O’ZBEKISTON RESPUBLIKASI XALQ TA’LIMI
VAZIRLIGI
NAVOIY DAVLAT PEDAGOGIKA INSTITUTI
FIZIKA-MATEMATIKA FAKULTETI
«Informatika va axborot texnologiyalari» кafedrasi

KURS ISHI

MAVZU: Massivlar va matritsalar ustida amallar bajarish (C++ Builder dasturida)

Qabul qildi: Xodjabayev F.D.
Bajardi: 3 kurs talabasi Temirova G.

Navoiy-2016

MUNDARIJA

K I R I SH

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 algoritmni 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 preprotsessor 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 etadi.

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

1.1. C++ dasturlash tilining yangi imkoniyatlari

C++Builder, nafaqat ANSI C++ standarti kiritayotgan yangiliklarni qo‘llab-quvvatlaydi, balki tilni yangi imkoniyatlar bilan boyitadi. Shuni tushunib olish muhimki, tilni kengaytirish hech qachon quruq maqsad bo‘lib 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 to‘liq foydalanish uchun, kiritilgan til kengaytirishlarni qabul qilishingizga to‘g‘ri keladi.

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

1.2. Komponentlar (tarkibiy qismlar)

Komponentlar ko‘p o‘rinda, C++standart sinflariga qaraganda, yuqoriroq darajadagi inkapsulyalashga erishadilar. Buni tugmachaga ega bo‘lgan dialogni ishlab chiqish kabi oddiy misolda ko‘rib chiqamiz. Windows uchun namunaviy C++dasturida tugmachani «sichqoncha» bilan bosish natijasida WM_LBUTTONDOWN xabarining generatsiyasi sodir bo‘ladi. 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 o‘zlashtirilishi qiyin bo‘lgan bu kabi dasturlash o‘yinlariga chek qo‘ydi. Komponenta tugmachasi avvaldanoq unga OnClick voqeasi bilan bosishga javob beradigan qilib dasturlangan. Bu o‘rinda talab qilinayotgan narsa-tayyor

metodni tanlab olish (yoki o‘zinikini 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, destruktor 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=funktsiyalarning 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-funktsiyalarga 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)<<endi; //main 4 qiymatini chiqarib beradi
```

RTTI turlarining identifikatsiyasi

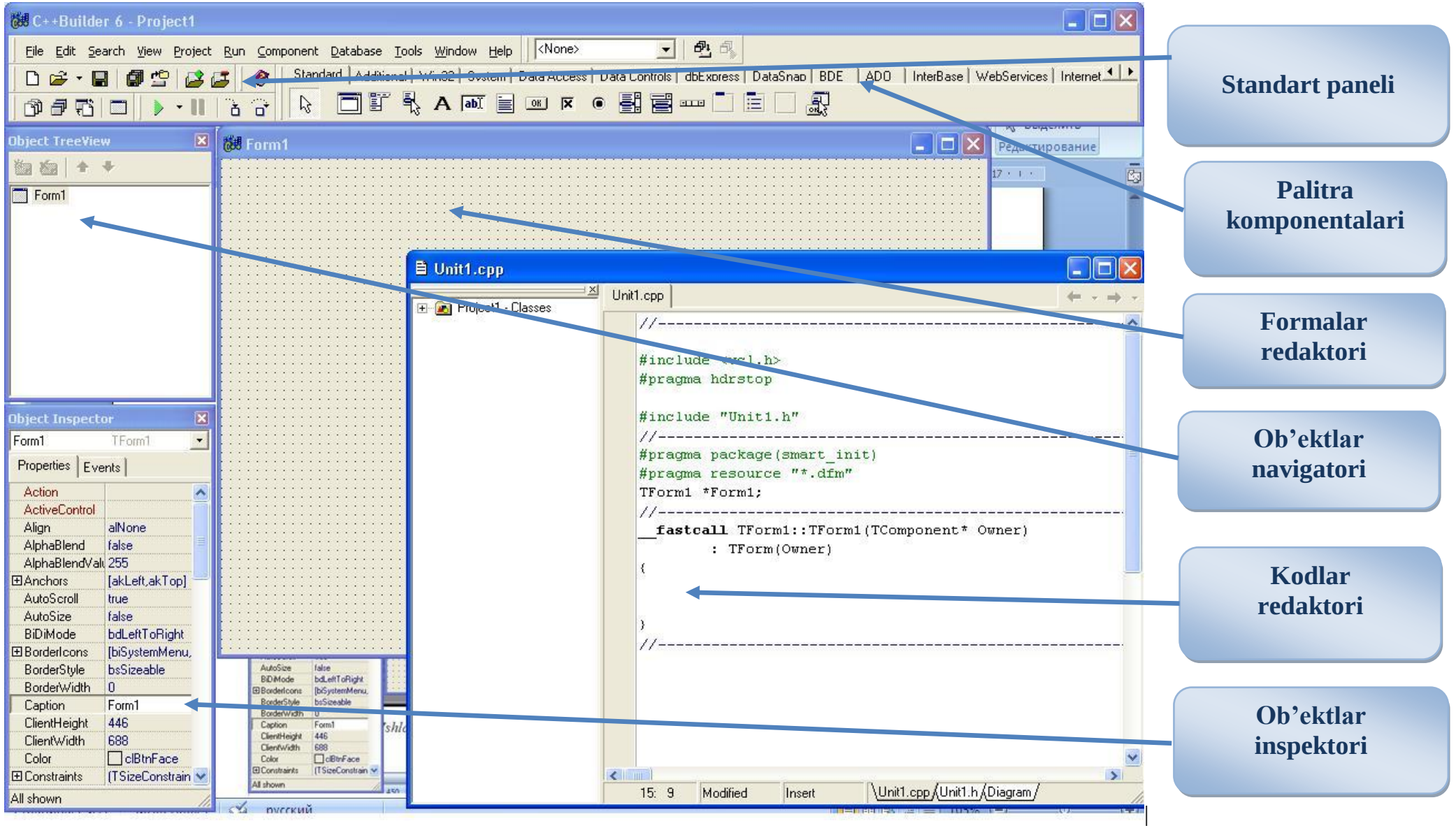
RTTI (Run-Tame Type Identification) dasturini bajarishda turlarning identifikatsiyasi sizga o‘tkaziladigan dasturni yozish imkonini beradi. Bunda, agar bajarilish vaqtida dasturda ushbu Obyekt ko‘rsatkichigagina kirish huquqi bo‘lgan taqdirda ham, dastur Obyektning faktik turini aniqlashga qodir bo‘ladi. 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 typeinto 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 fnuktsiyalarini 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 reaktsiyasi kodlari bilan bog‘lash mumkin bo‘ladi.

■ ***Obyektlar Xazinasi*** ma’lumotlarning shakl va modullari kabi Obyektlarga ega bo‘lib, ular ishlab chiqishda muvaqqat sarflarni kamaytirish maqsadida ko‘plpb 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 reaktsiyasini (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 Fale | New Arrlication 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 qilinatgan 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‘yxatiningxususiyatlar 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 massivlar bilan ishlash

Odatda massivlarning tartiblanishi bir xil turdagi elementlardan tashkil topadi va ularning birlashmasi bir xil nomlanadi. Massivlarning elementlari o'z nomiga ega, massiv elementlari kvadrat qovus ostiga olinadi va ular bir nechta bo'lishi ham mumkin: a[1], bb[I], c12[I,j*2], q[1,1,I*j-1]… Ushbu massiv elementlaridan ixtiyoriy tartibdagi massiv elementlarini hosil qilish mumkin. C++ dasturida massiv indekslari har doim noldan boshlanadi.

Dastur kodida massivlarning yozilishi:

```
const Nmax=10;                          // Berilgan maksimal indeks qiymati;
typedef double mas1[Nmax];              // Bir o'lchamli massiv turlarining yozilishi;
typedef double mas2[Nmax][Nmax];        // Ikki o'lchamli massiv turlarining yozilishi;
mas1 A;                                 // B – mas1 massiv turi;
mas2 B;                                 //A – mas2 massiv turi;
int Ss[10];                             // Ss – massivning o'ndan bir qismi;
char Y[5][10];                          // Y – ikki o'lchamli massiv turi.
```

Massiv elementlaridan foydalanib quyidagi o'zgaruvchi ifodalarni hosil qilish mumkin:

```
F=2*A[3]+A[Ss[I]+1]*3;
A[n]=1+sqrt(fabs(A[n-1]));
```

TStringGrid komponentasi

Massivlar bilan ishlash jarayonida axborotlarni ekranga kiritish va chiqarishda jadval ko'rinishi qulayroqdir. TStringGrid komponentasi belgilangan qiymatlarni ikki o'lchovli jadval ostida hosil qiladi va har bir yacheyka bir satrli tahrir ostida namoyon bo'ladi (analogli oyna TEdit). Qiymatlarni o'zgartiruvchi xossalar Cells[ACol, ARow: Integer]: string, ACol, Arow – ikki o'lchovli massiv elementining indeksi. ColCount va RowCount xossalari jadval satr va ustunlarini hosil qiladi. FixedCols va FixedRows xossalari esa fiksirlangan satr va ustun

maydonlarini hosil qiladi. Fiksirlangan maydon belgilangan boshqa rangda hosil bo'ladi va kiritilayotgan axborotlarni taqiqlamaydi.

2.1. Ushbu $\vec{Y} = A * \vec{B}$ vektorni hisoblash dasturi

1. Masalani yechishda quyidagi formuladan foydalanamiz:

$$Y_i = \sum_{j=0}^{N} A_{ij} \cdot B_i$$

Bu yerda A – kvadrat matritsa, NxN – kvadrat matritsa o'lchamlari va Y, B – bir o'lchamli massivlar.

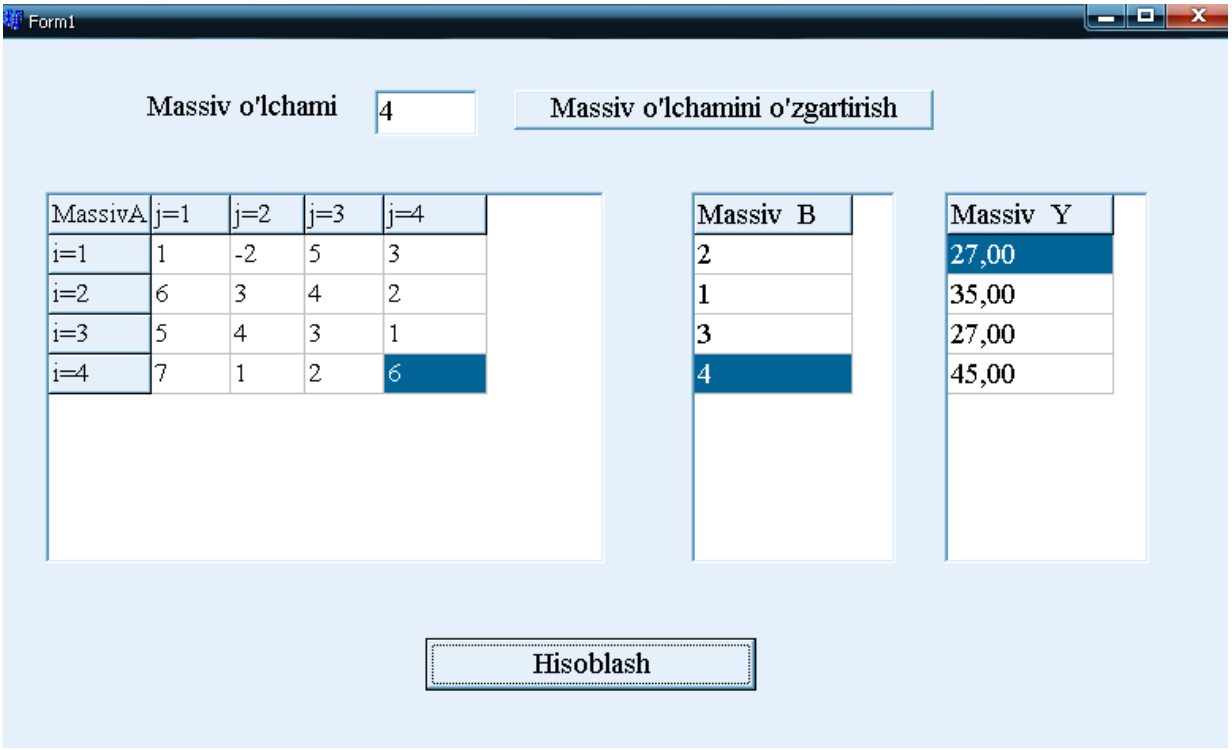
Dastur kodi:

```
#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//---------------------------------------------------------------------------
__fastcall TForm1::TForm1(TComponent* Owner)
        : TForm(Owner)
{
}
//---------------------------------------------------------------------------
const Nmax=10;
typedef double mas2[Nmax][Nmax];
typedef double mas1[Nmax];
int N=3;
void __fastcall TForm1::FormCreate(TObject *Sender)
{
```

```
Edit1->Text=FloatToStr(N);
StringGrid1->ColCount=N+1;
StringGrid1->RowCount=N+1;
StringGrid2->RowCount=N+1;
StringGrid3->RowCount=N+1;
StringGrid1->Cells[0][0]="MassivA";
StringGrid2->Cells[0][0]="Massiv  B";
StringGrid3->Cells[0][0]="Massiv  Y";
for(int i=1; i<=N; i++)
{
StringGrid1->Cells[0][i]="i="+IntToStr(i);
StringGrid1->Cells[i][0]="j="+IntToStr(i);
}
}
//---------------------------------------------------------------------------
void __fastcall TForm1::Button1Click(TObject *Sender)
{
N=StrToInt(Edit1->Text);
StringGrid1->ColCount=N+1;
StringGrid1->RowCount=N+1;
StringGrid2->RowCount=N+1;
StringGrid3->RowCount=N+1;
StringGrid1->Cells[0][0]="MassivA";
StringGrid2->Cells[0][0]="Massiv  B";
StringGrid3->Cells[0][0]="Massiv  Y";
for(int i=1; i<=N;i++)
{
StringGrid1->Cells[0][i]="i="+IntToStr(i);
StringGrid1->Cells[i][0]="j="+IntToStr(i);
}
```

```
}
//---------------------------------------------------------------------------
void __fastcall TForm1::BitBtn1Click(TObject *Sender)
{
mas2 a;
mas1 b,y;
int i,j;
for(i=0; i<N; i++)
for(j=0; j<N; j++)
a[i][j]=StrToFloat(StringGrid1->Cells[i+1][j+1]);
for( i=0; i<N;i++)
b[i]=StrToFloat(StringGrid2->Cells[0][i+1]);
for( i=0; i<N; i++)
{
double s=0;
for( j=0; j<N; j++)
s+=a[j][i]*b[j];
y[i]=s;
StringGrid3->Cells[0][i+1]=FloatToStrF(y[i],ffFixed,8,2);
}
 }
```

Dastur natijasi:



2.2. C++ Builder muhitining konsul rejimida massivlar bilan ishlash

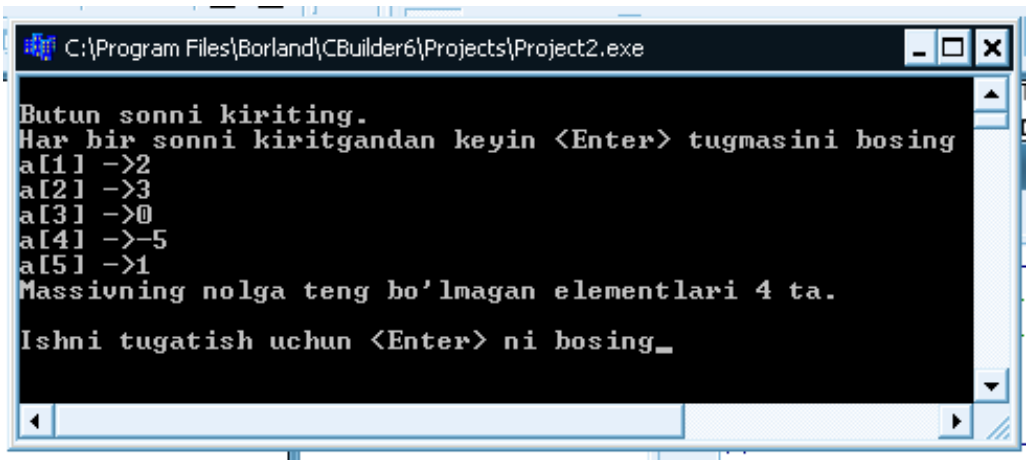
1. Bir o'lchamli 5 ta elementdan iborat bo'lgan massivning nolga teng bo'lmagan elementlar sonini chiqaruvchi dasturini tuzing.

Dastur kodi:

```
#include <stdio.h>
#include <conio.h>
#include <clx.h>
#pragma hdrstop
//---------------------------------------------------------------------------
#define SIZE 5
#pragma argsused
int main(int argc, char* argv[])
{
```

```
int a[SIZE];
int n = 0;
int i;
printf("\nButun sonni kiriting. \n");
printf("Har bir sonni kiritgandan keyin ");
printf("<Enter> tugmasini bosing\n");
for (i = 0; i < SIZE; i++)
{
printf("a[%i] ->",i+1);
scanf("%i", &a[i]);
if (a[i] != 0) n++;
}
printf("Massivning nolga teng bo'lmagan elementlari %i ta.\n", n);
printf("\nIshni tugatish uchun <Enter> ni bosing");
getch();
return 0;
}
```

Dastur natijasi:

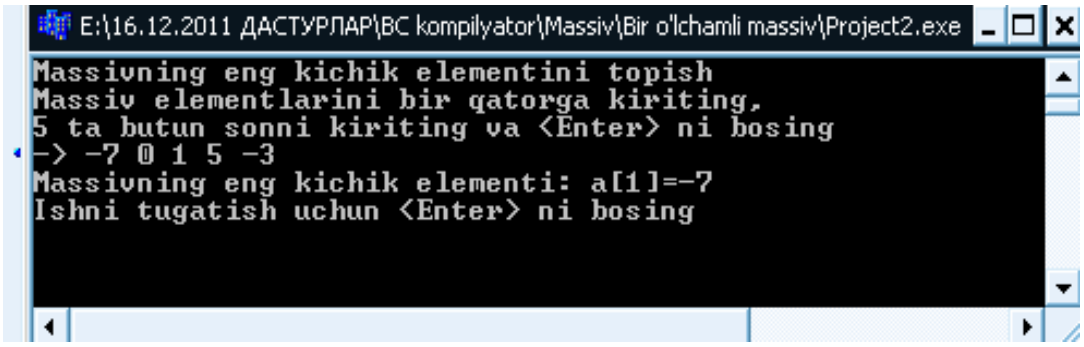


2. Bir o’lchamli 5 ta elementdan iborat bo’lgan massivning eng kichik elementini aniqlash dasturini tuzing.

Dastur kodi:

```
#include <stdio.h>
#include <conio.h>
#include <clx.h>
#pragma hdrstop
//---------------------------------------------------------------------------
#define HB 5
#pragma argsused
int main(int argc, char* argv[])
{
int a[HB];
int min;
int i;
printf("Massivning eng kichik elementini topish\n");
printf("Massiv elementlarini bir qatorga kiriting,\n");
printf("%i ta butun sonni kiriting va <Enter> ni bosing\n", HB) ;
printf("-> ") ;
for (i = 0; i < HB; i++)
scanf("%i",&a[i]);
min =0;
for (i = 1; i < HB; i++)
if (a[i] < a[min]) min = i;
printf("Massivning eng kichik elementi: ");
printf("a[%i]=%i ", min+1, a[min]);
printf("\nIshni tugatish uchun <Enter> ni bosing");
getch();
return 0;
}
```

Dastur natijasi:


E:\16.12.2011 ДАСТУРЛАР\BC kompilyator\Massiv\Bir o'lchamli massiv\Project2.exe
Massivning eng kichik elementini topish
Massiv elementlarini bir qatorga kiriting,
5 ta butun sonni kiriting va <Enter> ni bosing
-> -7 0 1 5 -3
Massivning eng kichik elementi: a[1]=-7
Ishni tugatish uchun <Enter> ni bosing

Xulosa

Ma’lumki, dastur mashina kodlaridagi qanday ketma–ketligi bo’lib, aniq bir hisoblash vositasini amal qilishini boshqaradi. Dastur vositasi yaratish jarayonini osonlashtrishi 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, Asssembler 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 takommillashgan 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 muhitida massivlar va matritsalar ustida amallar bajarildi. Ularning qo`llanilishi sodda dasturlar tuzish orqali tushuntirildi.

FOYDALANILGAN ADABIYOTLAR

1. Никита Культин C++ Builder в задачах и примерах. Санкт-Петербург «БХВ-Петербург» 2005.
2. А.А. Навроцкий, А.К. синицын, А.В. Щербаков. Программирование в среде C++ Builder. Минск 2002.
3. Эллис М., Страуструп Б. Справочное руководство по языку программирования C++ 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 C++. 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. C++ программалаш тили. –Тошкент: «УзМУ», 2008. -80 б.
8. Сайфиев Ж.Ф. C++ 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 C++. 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.