi va o'zgarmas tipli kattaliklar.

O'zgaruvchilar. Identifikatorlar. Ma'lumotlar tipi. O'zgaruvchilarni e'lon qilish. O'zgaruvchilarni initsializatsiya qilish.

17-mavzu. Dasturlash operatorlari

C++ dasturlash tilidagi operatsiyalar. Arifmetik operatorlar. Qiymatni bir birlikka o'zgartiruvchi operatorlar. Taqqoslash operatorlari.

18-mavzu. Shartli operatorlar

C++ dasturlash tilida o'tish operatori. Shartli operatorning qisqa ko'rinishi. Shartli operatorning uzun ko'rinishi. Tanlash operatorlari. Shartsiz o'tish operatori. Ko'p tarmoqlanishlar va variant tanlash operatorlari.

19-mavzu. C++ dasturlash tilida takrorlanuvchi jarayonlar

Sikl operatorlari. Oldingi shartli While takrorlash operatori. Keyingi shartli do-while takrorlash operatori. For takrorlash operatori. Uzish break operatori. Umumiy takrorlanish algoritmlari va ichma-ich takrorlanishlar. Continue operatori.

20-mavzu. C++ dasturlash tilida funksiyalar

Funksiyalar haqida tushuncha va ularni yaratish. Funksiyalarning tuzilishi. Funksiya parametrlari. Lokal va global o'zgaruvchilar. Tipsiz funksiyalar. Void funksiyasi. Funksiyalardan foydalanish. Qiymat qaytarmaydigan funksiyalar va ular yordamida masala yechish.

21-mavzu. C++ dasturlash tilida massivlar

Massivlar haqida tushuncha. Massivlarni tavsiflash va ulardan foydalanish. Bir o'lchovli massivlar. Ko'p o'lchovli (indeksli) massivlar. Massivlarni navlarga ajratish usullari.

2-mavzu. C++ da ko'rsatkichlar

Adres (manzil) operatori. Jo'natish operatori. Ko'rsatkich tipidagi o'zgaruvchilarni e'lon qilish. Ko'rsatkichga boshlang'ich qiymat berish. Ko'rsatkich ustida amallar. Adresni olish amali. Ko'rsatkichlar va adres oluvchi o'zgaruvchilar funktsiya parametri sifatida. Ko'rsatkichlar va massivlar.

23-mavzu. C++ da satrlar va ular ustida amallar

Satr uzunligini aniqlash funktsiyalari. Satrlarni nusxalash, ulash, solishtirish. Satrdagi harflar registrini almashtirish. Satrda izlash funktsiyalari. Turlarni o'zgartirish funktsiyalari.

24-mavzu. C++ da strukturalar va birlashmalar

Strukturalar. Ma'lumot strukturalari. Struktura ko'rsatkichlari. Strukturalar bilan ko'rsatkich a'zolar. Birlashmalar va ular ustida amallar. Foydalanuvchi tomonidan aniqlangan berilganlar turi. Sinflar.

25-mavzu. C++ da fayllar bilan ishlash

Matn fayllarini o'qish va yozish. Oqimni ochish. Fayldan o'qish. Faylga yozish. Binary fayllar bilan ishlash operatorlari. Matn va binar fayllar. O'qish-yozish oqimlari. Standart oqimlar. Belgilarni o'qish-yozish funktsiyalari. Satrlarni o'qish - yozish funktsiyalari. Fayldan o'qish-yozish funktsiyalari. Formatli o'qish va yozish funktsiyalari. Fayl ko'rsatkichini boshqarish funktsiyalari.

6-MODUL. VIZUAL DASTURLAR TUZISH

26-mavzu. Borland C++ Builder dasturlash muhiti

Borland C++ Builder dasturlash muhitiga kirish, ishchi muhit, oynalar. C++ Bulder komponentlari va ularning xossalari. Komponentlar hodisalari va metodlari. Komponentlar tarkibi. Hodisalar. Uslublar. Loyihalar menejeri. C++ Builder da ilova dastur yaratish. Oddiy ilova dasturini yaratish.

27-mavzu. Borland C++ Builderda tiplar

Borland C++ Builder da butun va haqiqiy sonlar. Tiplarni almashtirish. Borland C++ Builderda simvolli va satriy tiplar. Simvolli va satriy tiplarning berilishi, ular bilan bajariladigan amallar.

28-mavzu. Borland C++ Builderda amallar, matematik funktsiyalar va tanlash operatorlari

Matematik funktsiyalar va doimiyliklardan foydalanish. Amallar va ularning bajarilish tartiblari. Mantiqiy amallar. If va Switch operatorlari.

29-mavzu. Borland C++ Builderda sikllar

For sikli va uning qo'llanilishi. While va do_while sikllari. Ichma-ich joylashgan sikllar.

30-mavzu. Borland C++ Builder komponentlarini o'rganish

Borland C++ Builder komponentlar palitrasi va komponentlar xossalari. Guruhli operatsiyalar uchun komponentlarni tanlash. Komponentlar o'lchovlarini

o'zgartirish. Matn muharriri ikova dasturini yaratish. Hodisa jarayonlarini yaratish. Menyu yaratish. Panel va menyu yaratuvchi komponentlar: Panel, GroupBox, Bevel, ScrollBox, ToolBar, StatusBar.

31-mavzu. Borland C++ Builder Standard komponentlar palitrasi

Frame komponenti va uning xossalari. MainMenu, PopupMenu komponentlari. Label, Edit, Button, Memo, Panel komponentlari va ularning xossalari. CheckBox, ListBox, ComboBox, ListBox, RadioGroup, RadioButton, ScrollBar komponentlari.

32-mavzu. Borland C++ Builder Additional komponentlar palitrasi

BitBtn, MaskEdit, StringGrid, MaskEdit, CheckListBox, DrawGrid, Image, Shape va boshqa komponentlaridan foydalanish. Komponentlar xossalari.

33-mavzu. Borland C++ Builder Dialogs komponentlar palitrasi

Borland C++ Builderdagi Dialogs komponentlar palitrasi. OpenFileDialog, SaveDialog, FindDialog, ColorDialog, FontDialog va hokazo komponentlar va ularning xossalari.

34-mavzu. Borland C++ Builderda komponentlar xodisalari va metodlari

Komponentlar xodisalari: OnClick, OnDblClick, OnKeyDown, OnKeyPress, OnKeyUp, OnEnter, OnExit, OnMouseDown va OnMouseUp, OnChange va hokazo.

Komponentlar metodlari: Add, Hide, Show, Delete, CanFocus, ChangeScale, TextOut, MoveTo, LineTo va hokazo.

35-mavzu. Borland C++ Builderda massivlar.

Borland C++ Builderda massivlarni tavsiflash, e'lon qilish. Borland C++ Builderda massiv elementlarini kiritish va chiqarish. Borland C++ Builderda massiv elementlarini saralash va tartiblash.

36-mavzu. Borland C++ Builderda grafik tasvirlar yaratish

Tayyor grafik fayllardan foydalanish. Grafik fayllar formati. Image Editor grafik muharriri. Tugmalar uchun piktogrammalar yaratish.

37-mavzu. Borland C++ Builderning grafik vositalari

Borland C++ Builderning maxsus TCanvas, TFont, TPen, Tbrush klasslari. TFont klassi xossalari: Color, Name, Size, Style. TPen klassi xossalari: Color, Mode, Width, Style. TBrush klassi xossalari: Bitmap, Color, Style.

38-mavzu. Borland C++ Builderda multimedia va animatsiyalar.

Borland C++ Builderda multimedia va animatsiyalar. C++ tilida ko'p formalı loyihalar yaratish.

IV. Amaliy mashg'ulotlar bo'yicha ko'rsatma va tavsiyalar

Amaliy mashg'ulotlarda talabalar muayyan masala bo'yicha mavjud bo'lgan yoki mustaqil tarzda kichik ishchi guruhlar yordamida hosil qilingan algoritmlarni

muhokama qiladilar. Mazkur mavzularga oid test masalalar tuzib, ular asosida tuzilgan dasturlar majmuasini tuzadilar va kompyuterda olingan natijalarni birgalikda tahlil qiladilar.

Amaliy mashg'ulotlar uchun quyidagi mavzular tavsiya etiladi:

1. Komponent xossalari dinamik va statik o'zgartirish.
2. Standard va Additional bo'lini komponentlaridan foydalanish yo'llari
3. Delphidagi dasturlarda tiplardan foydalanish.
4. Delphi dasturlash tilida tarkibiy operatorlar va tanlash operatori.
5. Delphi dasturlash tilida sikl operatorlari.
6. Delphi dasturlash tilida massivlar va satriy kartaliklar.
7. Delphi dasturlash tilida protsedura va funksiyalar
8. Delphida modullar va ulardan foydalanish
9. Delphi dasturlash tilining Office dasturlari bilan hamkorligi
10. Delphi dasturlash muxitida fayllar bilan ishlash. Mul'timedia ilovalari
11. Delphida MBni boshqaradigan ilovalar tuzish
12. Delphi ning grafik komponentlari.
13. C++ da ma'lumotlarning asosiy turlari bilan amallar bajarish.
14. C++ tilida chiziqli dasturlash
15. C++ tilida Shartli va shartsiz o'tish operatorlari. Tanlash operatori
16. C++ tilida takrorlanish operatorlari (while, do while, for)
17. C++ tilida massivlar
18. C++ tilida funksiyalar yaratish
19. C++ tilida strukturalar va birlashmalar
20. C++ tilida ko'rsatkichlar
21. C++ tilida sinflar
22. C++ tilida multimedia va animatsiyalar
23. Borland C++ Builderda sikllar bilan ishlash
24. Borland C++ Builderda massivlar bilan ishlash
25. Borland C++ Builderda funksiya va protseduralar bilan ishlash
26. Borland C++ Builderda fayllar bilan ishlash
27. Borland C++ Builderda ko'p formal ilovalar yaratish

Amaliy mashg'ulotlarni tashkil etish bo'yicha kafedra professor-o'qituvchilari tomonidan ko'rsatma va tavsiyalar ishlab chiqiladi. Unda talabalar asosiy ma'ruza mavzulari bo'yicha olgan bilim va ko'nikmalarini amaliy masalalarga dasturlar tuzish orqali bilimlarini yanada boyitadilar. Shuningdek, darslik va o'quv qo'llanmalar asosida talabalar bilimlarini mustahkamlashga erishish, tarqatma materiallardan foydalanish, ilmiy maqolalar va tezislarni chop etish orqali talabalar bilimni oshirish, masalalarning dasturini tuzish, mavzular bo'yicha ko'rgazmali qurollar tayyorlash va boshqalar tavsiya etiladi.

Amaliy mashg'ulotlar multimedia qurilmalari bilan jihozlangan auditoriyada bir akadem guruhga bir o'qituvchi tomonidan o'tkazilishi lozim. Mashg'ulotlar faol va interfaktiv usullar yordamida o'tilishi, mos ravishda munosib pedagogik va axborot texnologiyalar qo'llanilishi maqsadga muvofiq.

V. Laboratoriya mashg'ulotlari bo'yicha ko'rsatma va tavsiyalar

Laboratoriya mashg'ulotlarida talabalar amaliy mashg'ulotlarda tuzilgan dasturlarni kompyuter yordamida natijalarini ko'rib, ularni tahlil qiladilar va xulosalar chiqaradilar.

Laboratoriya mashg'ulotlari uchun quyidagi mavzular tavsiya etiladi:

1. Delphidagi dasturning interfeys qismini yaratish,
2. Delphida muloqot dasturi yaratish
3. Delphi dasturlash muhitida chiziqli dasturlar tuzish
4. Delphi dasturlash muhitida tarmoqlanuvchi dasturlar tuzish
5. Delphi dasturlash muhitida For siklik operatoridan foydalanish
6. Delphi dasturlash muhitida While siklik operatoridan foydalanish
7. Delphi dasturlash muhitida Repeat siklik operatoridan foydalanish
8. Delphida bir o'lchovli massivlarga doir dastur tuzish
9. Delphida ikki o'lchovli massivlarga doir dastur tuzish
10. Delphi dasturlash tilida to'plamlar bilan ishlash
11. Delphida simvolli kattaliklar bilan ishlash
12. Delphida satriy kattaliklar bilan ishlash
13. Delphi dasturlash muhitida funksiyalardan foydalanish
14. Delphi dasturlash muhitida proseduralardan foydalanish
15. Delphida standart modullar va ulardan foydalanish
16. Delphida modul yaratish va undan foydalanish
17. Delphi dasturlash tilining Word dasturi bilan hamkorligi
18. Delphi dasturlash tilining Excel dasturi bilan hamkorligi
19. Delphi dasturlash tilining Access dasturi bilan hamkorligi
20. Delphi dasturlash tilida grafik elementlar bilan ishlash
21. Delphi dasturlash muhitida fayllar bilan ishlash
22. Delphida multimediyali ilovalar yaratish
23. C++ dasturlash tilida ma'lumotlarning asosiy turlari bilan amallar bajarish
24. C++ dasturlash tilida qiymat berish operatorlari bilan ishlash
25. C++ dasturlash tilida arifmetik va mantiqiy amallar bilan ishlash
26. C++ dasturlash tilida chiziqli dasturlar tuzish
27. C++ dasturlash tilida If/Else strukturasi bilan ishlash
28. C++ dasturlash tilida Switch strukturasi bilan ishlash
29. C++ dasturlash tilida While siklik operatoridan foydalanish
30. C++ dasturlash tilida Do_While siklik operatoridan foydalanish
31. C++ dasturlash muhitida For takrorlash strukturasi bilan ishlash
32. Boshqaruv strukturalarida Continue va Break ifodalarini qo'llash
33. C++ dasturlash tilida funksiyalar yaratish va ulardan foydalanish
34. C++ da bir o'lchovli massivlar bilan ishlash
35. C++ da ikki o'lchovli massivlar bilan ishlash
36. C++ dasturlash tilida ko'rsatkichlar bilan ishlash
37. C++ da satriy kattaliklar bilan ishlash
38. C++ dasturlash tilida strukturalar bilan ishlash
39. C++ dasturlash tilida strukturada ko'rsatkichlardan foydalanish
40. C++ dasturlash tilida matnli fayllar bilan ishlash

41. C++ dasturlash tilida funktsiyalarni qayta yuklash
42. Borland C++ Builderda Label, Edit va Button komponentlaridan foydalanib dastur tuzish
43. Borland C++ Builderda CheckBox, RadioGroup, ComboBox va ListBox komponentlaridan foydalanib dastur tuzish
44. C++ tilida fayllar bilan ishlash. Dialog oynalari
45. C++ tilida panel va menyu yaratuvchi komponentlar
46. Borland C++ Builderda massivlarga doir dastur tuzish
47. Borland C++ Builderda funksiya va protseduralarga doir dastur tuzish
48. Borland C++ Builderda satriy kattaliklar
49. Borland C++ Builderda fayllar bilan ishlash.
50. Borland C++ Builderda grafikaga doir dastur tuzish
51. Borland C++ Builderda multimedia va animatsiyalarga doir dastur tuzish

VL. Mustaqil ta'lim va mustaqil ishlar

Mustaqil ta'lim uchun tavsiya etiladigan mavzular:

1. Yuqori darjali dasturlash tillari.
2. Interpretatorlar va kompilyatorlar.
3. Ob'ektga yo'naltirilgan dasturlash tillari.
4. Ob'ektga yo'naltirilgan loyihalash.
5. Ob'ektlar ierarxiyasi asosida dasturlarni loyihalash.
6. Delphi qo'llaniladigan matematik funktsiyalar
7. TForm komponentlari va ularning xossalari
8. Delphi dasturlash tilida sodda dasturlarni tuzish
9. Delphi dasturlash tilida shartli va siklli dasturlar tuzish
10. Delphi dasturlash tilida tasodifiy sonlar bilan ishlash
11. Delphi dasturlash tilida massivlar bilan ishlash
12. Delphi dasturlash tilida sana-vaqt turi bilan ishlash
13. Delphi dasturlash tilida satriy kattaliklar bilan ishlash
14. Delphi dasturlash tilida to'plamlar bilan ishlash
15. Delphi dasturlash tilida fayllar bilan ishlash
16. Delphi dasturlash tilida funksiya va proseduralarni yaratish
17. Delphi dasturlash tilida grafik primitivlar bilan ishlash
18. Delphi dasturlash tilida ListBox da grafiklarni joylashtirish
19. Biror predmet sohasiga oid o'rgatuvchi dasturlar yaratish
20. Delphi dasturlash tilida nazorat qiluvchi dasturlar yaratish
21. Delphi dasturlash tili komponentlar palitrasi bo'limlari va ayrim komponentlar xossalari.
22. Delphida forma xossalari va ularni o'zgartirish.
23. Delphi dasturlash tilida grafika.
24. Delphi dasturlash tilining multimedia imkoniyatlari.
25. C++ tilining boshqarish operatorlari
26. C++ tilida funktsiyalar, strukturalar va birlashmalar.
27. C++ tilida ko'rsatkichlar va murojaatlar

- 28.C++ tilida bir o'ldhovli, ikki o'ldhovli va dinamik massivlar.
- 29.C++ tilida satriy kattaliklar
- 30.C++ tilida sinflar,
- 31.C++ tilida grafika
- 32.C++ tilida multimedia
- 33.C++ tilida animatsiyalar
- 34.C++ tilida fayllar bilan ishlash.
- 35.C++ tilida Dialog oynalari,
- 36.C++ tilida panel yaratuvchi komponentlar,
- 37.C++ tilida menyu yaratuvchi komponentlar,
- 38.C++ tilidagi dasturlarning tarkibiy qismlari.
- 39.C++ da maxsus belgilar,
- 40.O'zgarmlar. Literal o'zgarmlar,
- 41.Belgili o'zgarmlar.
- 42.Ifodalar va operatorlar,
- 43.Matematik operatorlar. Operatorlar prioriteti.
- 44.Increment va decrement operatorlari
- 45.Prefiks va postfiks
- 46.Bloklar va kompleks ifodalar,
- 47.Xotirani zahiralash. Butun sonlar o'ldhami.
- 48.Ishorali va ishorasiz tiplar. O'zgaruvchilarning tayanch tiplari
- 49.Ma'lumotlar tipini keltirish (data casting)
- 50.Matematik kutubhona funksiyalari
- 51.Funksiyalarning tuzilishi
- 52.Funksiyalarning qo'llanilishi.
- 53.E'lon fayllari
- 54.Tasodifiy qiymatlarni keltirib chiqarish
- 55.if operatori orqali murakkab konstruksiyalarni hosil qilish
- 56.Dastur birliklarining sifatleri
- 57.O'zgaruvchining qo'llanilish sohasi (scope rules)
- 58.Argument olmaydigan funksiyalar
- 59.Ko'rsatkichlar va funksiya chaqiriqlarida ularning qo'llanilishi
- 60.Funksiya argumentlarning berilgan qiymatlari
- 61.Funksiya ismi yuklanishi
- 62.Funksiya shablonlari
- 63.Bir necha indeksli massivlar
- 64.Pointer (ko'rsatkich) va satrlar
- 65.Pointer operatorlari
- 66.Pointer argumentli funksiyalar
- 67.Const sifatli pointerlar
- 68.Pointer va oddiy o'zgaruvchilarning egallagan adres kattaligi
- 69.Borland C++ Builderda grafik axborotlar bilan ishlovchi komponentlar
- 70.Borland C++ Builderda Image va PaintBox komponentlarida foydalanish
- 71.Borland C++ Builderda Chart va VtChart komponentlarida foydalanish

72. Borland C++ Builderda Animation va MediaPlayer komponentlarida foydalanish
73. Borland C++ Builderda Win32 komponentlar palitrasidan foydalanish
74. Borland C++ Builderda System komponentlar palitrasidan foydalanish
75. Borland C++ Builderda ColorDialog va ColorBox komponentlari
76. Borland C++ Builderda PrintDialog va PrintSetupDialog komponentlari
77. Borland C++ Builderda Data Access komponentlar palitrasi
78. Borland C++ Builderda xossa va metodlar
79. Borland C++ Builderda massivlarni saralash
80. Borland C++ Builderning Excel dasturi bilan hamkorligi

Mustaqil o'zlashtiriladigan mavzular bo'yicha talabalar tomonidan referatlar tayyorlash va uni taqdimot qilish tavsiya etiladi.

Fan bo'yicha kurs ishi. Fan bo'yicha kurs ishi rejalashtirilmagan.

VI. Asosiy va qo'shimcha o'quv adabiyotlar hamda axborot manbaalari

Asosiy adabiyotlar

1. Peter Gottschling. Discovering Modern C++. An Intensive Course for Scientists, Engineers, and Programmers. "Addison-Wesley", 2015 y.
2. A. R. Azamatov, B. Boltayev. Algoritmash va dasturlash asoslari. O'quv qo'llanma. T.: "Cho'lpon", 2010 y.
3. A. R. Azamatov, B. Boltayev. Algoritmash va dasturlash asoslari. O'quv qo'llanma. T.: "Cho'lpon", 2013 y.
4. Sh. I. Razzoqov, M. J. Yunusova. Dasturlash: Kasb-hunar kollejlari uchun o'quv qo'llanma. T.: "Bim Ziya", 2011 y.
5. M. Ashurov, M. Mirmaxmudov, Sh. Sanaev. Zamонавий dasturlash tillari fanidan laboratoriya ishlari. T.: TDU, 2008 й.
6. Меньев Михаил Федорович. Информационные технологии управления. Москва, «Издательский Омега», 2003 г.

Qo'shimcha adabiyotlar

1. Мирзиёев Шавкат Миромонович. Эркин ва фаровон, демократик Ўзбекистон давлатини биргаликда барпо этамиз. Ўзбекистон Республикаси Президенти лавозимига юришши тантанали маросимига бағишланган Олий Мажлис палаталарининг қўшма мажлисидаги нутқ / Ш.М. Мирзиёев. – Тошкент : Ўзбекистон, 2016. - 56 б.
2. Мирзиёев Шавкат Миромонович. Танқадий таҳлил, катъий тартиб-интизом ва шахсий жавобгарлик – ҳар бир раҳбар фаолиятининг кундалик қондаси бўлиши керак. Мамлакатимизни 2016 йилда ижтимоий-иқтисодий ривожлантиришнинг асосий якунлари ва 2017 йилга мўлжалланган иқтисодий дастурининг энг муҳим устувор йўналишларига бағишланган Вазирлар Маҳкамасининг кенгайтирилган

- мажлисидаги маъруза, 2017 йил 14 январ / Ш.М. Мирзиёев. – Тошкент : Ўзбекистон, 2017. – 104 б.
3. Мирзиёев Шавкат Миромонович. Қонун устуворлиги ва инсон манфаатларини таъминлаш – юрт тараққиёти ва халқ фаровонлигининг гарови. Ўзбекистон Республикаси Конституцияси қабул қилинганининг 24 йиллигига бағишланган тантанали маросимдаги маъруза. 2016 йил 7 декабр /Ш.М.Мирзиёев. – Тошкент: “Ўзбекистон”, 2017. – 48 б.
 4. Мирзиёев Шавкат Миромонович. Буюк келажакимизни мард ва олижаноб халқимиз билан бирга курашимиз. Маъмур китобдан Ўзбекистон Республикаси Президенти Шавкат Мирзиёевнинг 2016 йил 1 ноябрдан 24 ноябрга қадар Қорақалпоғистон Республикаси, вилоятлар ва Тошкент шаҳри сайловчилари вакиллари билан ўтказилган сайловолди учрашувида сўзлаган нутқи Урин олган. /Ш.М.Мирзиёев. – Тошкент: : “Ўзбекистон”, 2017. – 488 б
 5. Ўзбекистон Республикаси Президентининг Фармони. Ўзбекистон республикасини янада ривожлантириш бўйича ҳаракатлар стратегияси тўғрисида. (Ўзбекистон Республикаси қонун ҳужжатлари тўплами, 2017 й., 6-сон, 70-модда)
 6. Ўзбекистон Республикаси Конституцияси. Т.: Ўзбекистон. 2014. -46 б.
 7. Nazirov Sh.A., Qobulov R.V., Bobojanov M.R., Rahmatov Q.S. C va C++ tili. “Voris-nashriyot” MCHJ, Toshkent 2013. 488 b.
 8. Безручко В.Т., Практикум по курсу информатики. М. : «Финансы и статистика», 2004 г.
 9. Дарахвелидзе П., Марков Э., Программирование в Delphi7. Учебник. Санкт-Петербург, “БХВ-Петербург” 2003 г.
 10. Пестиков В. М., Маслобоев А. Н., Turbo PASCAL 7. 0. Изучаем на примерах. Санкт-Петербург. : “БХВ-Петербург”, 2004 г.
 11. Фаронов В. В. Программирование на языке высокого уровня Delphi. Учебник. М. : “Питер”, 2003 г.

Internet saytlari

1. www.ziyounet.uz – Axborot ta’lim portali
2. www.edu.uz – Oliy va o’rta maxsus ta’lim vazirligi portali
3. www.tdpu.uz – Nizomiy nomidagi TDPU rasmiy sayti
4. <http://acm.tuit.uz/> - dasturiy yechim to’g’riligini avtomatik testlovchi tizim.
5. <http://acm.tuit.uz/forum/>, <http://acm.timus.ru/> – dasturlarni testlovchi tizim.

5. М. Ашуров, М. Мирмахмудов, Ш. Сапиев. Законовый дастурлаш тиллари фанидан лаборатория ишлари. Т. : ТДПУ, 2008 й.
6. Меньев Михаил Федорович. Информационные технологии управления. Москва, «Издательский Омега», 2003 г.

Qo'shimcha adabiyotlar

1. Мирзиёев Шавкат Миромонович. Эркин ва фаровон, демократик Ўзбекистон давлатини биргаликда барпо этамиз. Ўзбекистон Республикаси Президенти лавозимига киришни тантанали маросимига бағишланган Олий Мажлис палаталарининг қўшма мажлисидаги нутқ / Ш.М. Мирзиёев. – Тошкент : Ўзбекистон, 2016. - 56 б.
2. Мирзиёев Шавкат Миромонович. Танқидий таҳлил, қатъий тартиб-интизом ва шахсий жавобгарлик – ҳар бир раҳбар фаолиятининг кундалик қондаси бўлиши керак. Мамлакатимизни 2016 йилда ижтимоий-иқтисодий ривожлантиришнинг асосий йўндалари ва 2017 йилга мўлжалланган иқтисодий дастурнинг энг муҳим устувор йўналишларига бағишланган Вазирлар Маҳкамасининг кенгайтирилган мажлисидаги маъруза, 2017 йил 14 январ / Ш.М. Мирзиёев. – Тошкент : Ўзбекистон, 2017. – 104 б.
3. Мирзиёев Шавкат Миромонович. Қонун устуворлиги ва инсон манфаатларини таъминлаш – юрт тараққиёти ва халқ фаровонлигининг гарови. Ўзбекистон Республикаси Конституцияси қабул қилинганининг 24 йиллигига бағишланган тантанали маросимдаги маъруза. 2016 йил 7 декабр / Ш.М.Мирзиёев. – Тошкент: “Ўзбекистон”, 2017. – 48 б.
4. Мирзиёев Шавкат Миромонович. Буюк келажагимизни мард ва олижаноб халқимиз билан бирга қурамыз. Мазкур китобдан Ўзбекистон Республикаси Президенти Шавкат Мирзиёевнинг 2016 йил 1 ноябрдан 24 ноябрга қадар Қорақалпоғистон Республикаси, вилоятлар ва Тошкент шаҳри сайловчилари вакиллари билан ўтказилган сайловолди учрашувларида сўзлаган нутқлари ўрин олган. Ш.М.Мирзиёев. – Тошкент : “Ўзбекистон”, 2017. – 488 б.
5. Ўзбекистон Республикаси Президентининг Фармони. Ўзбекистон республикасини янада ривожлантириш бўйича ҳаракатлар стратегияси тўғрисида. (Ўзбекистон Республикаси қонун ҳужжатлари тўплами, 2017 й., 6-сон, 70-модда)
6. Ўзбекистон Республикаси Конституцияси. Т.: Ўзбекистон, 2014. -46 б.
7. Nazirov Sh.A., Qobulov R.V., Bobojanov M.R., Rahmatov Q.S. C va C++ tili. “Yoris-nashriyot” MCHJ, Toshkent 2013. 488 b.
8. Безручко В.Т.. Практикум по курсу информатики. М. : «Финансы и

- статистика», 2004 г.
9. Дарахвелидзе П., Марков Э.. Программирование в Delphi7. Учебник. Санкт-Петербург, "БХВ-Петербург" 2003 г.
 10. Постиков В. М., Маслобоев А. Н.. Turbo PASCAL 7. 0. Изучаем на примерах. Санкт-Петербург, : "БХВ-Петербург", 2004 г.
 11. Фаронов В. В. Программирование на языке высокого уровня Delphi. Учебник. М. : "Питер", 2003 г.

Internet saytlari

1. www.ziyouet.uz – Axborot ta'lim portali
2. www.edu.uz – Oliy va o'rta maxsus ta'lim vazirligi portali
3. www.tdpu.uz – Nizomiy nomidagi TDPU rasmiy sayti
4. <http://acm.tuit.uz/> - dasturiy yechim to'g'riligini avtomatik testlovchi tizim.
5. <http://acm.tuit.uz/forum/>, <http://acm.timus.ru/> – dasturlarni testlovchi tizim.

**O'ZBEKISTON RESPUBLIKASI OLIY VA O'RTA
MAXSUS TA'LIM VAZIRLIGI**

GULISTON DAVLAT UNIVERSITETI

Amaliy matematika va axborot texnologiyalari kafedrası

“Tasdiqlayman”

O'quv ishlari bo'yicha prorektor

F.G.Sharipov

“ 29 ” 08 2020 y.

DASTURLASH TILLARI

fani bo'yicha

ISHCHI O'QUV DASTURI

Bilim sohasi: 100000 - Gumanitar
Ta'lim sohasi: 110000 - Pedagogika
Ta'lim yo'nalishi: 5110700 - Informatika o'qitish metodikasi

Bosqich 3

Semestr V

Umumiy yuklama hajmi: 130

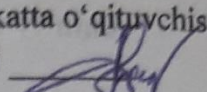
Umumiy o'quv soati: 76

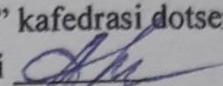
Shu jumladan:

Ma'ruza	- 24
Amaliy mashg'ulot	- 22
Laboratoriya mashg'uloti	- 30
Mustaqil ta'lim	- 54

GULISTON – 2020

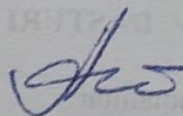
Fanning ishchi o'quv dasturi namunaviy o'quv dasturi va o'quv rejasiga muvofiq ishlab chiqildi.

Tuzuvchi: Abdurahimov D.B. – GulDU “Amaliy matematika va axborot texnologiyalari” kafedrasida katta o'qituvchisi, pedagogika fanlari nomzodi 

Taqrizchi: Allayorov S.P. – GulDU “Amaliy matematika va axborot texnologiyalari” kafedrasida dotsenti, texnika fanlari nomzodi 

Fanning ishchi o'quv dasturi “Amaliy matematika va axborot texnologiyalari” kafedrasining 2020 yil “27” 08 dagi 1 - sonli yig'ilishida muhokama qilindi va Fizika-matematika fakulteti Ilmiy – metodik Kengashida ko'rib chiqish uchun tavsiya qilindi

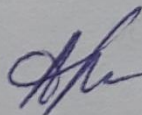
Kafedra mudiri:



A.A. Qalandarov

Fanning ishchi o'quv dasturi “Fizika-matematika” fakulteti Ilmiy-uslubiy Kengashining 2020 yil “28” 08 dagi 1 - sonli majlisida tasdiqlandi.

Fakultet Ilmiy-uslubiy
Kengashi raisi:



S.P. Allayorov

Fanning ishchi o'quv dasturi Guliston davlat universiteti O'quv-metodik Kengashining 2020 yil “29” avgustdagi 1 - sonli majlisida muhokama etildi va maqullandi.

I. O‘quv fanining dolzarbligi va oliy kasbiy ta’limdagi o‘rni

Mustaqil Respublikamizda yuz berayotgan siyosiy, iqtisodiy, ilmiy-texnikaviy va madaniy o‘zgarishlar Oliy ta’lim tizimida ham o‘z aksini topmoqda. O‘zbekistonda uzluksiz ta’lim-tarbiya tizimini yaratish, shu asosida ta’lim sifatini jahon andozalari darajasiga etkazish ta’lim sistemasining eng dolzarb vazifasiga aylandi. Bu esa barcha mutaxassisliklar qatori Informatika va dasturlash bo‘yicha kadrlar tayyorlash sifatini oshirishni ham taqozo etadi. Bu maqsad vazifalar ushbu fan dasturi mazmunini ham belgilaydi. Algoritm konsepsiyasining vujudga kelishi bilan algebra, sonlar nazariyasi, geometriya va matematikaning boshqa sohalariga tegishli bir qator muammolarning echimli yoki echimli emasligini aniqlashtirish imkonini berdi. Algoritm nazariyasi faoliyat sohasi EHMlar vujudga kelishi bilan yanada kengaydi. Yuqoridagi fikrlar “Dasturlash tillari” fanining asosiy mazmunini belgilashga yordam beradi.

“Dasturlash tillari” fani umumkasbiy fanlar blokiga kiritilgan kurs hisoblanib, 2- va 3-kurslarda o‘qitilishi maqsadga muvofiq. “Dasturlash tillari” fani “Informatika o‘qitish metodikasi” ta’lim yo‘nalishida o‘qitiladi. Mazkur fan Algoritm nazariyasi nazariy va uslubiy asosini tashkil qilib, o‘z rivojida aniq va tabiiy fanlar uchun zamin bo‘lib xizmat qiladi.

II. O‘quv fanining maqsadi va vazifasi

“Dasturlash tillari” fanini o‘qitishdan maqsad – talabalarga dasturlashning ilmiy-nazariy asoslarini, informatika o‘qituvchisining kasbiy sohasida egallashi lozim bo‘lgan bilimlar, amalda qo‘llash uchun ko‘nikma va makalalarni shakllantirish hamda rivojlantirishdan iborat.

Ushbu maqsadga erishish uchun fan talabalarni ob‘ektga yo‘naltirilgan dasturlash tillarida ishlash, amaliy masalalarga dasturlar tuzishga oid nazariy bilimlar, amaliy ko‘nikma va malakalarini shakllantirish vazifalarini bajaradi.

Fan bo‘yicha talabalarning bilim, ko‘nikma va malakalariga quyidagi talablar qo‘yiladi. **Talaba:**

- ob‘yektga yo‘naltirilgan dasturlash tillarining nazariy asoslari, ob‘yektlarni loyihalash, matematik va interfeys ob‘yektlari, voqealar va xabarlar, ob‘yektga yo‘naltirilgan muhitlarda xabarlarni uzatish, ularga ishlov berish mexanizmlari, ob‘yektlar iyerarxiyasi asosida dasturlarni loyihalash, muayyan ob‘yektga yo‘naltirilgan dasturlash tillari to‘g‘risida **tasavvurga ega bo‘lishi;**
- ob‘yektga yo‘naltirilgan dasturlash tillarida chiziqli, tarmoqlanuvchi va takrorlanuvchi va modulli dasturlar tuzishni, dasturlashning ob‘yektga yo‘naltirilgan paradigmasini, ob‘yektga yo‘naltirilgan muhitlarda dasturlarni loyihalashni **bilishi va ulardan foydalana olishi;**
- ob‘yektga yo‘naltirilgan dasturlash tillari muhitida ishlash, masalalarni tahlil qila olish, muayyan dasturlash tillari yordamida masalalarning dasturini tuzish va natijalarni taqqoslay olish **ko‘nikmalariga ega bo‘lishi lozim.**

Fandan o‘tiladigan mavzular va ular bo‘yicha mashg‘ulot turlariga ajratilgan soatlarning taqsimoti

№	Mavzu	Soatlar				
		Jami	Ma’ruza	Ama- liy	Labor	Mustaqi l ta’lim
	5 - semestr					
1	Borland C++ Builder dasturlash muhiti	2	2			
2	Borland C++ Builderda tiplar. Borland C++ Builderda amallar, matematik funktsiyalar va tanlash operatorlari	2	2			
3	Borland C++ Builderda amallar, matematik funktsiyalar va tanlash operatorlari bilan ishlash	2		2		
4	Borland C++ Builderda chiziqli dasturlar tuzish	4		2	2	
5	Borland C++ Builderda sikllar	2	2			
6	Borland C++ Builder komponentlarini o‘rganish	2	2			
7	Borland C++ Builder Standard komponentlar palitrasi	2	2			
8	Borland C++ Builder Standard komponentlar palitrasi komponentlari bilan ishlash	4		2		2
9	Borland C++ Builderda tarmoqlanuvchi dasturlar tuzish	4		2	2	
10	Borland C++ Builder Additional komponentlar palitrasi	6	2			4
11	Borland C++ Builder Additional komponentlar palitrasi komponentlari bilan ishlash	4		2		
12	Borland C++ Builderda sikllar bilan ishlash	2		2		
13	Borland C++ Builderda CheckBox, RadioGroup, ComboBox va ListBox komponentlaridan foydalanib dastur tuzish	4			4	
14	Borland C++ Builderda Win32 komponentlar palitrasidan foydalanish	4				4
15	Borland C++ Builder Dialogs komponentlar palitrasi	2	2			
16	Borland C++ Builderda ColorDialog va ColorBox komponentlari	4				4
17	Borland C++ Builder Dialogs komponentlar palitrasi komponentlari bilan ishlash	2		2		
18	Borland C++ Builderda System komponentlar palitrasidan foydalanish	4				4
19	Borland C++ Builderda komponentlar xodisalari va metodlari	2	2			
20	Borland C++ Builderda panel va menyu yaratuvchi komponentlar	2			2	
21	Borland C++ Builderda Data Access komponentlar palitrasi	4				4
22	Borland C++ Builderda massivlar	2	2			
23	Borland C++ Builderda massivlarni saralash	4				4
24	Borland C++ Builderda massivlar bilan ishlash	2		2		
25	Borland C++ Builderda massivlarga doir dastur tuzish	4			4	
26	Borland C++ Builderda funksiya va protseduralar	2		2		

	bilan ishlash					
27	Borland C++ Builderda funksiya va protseduralarga doir dastur tuzish	4			4	
	Borland C++ Builderda grafik tasvirlar yaratish	2	2			
	Borland C++ Builderda grafik axborotlar bilan ishlovchi komponentlar	4				4
	Borland C++ Builderda fayllar bilan ishlash	2		2		
	C++ tilida fayllar bilan ishlash. Dialog oynalari	2			2	
	Borland C++ Builderda xossa va metodlar	4				4
	Borland C++ Builderda satriy kattaliklar	2			2	
	Borland C++ Builderning grafik vositalari	2	2			
	Borland C++ Builderda grafikaga doir dastur tuzish	4			4	
	Borland C++ Builderda Image va PaintBox komponentlarida foydalanish	4				4
	Borland C++ Builderda Chart va VtChart komponentlarida foydalanish	4				4
	Borland C++ Builderda multimedia va animatsiyalar.	2	2			
	Borland C++ Builderda PrintDialog va PrintSetupDialog komponentlari	4				4
	Borland C++ Builderda ko'p formalı ilovalar yaratish	2		2		
	Borland C++ Builderda multimedia va animatsiyalarga doir dastur tuzish	4			4	
	Borland C++ Builderda Animation va MediaPlayer komponentlarida foydalanish	4				4
	Borland C++ Builderning Excel dasturi bilan hamkorligi	4				4
	5 - semestr jami:	130	24	22	30	54

III. Asosiy nazariy qism (ma'ruza mashg'ulotlari) Fanning nazariy mashg'ulotlari mazmuni

6-MODUL. VIZUAL DASTURLAR TUZISH

1-mavzu. Borland C++ Builder dasturlash muhiti. (2 soat).

Borland C++ Builder dasturlash muhitiga kirish, ishchi muhit, oynalar. C++ Bulder komponentlari va ularning hossalari. Komponentlar hodiasalari va metodlari. Komponentlar tarkibi. Hodisalar. Uslublar. Loyihalar menejeri. C++ Builder da ilova dastur yaratish. Oddiy ilova dasturini yaratish.

2-mavzu. Borland C++ Builderda tiplar. Borland C++ Builderda amallar, matematik funktsiyalar va tanlash operatorlari (2 soat).

Borland C++ Builder da butun va haqiqiy sonlar. Tiplarni almashtirish. Borland C++ Builderda simvolli va satriy tiplar. Simvolli va satriy tiplarning berilishi, ular bilan bajariladigan

amallar. Matematik funktsiyalar va doimiyliklardan foydalanish. Amallar va ularning bajarilish tartiblari. Mantiqiy amallar. If va Switch operatorlari.

3-mavzu. Borland C++ Builderda sikllar. (2 soat).

For sikli va uning qo'llanilishi. While va do_while sikllari. Ichma-ich joylashgan sikllar.

4-mavzu. Borland C++ Builder komponentlarini o'rganish. (2 soat).

Borland C++ Builder komponentlar palitrasi va komponentlar xossalari. Guruhli operatsiyalar uchun komponentlarni tanlash. Komponentlar o'lchovlarini o'zgartirish. Matn muharriri ilova dasturini yaratish. Hodisa jarayonlarini yaratish. Menyu yaratish. Panel va menyu yaratuvchi komponentlar: Panel, GroupBox, Bevel, ScroolBox, ToolBar, StatusBar.

5-mavzu. Borland C++ Builder Standard komponentlar palitrasi. (2 soat).

Frame komponenti va uning xossalari. MainMenu, PopupMenu komponentlari. Label, Edit, Button, Memo, Panel komponentlari va ularning xossalari. CheckBox, ListBox, ComboBox, ListBox, RadioGroup, RadioButton, ScrollBar komponentlari.

6-mavzu. Borland C++ Builder Additional komponentlar palitrasi. (2 soat).

BitBtn, MaskEdit, StringGrid, MaskEdit, CheckListBox, DrawGrid, Image, Shape va boshqa komponentlaridan foydalanish. Komponentlar xossalari.

7-mavzu. Borland C++ Builder Dialogs komponentlar palitrasi. (2 soat).

Borland C++ Builderdagi Dialogs komponentlar palitrasi. OpenFileDialog, SaveDialog, FindDialog, ColorDialog, FontDialog va hokazo komponentlar va ularning xossalari.

8-mavzu. Borland C++ Builderda komponentlar xodisalari va metodlari. (2 soat).

Komponentlar xodisalari: OnClick, OnDblClick, OnKeyDown, OnKeyPress, OnKeyUp, OnEnter, OnExit, OnMouseDown va OnMouseUp, OnChange va hokazo.

Komponentlar metodlari: Add, Hide, Show, Delete, CanFocus, ChangeScale, TextOut, MoveTo, LineTo va hokazo.

9-mavzu. Borland C++ Builderda massivlar. (2 soat).

Borland C++ Builderda massivlarni tavsiflash, e'lon qilish. Borland C++ Builderda massiv elementlarini kiritish va chiqarish. Borland C++ Builderda massiv elementlarini saralash va tartiblash.

10-mavzu. Borland C++ Builderda grafik tasvirlar yaratish. (2 soat).

Tayyor grafik fayllardan foydalanish. Grafik fayllar formati. Image Editor grafik muharriri. Tugmalar uchun piktogrammalar yaratish.

11-mavzu. Borland C++ Builderning grafik vositalari. (2 soat).

Borland C++ Builderning maxsus TCanvas, TFont, TPen, Tbrush klasslari. TFont klassi xossalari: Color, Name, Size, Style. TPen klassi xossalari: Color, Mode, Width, Style. TBrush klassi xossalari: Bitmap, Color, Style.

12-mavzu. Borland C++ Builderda multimedia va animatsiyalar. (2 soat).

Borland C++ Builderda multimedia va animatsiyalar. C++ tilida ko'p formalı loyihalar yaratish.

Amaliy mashg'ulotlar mazmuni.

1-mavzu. Borland C++ Builderda amallar, matematik funktsiyalar va tanlash operatorlari bilan ishlash. (2 soat).

Talabalarni Borland C++ Builderda amallar, matematik funktsiyalar va tanlash operatorlari bilan ishlashni o'rgatish.

2-mavzu. Borland C++ Builderda chiziqli dasturlar tuzish. (2 soat).

Talabalarni Borland C++ Builderda chiziqli dasturlar tuzishni o'rgatish.

3-mavzu. Borland C++ Builder Standard komponentlar palitrasi komponentlari bilan

ishlash. (2 soat).

Talabalarni Borland C++ Builder Standard komponentlar palitrasi komponentlari bilan ishlashga doir dasturlar tuzishga o'rgatish.

4-mavzu. Borland C++ Builderda tarmoqlanuvchi dasturlar tuzish. (2 soat).

Talabalarni Borland C++ Builderda tarmoqlanuvchi dasturlar tuzishga o'rgatish.

5-mavzu. Borland C++ Builder Additional komponentlar palitrasi komponentlari bilan ishlash. (2 soat).

Talabalarni Borland C++ Builder Additional komponentlar palitrasi komponentlari bilan ishlashga o'rgatish.

6-mavzu. Borland C++ Builderda sikllar bilan ishlash. (2 soat).

Talabalarni Borland C++ Builderda sikllar bilan ishlash foydalanib dastur tuzish ko'nikmalariga ega bo'lishni o'rgatish.

7-mavzu. Borland C++ Builder Dialogs komponentlar palitrasi komponentlari bilan ishlash. (2 soat).

Talabalarni Borland C++ Builder Dialogs komponentlar palitrasi komponentlari imkonioatlaridan foydalanib dastur tuzish ko'nikmalariga ega bo'lishni o'rgatish.

8-mavzu. Borland C++ Builderda massivlar bilan ishlash. (2 soat).

Talabalarni Borland C++ Builderda massivlar bilan ishlash ko'nikmalariga ega bo'lishni o'rgatish.

9-mavzu. Borland C++ Builderda funksiya va protseduralar bilan ishlash. (2 soat).

Talabalarni Borland C++ Builderda funksiya va protseduralar bilan ishlashga o'rgatish.

10-mavzu. Borland C++ Builderda fayllar bilan ishlash. (2 soat).

Talabalarni Borland C++ Builderda fayllar bilan ishlash ko'nikmalariga ega bo'lishni o'rgatish.

11-mavzu. Borland C++ Builderda ko'p formalı ilovalar yaratish. (2 soat).

Talabalarni Borland C++ Builderda ko'p formalı ilovalar yaratish ko'nikmalariga ega bo'lishni o'rgatish.

Amaliy mashg'ulotlar bo'yicha ko'rsatma va tavsiyalar

Amaliy mashg'ulotlarda talabalar muayyan masala bo'yicha mavjud bo'lgan yoki mustaqil tarzda kichik ishchi guruhlari yordamida hosil qilingan algoritmlarni muhokama qiladilar. Mazkur mavzularga oid test masalalar tuzib, ular asosida tuzilgan dasturlar majmuasini tuzadilar va kompyuterda olingan natijalarni birgalikda tahlil qiladilar.

Amaliy mashg'ulotlarni tashkil etish bo'yicha kafedra professor-o'qituvchilari tomonidan ko'rsatma va tavsiyalar ishlab chiqiladi. Unda talabalar asosiy ma'ruza mavzulari bo'yicha olgan bilim va ko'nikmalarini amaliy masalalarga dasturlar tuzish orqali bilimlarini yanada boyitadilar. Shuningdek, darslik va o'quv qo'llanmalar asosida talabalar bilimlarini mustahkamlashga erishish, tarqatma materiallardan foydalanish, ilmiy maqolalar va tezislarni chop etish orqali talabalar bilimini oshirish, masalalarning dasturini tuzish, mavzular bo'yicha ko'rgazmali qurollar tayyorlash va boshqalar tavsiya etiladi.

Amaliy mashg'ulotlar multimedia qurilmalari bilan jihozlangan auditoriyada bir akadem guruhga bir o'qituvchi tomonidan o'tkazilishi lozim. Mashg'ulotlar faol va interaktiv usullar yordamida o'tilishi, mos ravishda munosib pedagogik va axborot texnologiyalar qo'llanilishi maqsadga muvofiq.

Laboratoriya mashg'ulotlarining mazmuni.

1-mavzu. Borland C++ Builderda chiziqli dasturlar tuzish. (2 soat).

Talabalarni Borland C++ Builderda chiziqli dasturlar tuzishga o'rgatish.

2-mavzu. Borland C++ Builderda tarmoqlanuvchi dasturlar tuzish. (2 soat).

Talabalarni Borland C++ Builderda tarmoqlanuvchi dasturlar tuzishga o'rgatish.

3-mavzu. Borland C++ Builderda CheckBox, RadioGroup, ComboBox va ListBox

komponentlaridan foydalanib dastur tuzish. (4 soat).

Talabalarni Borland C++ Builderda CheckBox, RadioGroup, ComboBox va ListBox komponentlaridan foydalanib dastur tuzishga o'rgatish.

4-mavzu. Borland C++ Builderda panel va menyu yaratuvchi komponentlar. (2 soat).

Talabalarni Borland C++ Builderda panel va menyu yaratuvchi komponentlar yaratishga o'rgatish.

5-mavzu. Borland C++ Builderda massivlarga doir dastur tuzish. (4 soat).

Talabalarni Borland C++ Builderda massivlarga doir dastur tuzishga o'rgatish.

6-mavzu. Borland C++ Builderda funksiya va protseduralarga doir dastur tuzish. (4 soat).

Talabalarni Borland C++ Builderda funksiya va protseduralarga doir dastur tuzishga o'rgatish.

7-mavzu. Borland C++ Builderda satriy kattaliklar. (2 soat).

Talabalarni Borland C++ Builderda satriy kattaliklar bilan ishlashga o'rgatish.

8-mavzu. Borland C++ Builderda fayllar bilan ishlash. Dialog oynalari. (2 soat).

Talabalarni Borland C++ Builderda fayllar bilan ishlash va Dialog oynalarini hosil qilishga o'rgatish..

9-mavzu. Borland C++ Builderda grafikaga doir dastur tuzish. (4 soat).

Talabalarni Borland C++ Builderda grafikaga doir dastur tuzishga o'rgatish.

10-mavzu. Borland C++ Builderda multimedia va animatsiyalarga doir dastur tuzish. (4 soat).

Talabalarni Borland C++ Builderda multimedia va animatsiyalarga doir dastur tuzishga o'rgatish.

Laboratoriya mashg'ulotlari bo'yicha ko'rsatma va tavsiyalar

Laboratoriya mashg'ulotlarida talabalar amaliy mashg'ulotlarda tuzilgan dasturlarni kompyuter yordamida natijalarini ko'rib, ularni taxlil qiladilar va xulosalar chiqaradilar.

2.5. Mustaqil ta'lim topshiriqlari bo'yicha tavsiyalar

Darslik va o'quv qo'llanmalardan foydalanib, barcha mavzularni o'rganish. Tarqatma materiallar bo'yicha ma'ruza qismlarini o'zlashtirish

Talabalarning mustaqil ishlari har bir ma'ruza mavzusi asosida tashkil etiladi. Fanni o'rganish jarayonida mustaqil ishlarning bir necha turlaridan foydalaniladi:

- 1) adabiyotlar bilan ishlash;
- 2) ijodiy ish;
- 3) ishlarni elektron ko'rinishda bajarish;
- 4) ba'zi mavzular bo'yicha referatlar tayyorlash.

Mustaqil ta'limlarni tashkil etishda internet va axborot manbalaridan doimiy foydalaniladi.

Mustaqil ta'lim va mustaqil ishlar

1. Borland C++ Builder Standard komponentlar palitrasi komponentlari bilan ishlash
2. Borland C++ Builder Additional komponentlar palitrasi
3. Borland C++ Builderda Win32 komponentlar palitrasidan foydalanish
4. Borland C++ Builderda grafik axborotlar bilan ishlovchi komponentlar
5. Borland C++ Builderda Image va PaintBox komponentlarida foydalanish
6. Borland C++ Builderda Chart va VtChart komponentlarida foydalanish
7. Borland C++ Builderda Animation va MediaPlayer komponentlarida foydalanish
8. Borland C++ Builderda Win32 komponentlar palitrasidan foydalanish
9. Borland C++ Builderda System komponentlar palitrasidan foydalanish
10. Borland C++ Builderda ColorDialog va ColorBox komponentlari
11. Borland C++ Builderda PrintDialog va PrintSetupDialog komponentlari

12. Borland C++ Builderda Data Access komponentlar palitrasi
13. Borland C++ Builderda xossa va metodlar
14. Borland C++ Builderda massivlarni saralash
15. Borland C++ Builderning Excel dasturi bilan hamkorligi

Mustaqil o'zlashtiriladigan mavzular bo'yicha talabalar tomonidan referatlar tayyorlash va uni taqdimot qilish tavsiya etiladi.

2.6. Fanni o'qitish jarayonini tashkil etish va o'tkazish bo'yicha tavsiyalar

“Dasturlash tillari” fanini o'rganish davomida mashg'ulotlar paytida axborot (taqdimot, multimedia texnologiyalari) va ta'limning zamonaviy texnologiyalari (rivojlantiruvchi ta'lim texnologiyalari, fanni to'liq o'zlashtirishga yo'naltirilgan texnologiyalar, shaxsga yo'naltirilgan ta'lim texnologiyalari) hamda interfaol metodlar (“Aqliy hujum”, “BBB”, “Venn diagrammasi”, “T-chizma”, “Insert”, “Bir-biridan so'rash”, “FSMU”, “Bumerang”, “Klaster”) qo'llaniladi. Bundan tashqari darsliklar, o'quv qo'llanmalari, ma'lumotnomalar, pedagogik entsiklopediyalar va lug'atlar, ma'ruza matnlari, tarqatma materiallaridan foydalaniladi.

Ma'ruza darslarida zamonaviy kompyuter texnologiyalari yordamida prezentatsion va elektron-didaktik texnologiyalaridan, amaliy mashg'ulotlarda zamonviy pedagogik va innovatsion texnologiyalaridan, laboratoriya mashg'ulotlarida zamonaviy kompyuter sinflaridan foydalanish ko'zda tutilgan. Shuningdek buguni kun talabiga javob beradigan dasturlash tillaridan Paskal, Delphi, C++ dasturlash tillarini o'rnatuvchi disk ham bo'lishi lozim.

2.7. Fanni baholash tizimi:

2.7.1. Talabalar bilimni baholash mezonlari

Talabalarining bilimi quyidagi mezonlar asosida:

- ❖ talaba mustaqil xulosa va qaror qabul qiladi, ijodiy fikrlay oladi, mustaqil mushohada yuritadi, olgan bilimni amalda qollay oladi, fanning (mavzuning) mohiyatini tushunadi, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) bo'yicha tasavvurga ega deb topilganda — 5 (alo) baho;
- ❖ talaba mustaqil mushohada yuritadi, olgan bilimni amalda qollay oladi, fanning (mavzuning) mohiyatni tushunadi, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) boyicha tasavvurga ega deb topilganda — 4 (yaxshi) baho;
- ❖ talaba olgan bilimni amalda qollay oladi, fanning (mavzuning) mohiyatni tushunadi, biladi, ifodalay oladi, aytib beradi hamda fan (mavzu) bo'yicha tasavvurga ega deb topilganda — 3 (qoniqarli) baho;
- ❖ talaba fan dasturini o'zlashtirmagan, fanning (mavzuning) mohiyatini tushunmaydi hamda fan (mavzu) boyicha tasavvurga ega emas deb topilganda — 2 (qoniqarsiz) baho bilan baholanadi.

Talabaning “Dasturlash tillari” fani bo'yicha bilim, ko'nikma va malakalarini baholashda quyidagi mezonlarga asoslaniladi:

a) **5 (a'lo) baho** uchun talabaning bilim darajasi quyidagilarga javob berishi lozim:

kasbiy sohasida uchraydigan turli hil masalalarga algoritmlar tuza olishi, algoritmnining turlarni farqlay olish, tasvirlash usullariga oid misollar keltira olish, rekursiya va iteratsiya, algoritmnining murakkabligi tushunchalarni ajrat olishi, samarali algoritmlar ishlab chiqishning asosiy usullari(balansirovka, dinamik dasturlash va boshqalar)ni amaliy qo'llay olishi, biror bir dasturlash tillari va ularning turlarini farqlay olish, dasturlash tillalari yordamida amaliy masallalarga dasturlar tuza olish, massivlar, grafik operatorlar, satriy kattaliklar bilan ishlash, funktsiyalar va protseduralar, yozuvlar, ro'yxatlar, fayllar, modulli dasturlar haqidagi bilimlarni amalda qo'llay olish, ob'ektga yo'naltirilgan dasturlash tillaridan foydalana olish, boshqarish elementlari, oynalar, dialoglar;

voqealar va habarlar, ob'ektga yo'naltirilgan muhitlarda habarlarni uzatish va ularga ishlov berish, ob'ektlar ierarxiyasi asosida dasturlarni loyihalash haqidagi nazariy bilimlarga ega bo'lishi, ushbu nazariy bilimlarni amalda qo'llay olishi, kasbiy sohalarida fanning amaliy imkoniyatlaridan foydalana olishi, mustaqil ishlash ko'nikmalariga ega bo'lishi;

b) **4 (yaxshi) baho** uchun talabani bilim darajasi quyidagilarga javob berishi lozim:

turli hil masalalarga algoritmlar tuza olishi, algoritmning turlarni farqlay olish, tasvirlash usullariga oid misollar keltira olish, rekursiya va iteratsiya, algoritmning murakkabligi tushunchalarni ajrat olishi, biror bir dasturlash tillari va ularning turlarini farqlay olish, dasturlash tillalari yordamida amaliy masallalarga dasturlar tuza olish, massivlar, grafik operatorlar, satriy kattaliklar bilan ishlash, funktsiyalar va protseduralar, haqidagi bilimlarni amalda qo'llay olish, ob'ektga yo'naltirilgan dasturlash tillaridan foydalana olish, boshqarish elementlari, oynalar, ob'ektlar ierarxiyasi asosida dasturlarni loyihalash haqidagi nazariy bilimlarga ega bo'lishi; ushbu nazariy bilimlarni amalda qo'llay olishi; kasbiy sohalarida fanning amaliy imkoniyatlaridan foydalana olishi;

v) **3 (qoniqarli) baho** uchun talabani bilim darajasi quyidagilarga javob berishi lozim:

turli hil masalalarga algoritmlar tuza olishi, algoritmning turlarni farqlay olish, tasvirlash usullariga oid misollar keltira olish, rekursiya va iteratsiya, algoritmning murakkabligi tushunchalarni ajrat olishi, biror bir dasturlash tillari va ularning turlarini farqlay olish, dasturlash tillalari yordamida amaliy masallalarga dasturlar tuza olish haqidagi qisman tassavurga ega bo'lishi; amaliyotda ayrim dasturlarni ko'llay olishi;

g) fanning nazariy qismini tushunmaydigan, amaliy qo'llash imkoniyatlari juda past, dasturlarni mutaql ravishda ishlata olmaydigan talabalarga **2 (qoniqarsiz) baho** qo'yiladi.

2.7.2. Talabani amaliy va laboratoriya mashg'ulotlarni o'zlashtirish darajasi quyidagi mezon asosida aniqlanadi.

Baholash mezonlari	5 baholik shkala	100 ballik shkala
Yetarli nazariy bilimga ega. Topshiriqlarni mustaqil yechgan. Berilgan savollarga toliq javob beradi. Masalani mohiyatiga toliq tushunadi. Auditoriyada faol. Oquv tartib intizomiga toliq rioya qiladi. Topshiriqlarni namunali rasmiylashtirgan.	«5»	“Alo” 90 - 100%
Yetarli nazariy bilimga ega. Topshiriqlarni yechgan. Berilgan savollarga yetarli javob beradi. Masalani mohiyatini tushunadi. Oquv tartib intizomiga toliq rioya qiladi.	«4»	“Yaxshi” 70 - 89,9%
Topshiriqlarni yechishga harakat qiladi. Berilgan savollarga javob berishga harakat qiladi. Masalani mohiyatini chala tushungan. Oquv tartib intizomiga rioya qiladi.	«3»	“Qoniqarli” 60 - 69,9%
Talaba amaliy mashg'ulot darsi mavzusiga nazariy tayyorlanib kelmasa, mavzu boyicha masala, misol va savollariga javob bera olmasa, darsga sust qatnashsa bilim darajasi qoniqarsiz baholanadi	«2»	“Qoniqarsiz” 0 - 59,9%

ON ni baholash

Oraliq nazorat “Dasturlash tillari” fanining bir necha mavzularini qamrab olgan bo'limi bo'yicha yozma yoki test ravishda amalga oshiriladi. Bundan maqsad talabalarning tegishli savollarni bilishi yoki muammolarni echish konikmalari va bilim malakalari aniqlanadi.

Oraliq nazorat turini otkazish va mazkur nazorat turi boyicha talabani bilimni baholash tegishli fan boyicha oquv mashg'ulotlarini olib borgan professor-oqituvchi tomonidan amalga oshiriladi.

Oraliq nazorat turini topshirmagan, shuningdek ushbu nazorat turi boyicha «2» (qoniqarsiz) baho bilan baholangan talaba yakuniy nazorat turiga kiritilmaydi.

Talaba nazorat turi otkazilgan vaqtda uzrli sabablarsiz qatnashmagan hollarda jurnalga «0» belgisi yozib qoyiladi.

Oraliq nazorat turi har bir fan boyicha fanning xususiyatidan kelib chiqqan holda 2 martagacha otkazilishi mumkin. Oraliq nazorat turini otkazish shakli va muddati fanning xususiyati va fanga ajratilgan soatlardan kelib chiqib tegishli kafedra tomonidan belgilanadi.

Talabani oraliq nazorat turi boyicha baholashda, uning o'quv mashg'ulotlari davomida olgan baholari inobatga olinadi.

O'quv yilining kuzgi hamda bahorgi semestrda 2 ta yozma ish va 2 ta mustaqil ish rejalashtirilgan bo'lib, yozma ish 5 baholik shkalada baholanadi.

Oraliq baholash (OB) "Dasturlash tillari" fanining bir necha mavzularini qamrab olgan bolimi boyicha, tegishli nazariy, amaliy va laboratoriya mashg'ulotlari o'tib bo'lingandan so'ng yozma ish shaklida amalga oshiriladi. Bundan maqsad talabalarning tegishli savollarni bilishi yoki muammolarni yechish konikmalari va malakalari aniqlanadi. O'quv yilining **1-semestrda** 2 ta yozma ish va 10 ta mustaqil ish rejalashtirilgan bo'lib, yozma ishga 5 ball, mustaqil ishga ham 5 ball ajratilgan. OB nazorat ishlari yozma ish shaklida o'tkazilishi nazarda tutilgan, yozma ish savollari ishchi o'quv dastur asosida tayyorlanadi. OB da "2" baho olgan talaba o'zlashtirmagan hisoblanadi. OB ni o'zlashtirmagan talabalarga qayta topshirish imkoniyati beriladi.

YaN ni baholash

Yakuniy nazorat "Dasturlash tillari" fanining barcha mavzularini qamrab olgan bo'lib, nazariy, amaliy mashg'ulotlar o'tib bo'lingandan so'ng test yoki yozma ravishda amalga oshiriladi. Bundan maqsad talabalarning fan bo'yicha o'zlashtirish ko'rsatkichlari, yani bilim darajasi yoki muammolarni echish konikmalari va malakalari aniqlanadi.

Yakuniy nazorat ishlari test usulida ham o'tkazilishi nazarda tutilgan, test savollari ishchi o'quv dasturi asosida tayyorlanadi.

Yakuniy nazorat turini o'tkazish va mazkur nazorat turi bo'yicha talabaning bilimini baholash o'quv mashg'ulotlarini olib bormagan professor-o'qituvchi tomonidan amalga oshiriladi.

Talaba tegishli fan bo'yicha yakuniy nazorat turi o'tkaziladigan muddatga qadar oraliq nazorat turini topshirgan bo'lishlari shart.

Yakuniy nazorat turiga kirmagan yoki kiritilmagan, shuningdek ushbu nazorat turi bo'yicha «2» (qoniqarsiz) baho bilan baholangan talaba akademik qarzdor hisoblanadi.

Yakuniy nazorat turi bo'yicha talabaning bilimi «2» (qoniqarsiz) baho bilan baholangan yoki Jurnalga «0» belgisi yozib qo'yilgan hollarda ushbu baho yoki belgi talabaning Baholash daftariga yozilmaydi.

Yakuniy baholash (YaB) "Dasturlash tillari o'tib bo'lingandan so'ng test yoki yozma ish shaklida amalga oshiriladi. Bundan maqsad talabalarning fan bo'yicha o'zlashtirish ko'rsatkichlari, yani bilim darajasi yoki muammolarni yechish konikmalari va malakalari aniqlanadi. YaB nazorat ishlari test yoki yozma ish usulida ham otkazilishi nazarda tutilgan, test va yozma ish savollari ishchi o'quv dasturi asosida tayyorlanadi. OB ga ajratilgan balldan "2" va undan past ball to'plagan talaba o'zlashtirmagan hisoblanadi va YaB ga kiritilmaydi. YaB ni ozlashtirmagan talabalarga qayta topshirish imkoniyati beriladi. YaB bo'yicha olinadigan test yoki yozma ish variantlari kafedra mudiri rahbarligida tuziladi va dekanatga topshiriladi.

Test usulida YaN ni baholash mezonlari:

YaB test yoki yozma ish shaklida otkaziladi. YaB test shaklida otkazilsa talabalarga variantlar asosida 30 ta test savoli beriladi. Har bir tog'ri javob quyidagicha balldan baholanadi. To'g'ri javoblar soniga qarab talabaning YaB da to'plagan ballari aniqlanadi.

"5" – 27-30 ta

"4" – 21-26 ta

"3" – 18-20 ta

"2" – 0-16 ta

VII. Asosiy va qo‘shimcha o‘quv adabiyotlar hamda axborot manbalari

Asosiy adabiyotlar:

1. Peter Gottschling. Discovering Modern C++. An Intensive Course for Scientists, Engineers, and Programmers. “Addison-Wesley”, 2015 y.
2. A. R. Azamatov, B. Boltayev. Algoritmash va dasturlash asoslari. O‘quv qo‘llanma. T. : “Cho‘lpon”, 2010 y.
3. A. R. Azamatov, B. Boltayev. Algoritmash va dasturlash asoslari. O‘quv qo‘llanma. T. : “Cho‘lpon”, 2013 y.
4. Sh. I. Razzoqov, M. J. Yunusova. Dasturlash: Kasb-hunar kollejlari uchun o‘quv qo‘llanma. T. : “Ilm Ziyo”, 2011y.
5. М. Ашуров, М. Мирмахмудов, Ш. Сапаев. Замоновий дастурлаш тиллари фанидан лаборатория ишлари. Т. : ТДПУ, 2008 й.
6. Меняев Михаил Федорвич. Информационные технология управления. Москва, «Издательский ОмегаЛ», 2003 г.

Qo‘shimcha adabiyotlar

7. Мирзиёев Шавкат Миромонович. Эркин ва фаровон, демократик Ўзбекистон давлатини биргаликда барпо этамиз. Ўзбекистон Республикаси Президенти лавозимида киришиш тантанали маросимида бағишланган Олий Мажлис палаталарининг қўшма мажлисидаги нутқ / Ш.М. Мирзиёев. – Тошкент : Ўзбекистон, 2016. - 56 б.
8. Мирзиёев Шавкат Миромонович. Танқидий таҳлил, қатъий тартиб-интизом ва шахсий жавобгарлик – ҳар бир раҳбар фаолиятининг кундалик қондаси бўлиши керак. Мамлакатимизни 2016 йилда ижтимоий-иқтисодий ривожлантиришнинг асосий якунлари ва 2017 йилга мўлжалланган иқтисодий дастурнинг энг муҳим устувор йўналишларига бағишланган Вазирлар Маҳкамасининг кенгайтирилган мажлисидаги маъруза, 2017 йил 14 январ / Ш.М. Мирзиёев. – Тошкент : Ўзбекистон, 2017. – 104 б.
9. Мирзиёев Шавкат Миромонович. Қонун устуворлиги ва инсон манфаатларини таъминлаш – юрт тараққиёти ва халқ фаровонлигининг гарови. Ўзбекистон Республикаси Конституцияси қабул қилинганининг 24 йиллигига бағишланган тантанали маросимдаги маъруза. 2016 йил 7 декабр /Ш.М.Мирзиёев. – Тошкент: “Ўзбекистон”, 2017. – 48 б.
10. Мирзиёев Шавкат Миромонович. Буюк келажагимизни мард ва олижаноб халқимиз билан бирга қурамиз. Мазкур китобдан Ўзбекистон Республикаси Президенти Шавкат Мирзиёевнинг 2016 йил 1 ноябрдан 24 ноябрга қадар Қорақалпоғистон Республикаси, вилоятлар ва Тошкент шаҳри сайловчилари вакиллари билан ўтказилган сайловолди учрашувларида сўзлаган нутқлари ўрин олган. /Ш.М.Мирзиёев. – Тошкент: : “Ўзбекистон”, 2017. – 488 б
11. Ўзбекистон Республикаси Президентининг Фармони. Ўзбекистон республикасини янада ривожлантириш бўйича ҳаракатлар стратегияси тўғрисида. (Ўзбекистон Республикаси қонун ҳужжатлари тўплами, 2017 й., 6-сон, 70-модда)
12. Ўзбекистон Республикаси Конституцияси. Т.: Ўзбекистон. 2014. -46 б.
- Nazirov Sh.A., Qobulov R.V., Bobojanov M.R., Raxmanov Q.S. C va C++ tili. “Vorish-nashriyot” MCHJ, Toshkent 2013. 488 b.
13. Безручко В.Т.. Практикум по курсу информатики. М. : «Финансы и статистика», 2004 г.
14. Дарахвелидзе П., Марков Э.. Программирование в Delphi7. Учебник. Санкт-Петербург, “БХВ-Петербург” 2003 г.

15. Пестиков В. М., Маслобоев А. Н.. Turbo PASCAL 7. 0. Изучаем на примерах. Санкт-Петербург. : “БХВ-Петербург”, 2004 г.
16. Фаронов В. В. Программирование на языке высокого уровня Delphi. Учебник. М. : “Питер”, 2003 г.

Internet saytlari

- 17. www.ziyonet.uz** – Axborot ta’lim portali
- 18. www.edu.uz** – Oliy va o’rta maxsus ta’lim vazirligi portali
- 19. www.tdpu.uz** – Nizomiy nomidagi TDPU rasmiy sayti
- 20. <http://acm.tuit.uz/>** - dasturiy yechim to’g’riligini avtomatik testlovchi tizim.
- 21. <http://acm.tuit.uz/forum/>, <http://acm.timus.ru/>** – dasturlarni testlovchi tizim.

Testlar					
№	Test savollari	To'g'ri javob	Muqobil javob	Muqobil javob	Muqobil javob
1.	C++ tili qachon tuzildi?	* 1980 y	1999 y	1986 y	1991 y
2.	C++ tili kim tomonidan tuzilgan?	Jon Fon Neyman	Bil Gets	Bjarne Stroustrup	To'g'ri javob yo'q
3.	C++ tili qaysi tilga asoslangan?	Paskal	*C	Java	Delphi
4.	C++ alfavitiga qanday simvollar kiradi?	Katta va kichik lotin alfaviti xarflari (A,B,...,Z,a,b,...,z)	Raqamlar: 0,1,2,3,4,5,6,7,8,9	Maxsus simvollar: “ , { } [] () + - / % \ ; ‘ . : ? < = > _ ! & * # ~ ^	Hamma javoblar to'g'ri
5. lotin xarflari,ostki chiziq belgisi va sonlar ketma ketligidan iborat bo'ladi?	*Identifikator	Simvol	Raqamlar	Harflar
6.	Tilda ishlatiluvchi ya'ni dasturchi tomonidan uzgaruvchilar nomlari sifatida ishlatish mumkin bulmagan identifikatorlar deyiladi.	* Xizmatchi so'zlar	Raqamlar	Harflar	Simvollar
7.	C++ dagi xizmatchi so'zlarni toping?	* Hammasi to'g'ri	Goto	While	Else
8.	VARIABLES – bu	*o'zgaruvchilar	O'zgarmaslar	Simvollar	Raqamlar
9.	O'zgaruvchilarning qanday tiplari mavjud?	*hammasi	Char	Float	int
10.	Butun sonlar qaysi toifaga kiradi?	*Int	Char	Float	Hammasi
11.	Long char – bu ...	*Uzun simvol	Butun son	Bitta simvol	Haqiqiy son
12.	CONSTANTS - bu ...	*O'zgarmaslar	O'zgaruvchilar	Simvollar	Harflar
13.	C++ tilida necha turdagi konstantalar ishlatilishi mumkin?	*5 ta	8 ta	3 ta	7 ta
14.	Qanday konstantalar C++ tili konstantalariga kirmaydi?	*Satrli	Char	Float	Int
15.	Qanday konstanta bu ikkilik qavslarga olingan ixtiyoriy simvollar ketma ketligidir	*Satrli	Int	Float	Char
16.	C++da arifmetik operator qaysilar	+, -	/, *, %	To'g'ri javob yo'q	*a va b to'g'ri
17.	C++ da solishtirish amallari qatorini toping?	*hammasi	<, >	<=, >=	==, !=

18.	C++da kiritish operatori qaysi?	*cin	cout	in	Case
19.	C++da chiqarish operatori qaysi?	*cout	case	then	cin
20.	Char toifaning hajmi qancha	*8	13	5	9
21.	Int toifaning hajmi qancha	*16	9	8	24
22.	Int toifaning qiymatlar chegarasi qancha?	*_ 32768...32767	0..255	-128..127	0..65535
23.	Unsignet Int toifaning qiymatlar chegarasi qancha?	*0..65535	0..255	-128..127	- 32768...32767
24.	...- ko'rsatkich yagona arifmetik bulmagan konstantadir.	* NULL	ted	prt	Key
25.	int a, b, c; cout << "a="; cin >> a; cout << "b="; cin >> b; c = a + b; cout << c << endl; return 0; bu dastur nimani hisoblaydi?	*Ikki sonning yig'indisi	Ikki sonning bolinmasi	Ikki sonning ayirmasi	Ikki sonning kopaytmasi
26.	c++da matematik funksiyalardan foydalanganda qaysi fayllardan foydalaniladi?	*math.h	iostream.h	include	string.h
27.	C++da satrli ifodalardan foydalanganda qaysi fayllardan foydalaniladi?	*string.h	iostream.h	include	math.h
28.	C++da sqrt qaysi ifoda turiga kiradi?	*matematik	mantiqiy	ilmiy	tog'ri javob yo`q
29.	sqrt funksiyasi nimani ifodalaydi?	*ildiz	Kvadrat	ko'paytma	bo'linma
30.	power funksiyasi nimani ifodalaydi?	*daraja	ildiz	ko'paytma	Kvadrat
31.	l=sqrt(k) bunda l qaysi toifaga mansub bo'ladi?	*float	char	string	Int
32.	float a; cout << "a="; cin >> a; a = sqrt(a); cout << a << endl; return 0; a ning natijasini top?	*a sonning kvadrat ildizi	a sonning kopaytmasi	a sonning ildizi	tog'ri javob yoq
33.	Butun sonlar toifasini toping?	*int	float	string	Char
34.	Haqiqiy sonlar toifasini toping?	*float	int	string	Char
35.	Belgilar toifasini toping?	* char	float	string	int
36.	Satrli toifani toping?	* string	float	int	Char
37.	abs() funksiya nimani ifodalaydi?	*modul	funksiya	kvadrat	Ildiz
38.	cos funksiya nimani ifodalaydi?	*sonning cosinusi	modul	kvadrat	Ildiz
39.	exp funksiya nimani ifodalaydi?	*ex	ex	log	Lg

40.	ceil(x) funksiya nimani ifodalaydi?	* x ni x dan katta yoki unga teng b-n eng kichik butun songacha yahlitlaydi	x ni x dan kichik bo'lgan eng katta butun songacha yahlitlaydi	x/y ning qoldig'ini kasr son tipida beradi	x ning absolut qiymati
41.	floor(x) funksiya nimani ifodalaydi?	* x ni x dan kichik bo'lgan eng katta butun songacha yahlitlaydi	x ni x dan katta yoki unga teng b-n eng kichik butun songacha yahlitlaydi	x/y ning qoldig'ini kasr son tipida beradi	x ning absolut qiymati
42.	fmod(x,y) funksiya nimani ifodalaydi?	*x/y ning qoldig'ini kasr son tipida beradi	x ni x dan kichik bo'lgan eng katta butun songacha yahlitlaydi	x ni x dan katta yoki unga teng b-n eng kichik butun songacha yahlitlaydi	x ning absolut qiymati
43.	If (ifoda) 1- operator Else 2- operator bu qanday operator?	*shartli	Shartsiz	to'g'ri javob yoq	Takrorlanuvchi
44.	int a; cin >> a; if (a % 2 == 0) cout << "juft"; else cout << "toq"; return 0; dastur nimani aniqlaydi	*sonning juft yoki toqligi	Sonning juftligi	Sonning toqligi	To'g'ri javob yo`q
45.	int a, b, max; cout << "a="; cin >> a; cout << "b="; cin >> b; max = (a > b) ? a : b; cout << max << endl; return 0; nimani aniqlaydi?	*sonning maksimali	sonning minimali	sonning toqligi	sonning juftligi
46.	int main() {for (int i = 0; i < 5; i++){cout << i << endl;} return (0); yachimni toping?	*1..5 gacha sonlar	sonning minimali	sonning maksimali	sonning juftligi
47.	for (int i = 0; i < 10 ; i++) cout << "Hello!"<< endl; nima chop etiladi?	* hello so'zi 10 marta	hello so'zi 11 marta	hello so'zi 12 marta	hello so'zi 9 marta
48. – funksiyasini har qanday sikl operatoriga qo'llash mumkin. Bu funksiya sikl tugatilishini ta'minlaydi. Ya'ni boshqarilishni sikl operatoridan keyingi operatorga uzatadi.	* break	while	for	Repeat
49.- funksiyasini har qanday sikl operatoriga qo'llash mumkin. Bu funksiya sikl parametrining keyingi	* continue	for	repeat	While

	qiymatni qabul qilishini taminlaydi. Boshqacha soʻz bilan aytganda sikl tanasi tugatiladi. Bunda siklning oʻzi tugatilmaydi.				
50.	Switch Case qanday operator?	* Kalit boʻyicha tanlash	Shartsiz	Modul	Takrorlanish
51. sikllarni tashkil qilishning eng umumiy (ommaviy) usulidir.	*For	Case	Then	While
52.	Belgilar satrini teskari tartibda yozish uchun qanday operator ishlatiladi?	*Top va bot	For va while	ceil	Break va case
53.	...operatori birlashgan switch, do, for, while sikllardan eng ichkisining bajarilishi tugallanilishini taʼminlaydi.	*break	while	continue	For
54.	C++ Builderda Ctrl+R tugmalari birgalikda bosilsa qanday oyna hosil boʻladi?	*Izlash va almashtirish	Izlash darchasini ochish	Joriy faylni saqlash	Shablonlar roʻyxatini ochish
55.	Komponentalarning qaysi xususiyati orqali matnlarni oʻngga, chapga, oʻrtaga tekislash mumkin?	* Alignment	Align	Actions	Caption
56.	Satrlı sonni butun songa oʻtkazish qanday amalga oshiriladi?	* StrToInt	IntToStr	StrToFloat	FloatToStr
57.	Satrlı sonni haqiqiy songa oʻtkazish qanday amalga oshiriladi?	*StrToFloat	FloatToStr	StrToInt	IntToStr
58.	Taymerdan foydalanishni oʻzgartirish ...	*T = !T;	Button2->Caption = "Pusk"	if(!T)	Button2->Caption = "Pauza"
59.	Borland C++ Builder dasturining asosiy oynalari nechta?	*5	4	3	6
60.	C++ Builder dasturining Align To Grid buyrugʻi menyular satrining qaysi boʻlimida joylashgan?	*Edit	Run	Search	File
61.	Memo komponentasi qaysi komponentalar palitrasiga kiradi?	* Standart	Win32	Additional	System
62.	Timer komponentasi qaysi komponentalar palitrasiga kiradi?	*Win32	Standart	Additional	System
63.	Object Inspector oynasining boʻlimlari nechta?	*2	1	3	4

64.	C++ Builderda nechta forma qo'llanilganini ko'rish uchun klaviaturadan qaysi tugma orqali bajarish mumkin?	*F12	F8	F10	F5
65.	LabeledEdit komponentasi qaysi komponentalar palitrasiga kiradi?	*Additional	Win32	Standart	System
66.	C++ Builderda Object Inspector qaysi tugma orqali chiqadi?	*F11	F5	F9	F3
67.	Label komponentasi vazifasi nima?	*sitr joylashtirish	kiritish qatori	tugma	ro'yxatdan tanlash
68.	Edit komponentasi vazifasi nima?	*kiritish qatori	tugma	ro'yxatdan tanlash	sitr joylashtirish
69.	Button komponentasi vazifasi nima?	*tugma hosil qilish	ro'yxatdan tanlash	sitr joylashtirish	ranglar jadvali
70.	CheckBox komponentasi vazifasi nima?	bayroqchali belgilash	ro'yxatdan tanlash	sitr joylashtirish	ranglar jadvali
71.	Button bilan buttoning farqi nimada?	*dizaynda	amal bajarishda	rangida	tilida
72.	Timer komponentasi vazifasi nima?	*vaqt bilan ishlash	bayroqchali belgilash	ro'yxatdan tanlash	sitr joylashtirish
73.	MediaPlayer komponentasi qaysi komponentalar palitrasiga kiradi?	*System	Standart	Win32	Additional
74.	ColorGrid komponentasi vazifasi nima?	ranglar jadvali	bayroqchali belgilash	ro'yxatdan tanlash	sitr joylashtirish
75.	CCalendar komponentasi qaysi komponentalar palitrasiga kiradi?	*Samples	Additional	Win32	System
76.	Kontekstli menyu yaratishda asosan qaysi komponentadan foydalaniladi?	*popupmenu	ColorGrid	Timer	Speedbutton
77.	Formaning ixtiyoriy joyiga grafik tasvirni joylashtirish imkonini beradigan obyekt	*Image	Brush	Shape	Assign
78.	Tols-?	*servis xizmatidan foydalanish	yordam chaqirish.	formani ishga tushirish	kiritish qatori
79.	Sarlavha matnini ... aniqlaydi.	*Caption	Uses	Top va Lift	Width va Height
80.	Forma balandligini o'zgartiring.	*Width	Caption	Name	Brush
81.	Delphi loyiha fayl kengaytmasi	*dpr	pas	res	jpg
82.	Delete funksiyaning qiymati qaysi tipga tegishli.	*String	Intejer	Real	Char
83.	Fiksirlangan ustun rangi qanday nomlanadi?	* Fixed cols	Fixed Color	Key option	Strings

84.	Fayl o'zgaruvchisi bilan asosiy fayl orasida aloqa o'rnatadigan funksiya.	*Assign	Label	Close	Type
85.	Dinamik bog'lanuvchi bibliotekalar so'zining qisqartmasi.	*DLL	GLL	LLL	BLL
86.	Ma'lum shartlarga muvofiq bajariladigan algoritm qanday.	*tarmoqlanuvchi	takrorlanuvchi	chiziqli	barchasi to'g'ri
87.	Siljitish yo'lchasiga ega panelni ko'rsating.	*SpeedButton	BitBtn	ScrollBar	Memo
88.	ListBox komponentasi qaysi palitrasida joylashgan	*Standart	Additional	Win32	DataAccess
89.	DrawGrid komponentasi qaysi palitrasida joylashgan	*Additional	Standart	Win32	DataAccess
90.	SpinEdit komponentasi qaysi palitrasida joylashgan	*Samples	Additional	Win32	Standart
91.	Belgi sifatida qanday sonlardan foydalaniladi?	* to'rtta raqamdan oshmaydigan ishorasiz butun son va lotin harflar	haqiqiy sonlar	to'rt xonali butun sonlar	hamma javob to'g'ri
92.	O'zgaruvchilarni tasvirlash bo'limi qanday kalit so'z bilan boshlanadi	*VAR	CONST	TYPE	CASE
93.	Ma'lumotlarning barcha turini nechtaga ajratish mumkin?	*5 ta	3 ta	4 ta	2 ta
94.	Ko'rsatilganlarning qaysi biri standart turga tegishli	* INTEGER, REAL, BOOLEAN, CHAR, STRING	PROGRAM, BEGIN, END	LABEL, CONST, VAR, TYPE	INPUT, OUTPUT, WRITE
95.	Massivlar, yozuvlar, to'plamlar va fayllar ma'lumotlarining qanday turi hisoblanadi?	*Murakkab turlar	standart turlar	oddiy turlar	hamma javob to'g'ri
96.	Butun turdagi o'zgarma deb nimaga aytiladi?	* Nuqtasiz yozilgan ixtiyoriy o'nli son	Ixtiyoriy manfiy son	Nuqtasiz yozilgan ixtiyoriy musbat son	Ixtiyoriy musbat son
97.	Butun turdagi ma'lumotlar ustida qanday amallar bajarilganda butun natijalar olish mumkin?	*+, -, *, DIV, MOD	+, -, *, /	*, /, darajaga ko'tarish	+, -, kvadrat ildiz olish
98.	Paskal tilida haqiqiy o'zgarma necha xil ko'rinishda tasvirlanadi	*2 xil	3 xil	1 xil	5 xil
99.	Quyidagi javoblarning qaysi birida qo'zg'almas konstanta ko'rsatilgan?	*45.6254	-0.2865E-5	5.28E6	-2.65E5