

O'ZBEKISTON RESPUBLIKASI XALQ TA'LIMI

VAZIRLIGI

NAVOIY DAVLAT PEDAGOGIKA INSTITUTI

FIZIKA-MATEMATIKA FAKULTETI

«Informatika va axborot texnologiyalari» kafedrası

KURS ISHI

MAVZU: C++ Builder dasturida chiziqli jarayonlarga doir masalalar yechish

Qabul qildi: Xodjabayev F.D.

Bajardi: 3 kurs talabasi Parmonov L.

Navoiy-2016

MUNDARIJA

KIRISH.....	3
1. C++ dasturlash tilining imkoniyatlari va unda ishlash asoslari	5
1.1. C++ dasturlash tilining yangi imkoniyatlari	5
1.2. Komponentlar (tarkibiy qismlar).....	5
1.3. Komponentli sinflarni e’lon qilish.....	6
1.4. O‘zgaruvchan e’lonlar.....	7
1.5. C++Builder dasturining asoslari	9
1.6. Vizual loyihalash.....	12
2. C++ Builder dasturida chiziqli jarayonlarni dasturlash.....	13
2.1. Y sonini X-darajali ildizini hisoblash dasturi.....	13
2.2. Ixtiyoriy uchburchaklarning yuzasini hisoblash dasturi.....	14
2.3. Ikkinchi tartibli determinantlarni hisoblash.....	15
2.4. Chiziqli tenglamalar sistemasini Kramer usulida yechish.....	18
XULOSA.....	22
FOYDALANILGAN ADABIYOTLAR.....	23

K I R I S H

Informatika fani tabiat va jamiyatda kechayotgan jarayonlarni o'rganish va tahlil etishda asosiy vositalardan biri sifatida e'tirof etiladi. Ushbu vositalarning imkoniyatlaridan samarali va tez suratlar bilan foydalanishni kompyuter texnologiyalarining zamonaviy yutuqlarisiz tasavvur etib bo'lmaydi.

Kompyuter – murakkab elektron qurulma, mufassal o'rganish uchun ancha vaqt zarur bo'ladi. Biroq, kompyuterdan foydalanuvchilar – dasturchilar uchun uning ishi haqida eng umumiy ma'lumotlarga ega bo'lish yetarlidir. Dasturchini, asosan, mashina dastur buyruqlarini qanday qilib amalga oshirishi, bunda u qanday amallarni bajarishi qiziqtiradi. Kompyuter dasturni bajara borib, turli ma'lumotlar (sonlar, mantiqiy qiymatlar, matnlar va h.k) ustida amallar bajaradi. Ishlash usuliga ko'ra, translyatorlar kompilyatorlarga va interpretatorlarga bo'linadi. Kompilyatorning interpretatorlardan farqi shuki, kompilyator foydalanuvchi yozgan dasturni EHM uchun tushunarli ko'rinishga o'tkazadi (u ichki ko'rinish deyiladi), so'ngra bu ko'rinishdagi dastur bajariladi. Interpretator esa har bir ko'rsatmani ichki ko'rinishga o'tkazib bajaradi.

Foydalanuvchidan kompyuter bilan muloqot qilish uchun kompyuter tilini bilish ham talab qilinadi. Kompyuter tushunadigan til – bu dasturlash tili deb ataladi. Biror masalani kompyuterda yechish uchun, avvalo, uning algoritmi tuzilishi va bu algoritmnini kompyuter tushunadigan ko'rsatmalar va qonun-qoidalar asosida yozilishi kerak bo'ladi. Bu yozuv dastur bajarishi mumkin bo'lgan ko'rsatmalarning izchil tartibidan iboratdir. Kompyuter uchun dastur tuzish jarayoni dasturlash va dasturni tuzadigan kishi dasturchi deb ataladi.

Hozirgi paytda dastrulash tillarining soni juda ko'payib ketmoqda. Lekin, shuni aytish kerakki, har qanday dasturlash tili o'zining darajasi va qo'llash sohasiga ega. Ba'zi bir tillar bir necha xil soha masalalarini yechishda ishlatiladi. Bunday tillar universal tillar deb ham ataladi.

C++ Builder tili ham universal til hisoblanadi. U dasturlashning asosiy tushunchalari va konstruksiyalarini o'z ichiga olishi bilan birga boshqa universal dasturlash tillariga qaraganda ancha mukammal.

C++ Builder tilida dastur yaratish bir nechta bosqichlardan iborat bo'ladi. Dastlab, matn tahririda (odatda dasturlash muhitining tahririda) dastur matni teriladi, bu faylning kengaytmasi ".srr" bo'ladi, Keyingi bosqichda dastur matn yozilgan fayl kompilyatorga uzatiladi, agarda dasturda xatoliklar bo'lmasa, kompilyator ".obj" kengaytmali ob'ekt modul faylini hosil qiladi. Oxirgi qadamda komponovka (yig'uvchi) yordamida ".exe" kengaytmali bajariluvchi fayl - dastur hosil bo'ladi. Bosqichlarda yuzaga keluvchi fayllarning nomlari boshlang'ich matn fayl nomi bilan bir xil bo'ladi.

Kompilyatsiya jarayonining o'zi ham ikkita bosqichdan tashkil topadi. Boshda preprotssessor ishlaydi, u matndagi kompilyatsiya direktivalarini bajaradi, xususan #include direktivasi bo'yicha ko'rsatilgan kutubxonalardan C++ Builder tilida yozilgan modullarni dastur tarkibiga kiritadi. Shundan so'ng kengaytirilgan dastur matni kompilyatorga uzatiladi. Kompilyator o'zi ham dastur bo'lib, uning uchun kiruvchi ma'lumot bo'lib C++ Builder tilida yozilgan dastur matni hisoblanadi. Kompilyator dastur matnini leksem (atomar) elementlarga ajratadi va uni leksik, keyinchalik sintaksik tahlil qiladi. Leksik tahlil jarayonida u matni leksamalarga ajratish uchun "probel ajratuvchisini" ishlatadi. Probel ajratuvchisiga – probel belgisi (' '), '\t' – tabulyatsiya belgisi, '\n' – keyingi qatorga o'tish belgisi, boshqa ajratuvchilar va kommentariylar (izohlar) kiradi.

C++ Builder tilida izohlar ikki ko'rinishda yozilishi mumkin.

Birinchisi /* dan boshlanib va */ belgi bilan tugagan barcha belgilar ketma-ketligi hisoblanadi, ikkinchisi // belgilardan boshlangan va satr oxirigacha yozilgan belgilar ketma-ketligi bo'lib "satriy izoh" deb nomlanadi. Birinchi ko'rinishda yozilgan kommentariylar bir necha satrga yozilishi mumkin va undan keyin C++ Builder operatorlari davom etish mumkin.

1. C++ dasturlash tilining imkoniyatlari va unda ishlash asoslari

1.1. C++ dasturlash tilining yangi imkoniyatlari

C++Builder, nafaqat ANSI C++ standarti kiritayotgan yangiliklarni qoʻllab-quvvatlaydi, balki tilni yangi imkoniyatlar bilan boyitadi. Shuni tushunib olish muhimki, tilni kengaytirish hech qachon quruq maqsad boʻlib qolmagan, va siz xali-hamon standart C++ doirasida yozilgan mantlarni kompilyatsiya qila olasiz. Biroq ilovalarni tez ishlab chiqish texnologiyasi (RAD) uchun C++Builder taqdim etgan imtiyozlardan toʻliq foydalanish uchun, kiritilgan til kengaytirishlarni qabul qilishingizga toʻgʻri keladi.

Kengaytirishlarning ayrimlari (maslan, `_classid`) ni C++Builder asosan ichki foydalanish uchun rezervlaydi. Boshqa kengaytirishlar (`_int8`, `_int6` va h.k.) ochiq-oydin tushunarli boʻlib turibdi, shuning uchun bu erda ular koʻrib chiqilmaydi. Bizning diqqatimiz C++ning eng ahamiyatli kengaytirishlariga qaratiladi. Ular asosan tarkibli sinflarga mansub boʻlib, kitob matnida ham, C++Builder muhitida ishlab chiqilayotgan ilovalaringizda ham muttasil uchrab turadi.

1.2. Komponentlar (tarkibiy qismlar)

Komponentlar koʻp oʻrinda, C++standart sinflariga qaraganda, yuqoriroq darajadagi inkapsulyalashga erishadilar. Buni tugmachaga ega boʻlgan dialogni ishlab chiqish kabi oddiy misolda koʻrib chiqamiz. Windows uchun namunaviy C++dasturida tugmachani «sichqoncha» bilan bosish natijasida `WM_LBUTTONDOWN` xabarining generatsiyasi sodir boʻladi. Bu xabarni dastur yo switch operatorida, yoki chaqiriqlar jadvali (`RESPONCE_TABLE`) ning tegishli satrida «tutib olish»i, keyin esa ushbu xabarga javob protsedurasiga uzatishi kerak.

C++Builder oʻzlashtirilishi qiyin boʻlgan bu kabi dasturlash oʻyinlariga chek qoʻydi. Komponenta tugmachasi avvaldanoq unga `OnClick` voqeasi bilan bosishga javob beradigan qilib dasturlangan. Bu oʻrinda talab qilinayotgan narsa-tayyor

metodni tanlab olish (yoki o'zini yozish) hamda Obyektlar Inspektori yordamida berilgan voqea-hodisaga ishlov bergichga kiritish.

1.3. Komponentli sinflarni e'lon qilish

C++Builder tarkibiga kiradigan Vizual Komponentalar Kutubxonasi - VCL sinflarining ilgarilovchi e'lonlari `_declspec` modifikatoridan foydalanadi:

`_declspec(<spetsifikator>)`

Bu kalit-so'z, nafaqat bevosita modifikatsiyalanayotgan e'lon oldidan, balki e'lonlar ro'yxatining to'g'ri kelgan erida paydo bo'lishi mumkin, bunda spetsifikator quyidagi qiymatlardan birini qabul qiladi:

`delphiclass` - u TObject sinfiga tegishli VCL ning bevosita yoki bilvosita hosilalarining ilgarilovchi e'loni uchun qo'llanadi. U VCL ning RTTI, konstruktorlar, destruktorga va istisnolar bilan muomalasida muvofiqlik qoidalarini belgilaydi.

`delphireturn` - u Currency, AnsiString, Variant, TDateTime va Set sinflariga tegishli VCL ning bevosita yoki bilvosita hosilalarining ilgarilovchi e'loni uchun qo'llanadi. U VCL ning parametrlar va a'zo=funksiyalarning qaytarilayotgan qiymatlari bilan muomalasida muvofiqlik qoidalarini belgilaydi.

Pascalimplementation tarkibli sinf Obyektli Paskal tilida ishga tushirilganini ko'rsatadi.

VCL sinf quyidagi cheklanishlarga ega:

- Virtual bazaviy sinflarga vorislik qilish man etilgan.
- Tarkibli sinflarning o'zlari vorislik uchun bazaviy sinf sifatida xizmat qila olmaydi.
- Tarkibli Obyektlar uyumning dinamik xotirasida new operatori yordamida yaratiladi.

1.4. O‘zgaruvchan e’lonlar

Fon masalasi, uzish ishlatgichi yoki kiritish-chiqarish porti tomonidan o‘zgartirilishi mumkin bo‘lgan o‘zgaruvchini e’lon qilishda **volatile** modifikatori qo‘llanadi:

volatile<tur><Obyekt nomi>;

C++da volatile kalit-so‘zning qo‘llanishi sinflar va a‘zo-funksiyalarga ham tegishlidir. Bu kalit-so‘z ko‘rsatilgan Obyekt qiymatiga nisbatan tahminlar qilishni kompilyatorga ta‘qiqlaydi, chunki bunday qilinsa, ushbu Obyektni o‘z ichiga olgan ifodalarni hisoblashda, uning qiymati har bir daqiqada o‘zgarib ketishi mumkin. Bundan tashqari o‘zgarib turadigan o‘zgaruvchi register modifikatori bilan e’lon qilinishi mumkin emas. Listing 3.15 ticks o‘zgaruvchisini vaqtli uzilishlar qayta ishlagichi modifikatsiya qiladigan taymerni ishga tushirishga misol bo‘la oladi.

```
{  
volatile int ticks;  
void timer( ) // Taymer funksiyasini e’lon qilish  
tickC++;  
void wait (int interval)  
ticks = 0;  
while (ticks < interval); // Kutish sikli  
}
```

Aytaylik, *o‘zilishni qayta ishlatgichi - timer* real vaqt soatidagi apparat uzilishi bilan tegishli tarzda assotsiatsiya qilindi. ticks o‘zgaruvchisining qiymati ushbu qiymat parametri tomonidan berilgan vaqt intervaliga teng kelmaguncha, wait protsedurasi kutish siklini ishlataveradi. C++kompilyatori sikl ichidagi har bir qiyoslash oldidan volatile ticks o‘zgaruvchisining qiymatini, sikl ichidagi o‘zgaruvchining qiymati o‘zgarmaganiga qaramay, ortiqcha yuklashi lozim. Ayrim optimallashtiruvchi kompilyatorlar bunday «havfli»xatoga yo‘l qo‘yishlari mumkin.

Xatto konstantali ifodaga kirganiga qaramay o'zgartirilishi mumkin bo'lgan o'zgaruvchan o'zgaruvchining boshqa bir turi **mutable** modifikatori yordamida e'lon qilinadi:

```
mutable<o'zgaruvchining nomi>;
```

mutable kalit-so'zning vazifasi shundan iboratki, u biron-bir sinf ma'lumotlari a'zolarini spetsifikatsiya qiladi, bunda ushbu ma'lumotlar a'zolari mana shu sinfning konstantali funktsiyalari tomonidan modifikatsiya qilinishi mumkin bo'lishi kerak. Ma'lumotlar a'zosi sount ni F1 konstantali funktsiya modifikatsiya qiladigan misolni ko'rib chiqaylik:

```
class A{  
  public: mutable int count; int F1 (int p=0)const// F1 funktsiyasini e'lon  
  qilish  
  count=p++return count;//PI count ni qaytarib beradi  
  }  
  void vain(){  
  A, a;  
  Cout<<a.F1(3)<<endl; //main 4 qiymatini chiqarib beradi  
  RTTI turlarining identifikatsiyasi
```

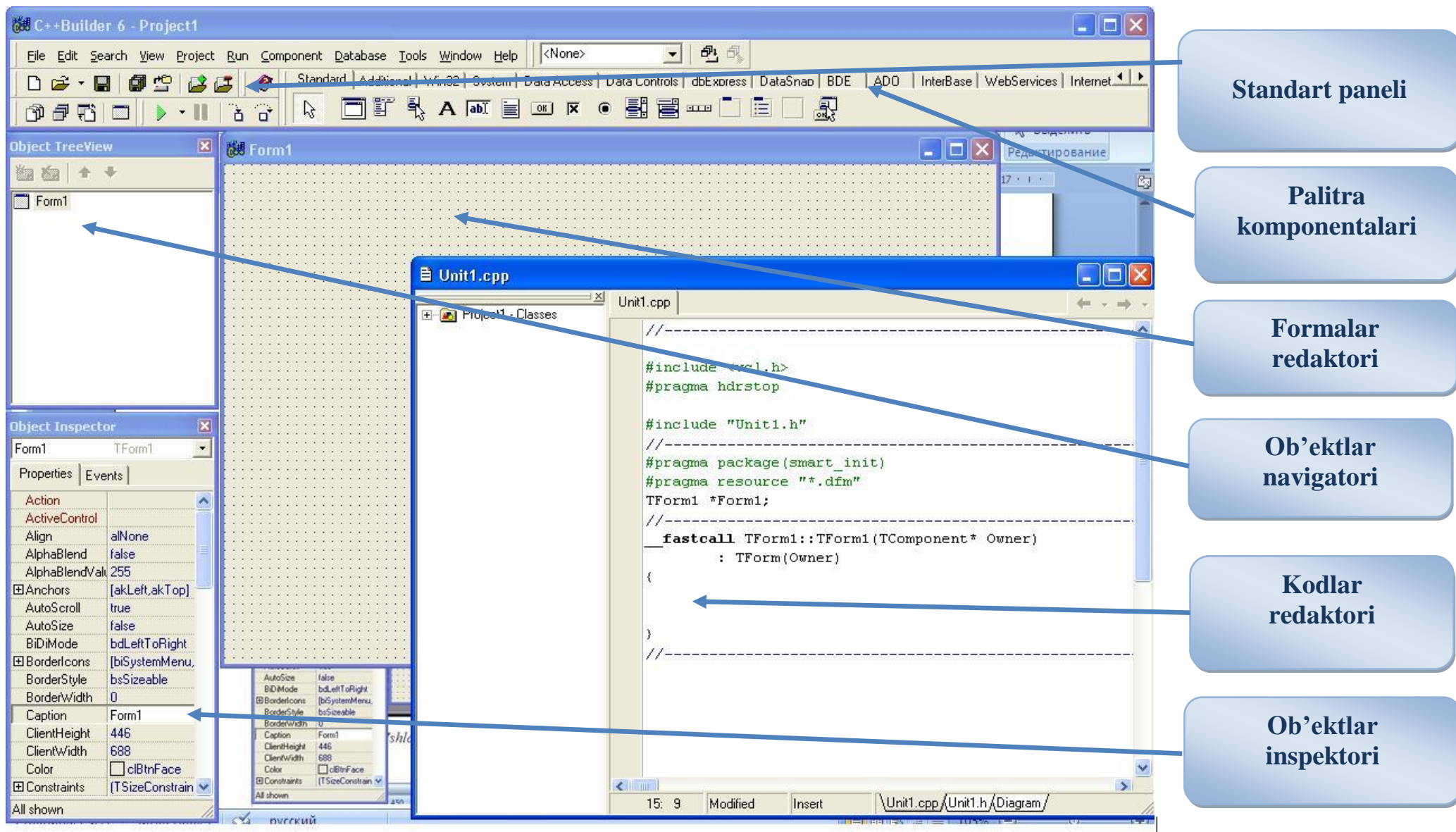
RTTI (Run-Time Type Identification) dasturini bajarishda turlarning identifikatsiyasi sizga o'tkaziladigan dasturni yozish imkonini beradi. Bunda, agar bajarilish vaqtida dasturda ushbu Obyekt ko'rsatkichigagina kirish huquqi bo'lgan taqdirda ham, dastur Obyektning faktik turini aniqlashga qodir bo'ladi. Bu, masalan, virtual bazaviy sinf ko'satkichini ushbu sinfga mansub faktik Obyektning hosila turi ko'satkichiga qayta o'zgartirish imkonini beradi. Shunday qilib, turlar faqat statik tarzda - kompilyatsiya fazasidagina emas, balki dinamik tarzda - bajarilish jarayonida ham qayta o'zgartirilishi mumkin. Ko'rsatkichni berilgan turga dinamik qayta o'zgartirish **dynamic_cast** operatori yordamida amalga oshiriladi.

RTTI mexanizmi ham Obyekt biron-bir aniq turga egami yoki ikkita Obyektning ikkalasi ham bitta turga tegishlimi ekanini tekshirib ko'rish imkonini beradi. typeid operatori argument (dalil) ning faktik turini aniqlaydi hamda ko'rsatkichni ushbu turni tavsiflaydigan typeid sinfi Obyektiga qaytaradi.

Bajarilish paytida RTTI ni Obyektlar Inspektoriga qaytarar ekan, C++Builder unga ushbu sinf xususiyatlari va a'zolarining turlari haqida axborot beradi.

1.5. C++Builder dasturining asoslari

Ishlab chiqishning integratsiyalashgan muhiti Komponentalar palitrasini birlashtiradi. Shakllar Muharriri, Kod Muharriri, Obyektlar Noziri, Obyektlar Xazinasi - bular hammasi kod va zahiralar ustidan to'liq nazoratni ta'minlovchi dasturiy ilovalarni tez ishlab chiqish instrumetlari (1.2.1-rasm).



1.2.1-rasm. Ishlab chiqish muhitining tuzilishi

■ **Komponentlar Palitrasi** ilovalarni qurishda taklif qilinadigan 100 dan ortiq takroran qo‘llanadigan komponentlardan iborat.

■ **SHakllar Muharriri** dasturning foydalanuvchi bilan interfeysini yaratish uchun mo‘ljallangan.

■ **Kod Muharriri** dastur matnini, xususan, voqealarga ishlov berish funktsiyalarini yozish uchun mo‘ljallangan.

■ **Obyektlar Noziri** qotib qolgan chigal dasturlash zaruratisiz Obyektlar xususiyatlarini vizual o‘rnatish imkonini beradi hamda shunday voqealarni o‘z ichiga oladiki, bu voqealarni ularning paydo bo‘lishiga nisbatan Obyektlar reaksiyasi kodlari bilan bog‘lash mumkin bo‘ladi.

■ **Obyektlar Xazinas**i ma’lumotlarning shakl va modullari kabi Obyektlarga ega bo‘lib, ular ishlab chiqishda muvaqqat sarflarni kamaytirish maqsadida ko‘plab ilovalar bilan bo‘linadi.

C++Builder ilovalarni qurishning vizual metodikasini Komponentlar Palitrasidan kerakli boshqarish elementlarini tanlab olish vositasida joriy etadi. Har bir komponenta (masalan, tugmacha) bilan ushbu komponenta turini va xulq-atvorini o‘zgartiradigan xususiyatlar bog‘liq bo‘ladi. Har qanday komponenta ushbu komponentaning turli xildagi ta’sirlarga reaksiyasini (munosabatini) aniqlab beradigan voqealar seriyasini keltirib chiqarishi mumkin. Bunday keyin => belgilari siz C++Builder muhitida amalga oshiradigan xatti-harakatlarni bildiradi.

=>C++Builder ni chaqiring va bosh menyudagi File | New Application komandasi bo‘yicha yangi ilovalar ustida ishlashni boshlang.

=>sichqonchani Komponentalar Palitrasining qo‘shimcha ilovalari ustida bosib, foydalanuvchi ish ko‘radigan dastur interfeysi elementlarining mavjud assortimentini ko‘rib chiqing.

Palitraning bir qo‘shimcha ilovasidan ikkinchisiga o‘tib, kirish mumkin bo‘lgan komponentlar to‘plami o‘zgarayotganining guvohi bo‘lishimiz mumkin. Sichqoncha kursori komponentlar belgisi ustida to‘xtaganda, aytib turish nomi paydo bo‘ladi. Agar F1 klavishasini bossak, tizimning ma’lumotnomalar xizmati tanlab olingan komponenta haqida to‘liq ma’lumot chiqarib beradi.

1.6. Vizual loyihalash

Bizning birinchi ilovamiz bolalarning «O‘nta negr bolasi» sanoq she’rini generatsiya qiladi. Dastlabki versiyada faqat uchta Obyekt kerak bo‘ladi: ro‘yxat, tahrir qilish maydoni va tugmacha. Komponentalarni loyihalash shakliga olib o‘tamiz hamda ilovani asta-sekin rivojlantira boshlaymiz. Tashib olib o‘tish metodi (drag-and-drop) quyidagilardan iborat: sichqoncha tugmachasini tanlab olingan komponenta ustida bosing, kursorni shaklning to‘g‘ri kelgan eriga o‘tkazing, keyin esa sichqoncha tugmachasini yana bosing. Boshida faqat «standart» Palitra Komponentlari bilan cheklanamiz:

=> **Standart** qo‘shimcha ilovani tanlab oling.

=> Ro‘yxat komponentasini ListBox shakliga olib o‘ting.

=> Tahrir qilinagan kiritish maydoni EditBox ni olib o‘ting.

=> Button tugmachasi komponentasini olib o‘ting.

=> Komponentalarni o‘zingizning ilovangizdagi darchada qanday ko‘rmoqchi bo‘lsangiz, shunday joylashtiring va o‘lchamlarini shunday o‘zgartiring.

Obyekt Noziri yordamida komponentalar xususiyatlarining boshlang‘ich qiymatlarini aniqlang. Items ro‘yxatining xususiyatlar qiymatlari katagida tugmachani bosing, ochilgan muharrir darchasida she’rning dastlabki 7 satrini kiriting. Shakl va tugmachaning Caption xususiyatida ularning ma’noli nomlarini ko‘rsating (mos ravishda, «O‘nta negr bolasi» va «Natija»). Tahri qilish maxdonining Text xususiyatida natijani aytib berish satrini kiriting («To‘qqizta negr bolasi»).

Endi Kod Muharririga ulanish hamda, avval qabul qilinganidek, C++tilidagi har qanday dasturni yozish mumkin, shu jumladan, ANSI/ISO standartining so‘nggi kengaytmalarini ham. Biroq, avval ilovalarni tez ishlab chiqishning yangi vositalari hamda C++Builder da mavjud bo‘lgan qo‘shimcha komponentalar atributlaridan foydalanishga harakat qilib ko‘ramiz.

2. C++ Builder dasturida chiziqli jarayonlarni dasturlash

2.1. Y sonini X-darajali ildizini hisoblash dasturi

1. Masalani yechishda quyidagi formuladan foydalanamiz:

$$\sqrt[x]{y} = y^{\frac{1}{x}} = e^{\ln y^{\frac{1}{x}}} = e^{\frac{1}{x} \ln y}$$

Dastur kodi:

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
float x, y, z;
x=StrToFloat(Edit1->Text);
y=StrToFloat(Edit2->Text);
z=exp((1/x)*log(y));
Label1->Caption=(FloatToStr(z));
}
//-----
void __fastcall TForm1::Button2Click(TObject *Sender)
{
Edit1->Text= " ";
Edit2->Text= " ";
Label1->Caption= " ";
}
//-----
void __fastcall TForm1::Button3Click(TObject *Sender)
{
Close();
}
```

Dastur natijasi:

2.2. Ixtiyoriy uchburchaklarning yuzasini hisoblash dasturi

1. Masalani yechishda Geron formulasidan foydalanamiz:

$$P = \frac{a + b + c}{2},$$

$$S = \sqrt{P(P - a)(P - b)(P - c)}.$$

Formulada **a**, **b**, **c** lar uchburchakning tomonlari **P**, **S** lar uchburchak yarim perimetr va yuzasi.

Dastur kodi:

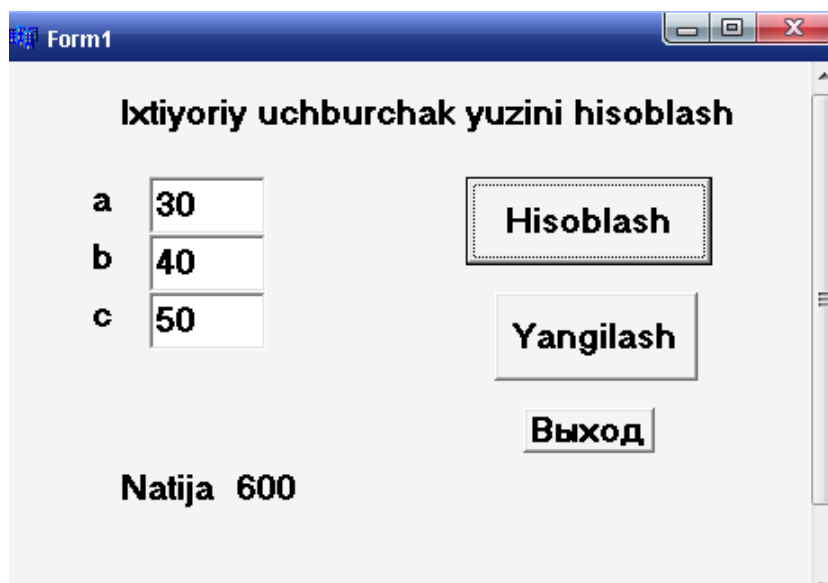
```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
float a, b, c, P, S;
a=StrToFloat(Edit1->Text);
b=StrToFloat(Edit2->Text);
c=StrToFloat(Edit3->Text);
P=(a+b+c)/2;
S=sqrt(P*(P-a)*(P-b)*(P-c));
Label7->Caption=(FloatToStr(P));
Label8->Caption=(FloatToStr(S));
}
```

```
//-----
void __fastcall TForm1::Button2Click(TObject *Sender)
{
    Edit1->Text= " ";
    Edit2->Text= " ";
    Edit3->Text= " ";
    Label7->Caption= " ";
    Label8->Caption= " ";
}
//-----

void __fastcall TForm1::Button3Click(TObject *Sender)
{
    Close();
}

```

Dastur natijasi:



2.3. Ikkinchi tartibli Determinantlarni hisoblash

1. Ikkinchi tartibli determinantlar quyidagicha ko`rinishga ega:

$$\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}.$$

Bu yerdagi $a_{ij}(i=2, j=2)$ sonlardan iborat bo`lib, determinantning elementlaridir.

Ikkinchi tartibli determinantni hisoblash dasturini ko`rib chiqamiz.

Dastur kodi:

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
float a1,a2,b1,b2,d;
a1=StrToFloat(Edit1->Text);
a2=StrToFloat(Edit2->Text);
b1=StrToFloat(Edit3->Text);
b2=StrToFloat(Edit4->Text);
d=a1*b2-b1*a2;
Label10->Caption=("Natija D = "+FloatToStrF(d,ffFixed,8,2));
}
//-----

void __fastcall TForm1::Button2Click(TObject *Sender)
{
Edit1->Text= " ";
Edit2->Text= " ";
Edit3->Text= " ";
Edit4->Text= " ";
Label10->Caption= " ";
}
//-----

void __fastcall TForm1::Button3Click(TObject *Sender)
{
Close();
}
//-----

void __fastcall TForm1::Hisoblash1Click(TObject *Sender)
{
float a1,a2,b1,b2,d;
```

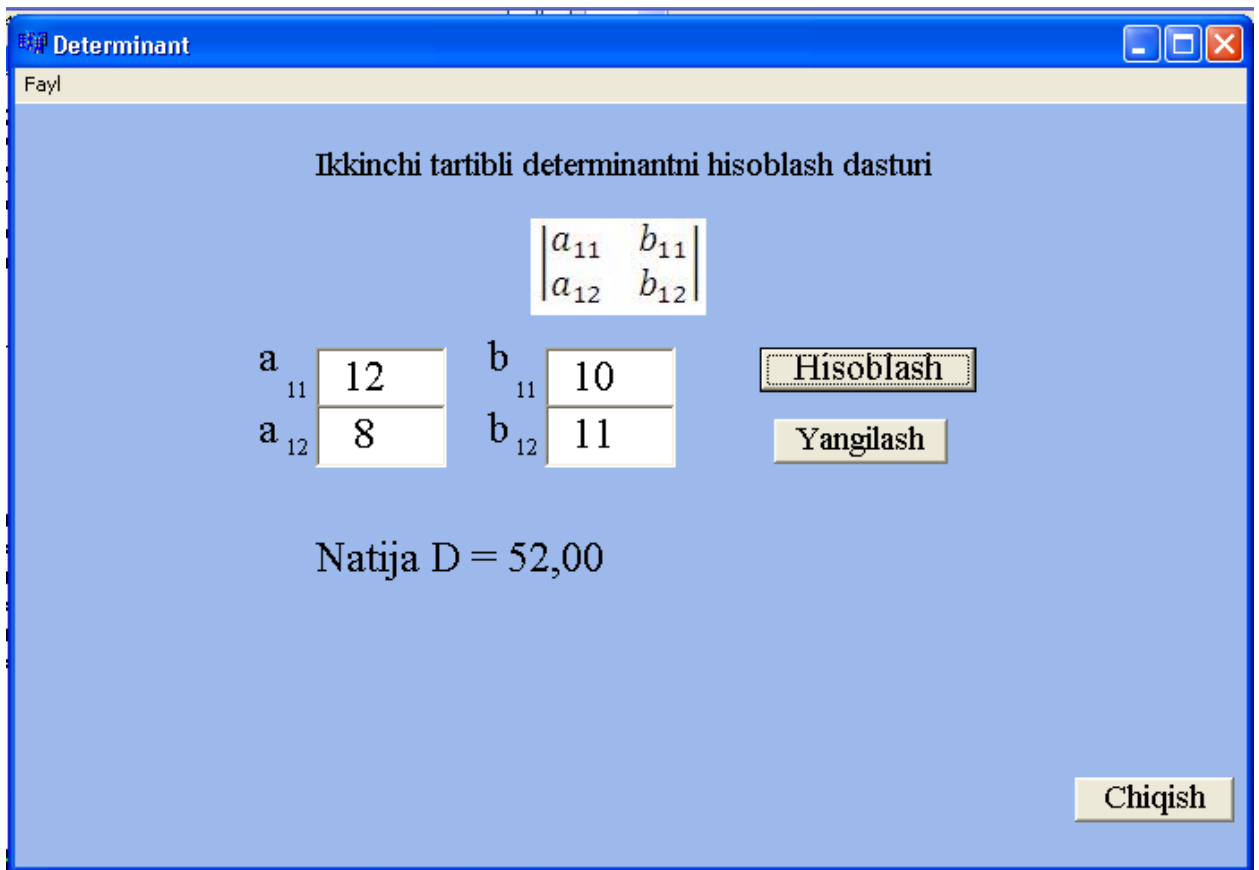


```

a1=StrToFloat(Edit1->Text);
a2=StrToFloat(Edit2->Text);
b1=StrToFloat(Edit3->Text);
b2=StrToFloat(Edit4->Text);
d=a1*b2-b1*a2;
Label10->Caption=("Natija D = "+FloatToStrF(d,ffFixed,8,2));
}
//-----
void __fastcall TForm1::Yangilash1Click(TObject *Sender)
{
Edit1->Text= " ";
Edit2->Text= " ";
Edit3->Text= " ";
Edit4->Text= " ";
Label10->Caption= " ";
}
//-----
void __fastcall TForm1::Chiqish1Click(TObject *Sender)
{
Close();
}

```

Dastur natijasi:



2.4. Chiziqli tenglamalar sistemasini Kramer usulida yechish

1. Uch noma'lumli chiziqli tenglamalar sistemasini berilgan bo'lsin:

$$\begin{cases} a_1x + b_1y + c_1z = d_1 \\ a_2x + b_2y + c_2z = d_2 \\ a_3x + b_3y + c_3z = d_3 \end{cases}$$

Asosiy va yordamchi determinantlarni tuzamiz:

$$\Delta = \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}, \Delta x = \begin{vmatrix} d_1 & b_1 & c_1 \\ d_2 & b_2 & c_2 \\ d_3 & b_3 & c_3 \end{vmatrix},$$

$$\Delta y = \begin{vmatrix} a_1 & d_1 & c_1 \\ a_2 & d_2 & c_2 \\ a_3 & d_3 & c_3 \end{vmatrix}, \Delta z = \begin{vmatrix} a_1 & b_1 & d_1 \\ a_2 & b_2 & d_2 \\ a_3 & b_3 & d_3 \end{vmatrix}.$$

Agar $\Delta \neq 0$ bo'lsa, sistema yagona $x = \frac{\Delta x}{\Delta}$, $y = \frac{\Delta y}{\Delta}$, $z = \frac{\Delta z}{\Delta}$ yechimlarga ega bo'ladi.

Uchta noma'lumli chiziqli tenglamalar sistemasini Kramer usulida yechish dasturini ko'rib chiqamiz.

Dastur kodi:

```

void __fastcall TForm1::Button1Click(TObject *Sender)
{
float a1,a2,a3,b1,b2,b3,c1,c2,c3,d,dx,dy,dz,x,y,z,d1,d2,d3;
a1=StrToFloat(Edit1->Text);
a2=StrToFloat(Edit2->Text);
a3=StrToFloat(Edit3->Text);
b1=StrToFloat(Edit4->Text);
b2=StrToFloat(Edit5->Text);
b3=StrToFloat(Edit6->Text);
c1=StrToFloat(Edit7->Text);
c2=StrToFloat(Edit8->Text);
c3=StrToFloat(Edit9->Text);
d1=StrToFloat(Edit10->Text);
d2=StrToFloat(Edit11->Text);
d3=StrToFloat(Edit12->Text);
d=a1*b2*c3+b1*c2*a3+a2*b3*c1-c1*b2*a3-a1*b3*c2-b1*a2*c3;
dx=d1*b2*c3+b1*c2*d3+d2*b3*c1-c1*b2*d3-c1*b3*d1-b1*d2*c3;
dy=a1*d2*c3+d1*c2*a3+a2*d3*c1-c1*d2*a3-a1*d3*c2-d1*a2*c3;
dz=a1*b2*d3+b1*d2*a3+a2*b3*d1-d1*b2*a3-a1*b3*d2-b1*a2*d3;
x=dx/d;
y=dy/d;
z=dz/d;
Label28->Caption=("x = "+FloatToStrF(x,ffFixed,8,2));
Label29->Caption=("y = "+FloatToStrF(y,ffFixed,8,2));
Label30->Caption=("z = "+FloatToStrF(z,ffFixed,8,2));
Label35->Caption=("delta = "+FloatToStrF(d,ffFixed,8,2));
Label36->Caption=("delta x = "+FloatToStrF(dx,ffFixed,8,2));
Label37->Caption=("delta y = "+FloatToStrF(dy,ffFixed,8,2));
Label38->Caption=("delta z = "+FloatToStrF(dz,ffFixed,8,2));
}

```

```

//-----
void __fastcall TForm1::Button2Click(TObject *Sender)
{
Edit1->Text= " ";
Edit2->Text= " ";
Edit3->Text= " ";
Edit4->Text= " ";
Edit5->Text= " ";
Edit6->Text= " ";
Edit7->Text= " ";
Edit8->Text= " ";
Edit9->Text= " ";
Edit10->Text= " ";
Edit11->Text= " ";
Edit12->Text= " ";
Label28->Caption= " ";
Label29->Caption= " ";
Label30->Caption= " ";
Label35->Caption= " ";
Label36->Caption= " ";
Label37->Caption= " ";
Label38->Caption= " ";
}
//-----

void __fastcall TForm1::Button3Click(TObject *Sender)
{
Close();
}

```

Dastur natijasi:

Tenglamalar sistemasini Kramer usulida yechish

$$\begin{cases} a_1x + b_1y + c_1z = d_1 \\ a_2x + b_2y + c_2z = d_2 \\ a_3x + b_3y + c_3z = d_3 \end{cases}$$

Koeffisientlari:

a_1	<input type="text" value="5"/>	b_1	<input type="text" value="3"/>	c_1	<input type="text" value="3"/>	d_1	<input type="text" value="0"/>
a_2	<input type="text" value="6"/>	b_2	<input type="text" value="-1"/>	c_2	<input type="text" value="2"/>	d_2	<input type="text" value="5"/>
a_3	<input type="text" value="1"/>	b_3	<input type="text" value="2"/>	c_3	<input type="text" value="1"/>	d_3	<input type="text" value="4"/>

Natijalar: $x = 25,50$
 $y = 21,00$
 $z = -63,50$

delta = 2,00
delta x = 51,00
delta y = 42,00
delta z = -127,00

Xulosa

Ma'lumki, dastur mashina kodlaridagi qanday ketma–ketligi bo'lib, aniq bir hisoblash vositasini amal qilishini boshqaradi. Dastur vositasi yaratish jarayonini osonlashtirishi uchun yuzlab programalash tillari yaratilgan. Barcha dasturlash tillarini ikkita toifaga ajratish mumkin:

- quyi darajadagi dasturlash tillari;
- yuqori darajadagi dasturlash tillari.

Quyi darajadagi dasturlash tillariga Assembler tillari kiradi. Bu tillar nisbatan qisqa va tezkor bajariluvchi kodlarni yaratish imkoniyatini beradi. Lekin, Assembler tilida dastur tuzish zahmatli, nisbatan uzoq davom etadigan jarayondir. Bunga qarama–qarshi ravishda yuqori bosqich tillari yaratilganki, ularda tabiiy tilning (ingliz tilining) cheklangan ko'rinishidan foydalangan holda dastur tuziladi. Yuqori bosqich tillaridagi operatorlar, berilganlarning turlari, o'zgaruvchilar va dastur yozishning turli usullari tilning ifodalash imkoniyati oshiradi va dasturni «o'qimishli» bo'lishini ta'minlaydi. Yuqori bosqich tillariga Fortran, PL/1, Prolog, Lisp, Basic, Pascal, C va boshqa tillarni misol keltirish mumkin. Kompyuter arxitekturasini takomillashuvi, kompyuter tarmog'ining rivojlanishi mos ravishda yuqori bosqich tillarini yangi variantlarni yuzaga kelishiga, angi tillarni paydo bo'lishiga, ayrim tillarni yo'qolib ketishiga olib keldi. Hozirda keng tarqalgan tillarga Object Pascal, C++, C#, C++ Builder, Php, Java, Asp tillari hisoblanadi. Xususan, C tilining takomillashgan variantlari sifatida C++, C++ Builder tillarini olishimiz mumkin. 1972 yilda Denis Ritch va Brayan Kernegi tomonidan C tili yaratildi. 1980 yilda Byarn Straustrop C tilining avlodi C++ tilini yaratdiki, unda strukturali va ob'ektga yo'naltirilgan dasturlash texnologiyasiga tayangan holda dastur yaratish imkoniyati tug'ildi.

Mazkur kurs ishida C++ Builder dasturi yordamida chiziqli jarayonlariga doir masalalarini yechish texnologiyalari o'rganildi. Ularning qo'llanilishi sodda dasturlar tuzish orqali tushuntirildi.

FOYDALANILGAN ADABIYOTLAR

1. Никита Культин С++ Builder в задачах и примерах. Санкт-Петербург «БХВ-Петербург» 2005.
2. А.А. Навроцкий, А.К. синицын, А.В. Щербаков. Программирование в среде С++ Builder. Минск 2002.
3. Эллис М., Страуструп Б. Справочное руководство по языку программирования С++ с комментариями. Проект стандарта ANSI: Пер. с англ.-М.:Мир, 1992.-445с.
4. Керниган Б., Ритчи Д. Язык программирования Си: Пер. с англ. –М.: Финансы и статистика, 1992. -272с.
5. Turbo С++. Руководство пользователя: Пер. с англ. -М.: СП ИНТЕРКВАДРО, 1991.-298с.
6. Дункан Р. Си++ для тех, кто знает Си // PC Magazine/ USSR, 1991 -№3. – с.84-106.
7. Мадрахимов Ш.Ф., Гайназаров С.М. С++ программалаш тили. –Тошкент: «УзМУ», 2008. -80 б.
8. Сайфиев Ж.Ф. С++ тилига кириш.- Бухоро: БухДУ, 2005. -147 б.
9. Белкин В. Обработка исключительных ситуаций в Си++: что, когда, как. PC magazine / Russian edition, 1995. -№4. –с. 180-186.
10. Зуев Е., Кротов А. Новые возможности Си++ // PC magazine / Russian edition, 1994. -№7. –с. 176-181.
11. Бабе Бруно Просто и ясно о Borland С++. Пер. с англ. –М.: Бином, 1995. - 400 с.
12. www.ziyonet.uz - Axborot ta`lim tarmog`I sayti.
13. www.referat.uz - Referatlar, kurs ishi va diplom ishlari.
14. www.exponenta.ru - Matematik tizimlar haqidagi sayt.
15. www.Intuit.ru - Интернет-Университет информационных технологий. Москва.