

**O‘ZBEKISTON RESPUBLIKASI OLIY VA O‘RTA MAXSUS
TA‘LIM VAZIRLIGI
ISLOM KARIMOV NOMIDAGI
TOSHKENT DAVLAT TEXNIKA UNIVERSITETI**

**«AXBOROT TEXNOLOGIYALARI»
kafedrası**

**«TEXNIK TIZIMLARDA AXBOROT
TEXNOLOGIYALARI»**

fanidan amaliy mashg‘ulotlarni bajarish uchun
o‘quv-uslubiy ko‘rsatmalar
2-qism

Barcha bakalavriat ta‘lim yo‘nalishlari uchun

TOSHKENT – 2018

Tuzuvchilar: Yuldasheva M.T., Karimova N.O., Akbarova Sh.O., Tojixojayeva N.Z., Kasimova G.I.

“Texnik tizimlarda axborot texnologiyalari” fanidan amaliy mashgʻulotlarni bajarish uchun oʻquv-uslubiy koʻrsatmalar (2-qism)– Toshkent: ToshDTU, 2018. 50 b.

Ushbu oʻquv-uslubiy koʻrsatmalarda ilmiy-texnik masalalarni yechishda masalaning algoritmlarini tuzish asoslari, Windows amaliyot tizimi asosida ishga tushiriladigan C++ Builder 6 dasturlashtirish muhitida dasturlarni bajarish qoidalari yoritilgan. Oʻquv-uslubiy koʻrsatmaning asosiy qismini talabalarda C++ dasturlash tilida chiziqli, tarmoqlanuvchi, takrorlanuvchi algoritmlarni dasturlash, protsedura va funktsiyalardan foydalanish hamda muntazam, faylli, aralash, toʻplam kabi murakkab toifalarni ishlatgan holda dasturlar tuzish, bu jarayonlarni C++ Builder 6ning forma ilovasida bajarish vositalari va usullaridan foydalanish koʻnikmalarini yaratish tashkil etadi.

Har bir amaliy ishning tavsifi qisqacha nazariy qismga ega boʻlib, ishning mohiyati, mazmuni va dasturni tuzish aniq misollar orqali ifodalangan.

Islom Karimov nomidagi Toshkent davlat texnika universiteti ilmiy-uslubiy kengashining qaroriga muvofiq nashrga tayyorlandi.

Taqrizchilar:

Fayzullayev S	TKTI “Informatika avtomatlashtirish va boshqaruv” kafedrasi dotsenti
Tursunov M	TDTU Asbobsozlik kafedrasi dotsenti, t.f.n

© Toshkent davlat texnika universiteti, 2018.

So‘z boshi

Xozirgi kunda ishlab chiqarishning barcha sohalarida axborot texnologiyalaridan keng foydalanilmoqda. Avtomatlashtirilgan loyihalash tizimlari, geometrik modellash dasturlari tarmoqdagi ma‘lumotlar bazalarini ishlab chiqarish jarayonida va texnik tizimlardagi muammolarni yechishda samarali qo‘llanilmoqda. Avtomatlashtirish har bir sohani shiddatli qamrab olayotgan bir vaqtda talabalarning bilimlarni egallashda ta‘lim jarayonida va mustaqil ta‘limda axborot texnologiyalaridan unumli foydalana olishi va zamonaviy texnologiyalarni tadbiq etishi talab qilinadi.

“Texnik tizimlarda axborot texnologiyalari” fanini o‘qitishdan maqsad zamonaviy informatsion fikrlash va ilmiy dunyoqarashni shakllantirilgan holda talabalarni kompyuter imkoniyatlaridan foydalanish, kompyuter bilan muloqot o‘rnatish usullarini o‘rgatish va unda turli masalalarni yecha olishga yo‘naltirishdan iborat. Jumladan, yo‘nalishlar sohalaridagi masalalarni yechishda, mashinasozlikni loyihalashtirishda, mahsulot dizaynini ishlab chihishda, sanoat mahsulotlarini tizimli taxlilida, tizim va kommunikatsiyani loyihalashda, muxandislik taxlilida, elektronika sohasida, geologiya muxandislik ishlarida, mexanika muxandislik sohasida loyihalash jarayonlarini avtomatlashtirishda, texnik tizimlarda ma‘lumotlar almashish jarayonida zamonaviy dasturiy vositalardan va texnologiyalardan hamda Internet texnologiyalaridan mukammal foydalanish kabilar katta ahamiyatga ega.

Ushbu fanning asosiy bo‘limlaridan hisoblangan algoritmlash va dasturlash bo‘limida talabalarga zamonaviy dasturlashtirish texnologiyalarini o‘rgatish, ulardan amaliyotda foydalanish ko‘nikmalarini mustahkamlashga alohida e‘tibor qaratilgan.

Kompyuterda dasturlash oxirgi yillarda juda tez rivojlanib, dastur tuzishga qiziquvchilar soni oshib bormoqda. Hozirgi vaqtda yuqori darajali dasturlash tillaridan Delphi, C++, Java kabi tillar Windows muhitida dasturlash imkonini beradi.

Borland C++ Builder 6 – Windows amaliyot tizimida dastur yaratishga imkon yaratadigan, obyektga yo‘naltirilgan dasturlash muhitidir. C++ da dastur tuzish zamonaviy vizual loyihalash texnologiyalariga asoslangan bo‘lib, unda dasturlashning obyektga yo‘naltirilgan g‘oyasi mujassamlashgan. C++ining oxirgi lahjalarida ma‘lumotlar bazasini yaratish va qayta ishlash, Internet tarmog‘idan foydalangan holda ma‘lumotlar alamashinuvini o‘rnatish, dasturlashning obyektga yo‘naltirilgan modelini keng qo‘llash, vizual dasturlashda yangi komponentalar kutubxonasini (VCL) yaratish kabi asosiy farqli imkoniyatlarni o‘z ichiga oladi.

1-amaliy mashg'ulot

Ilmiy-texnik masalalarni kompyuterda yechish bosqichlari.

Algoritmash aoslari.

Ishdan maqsad:

1. Ilmiy-texnik masalani yechish bosqichlari bilan tanishish.
2. Masala yechishning birinchi 3 ta bosqichini aniq misolda ko'rib chiqish.
3. Algoritmash, algoritm va uning xossalari, tavsiflash usullari, turlari bilan tanishish.

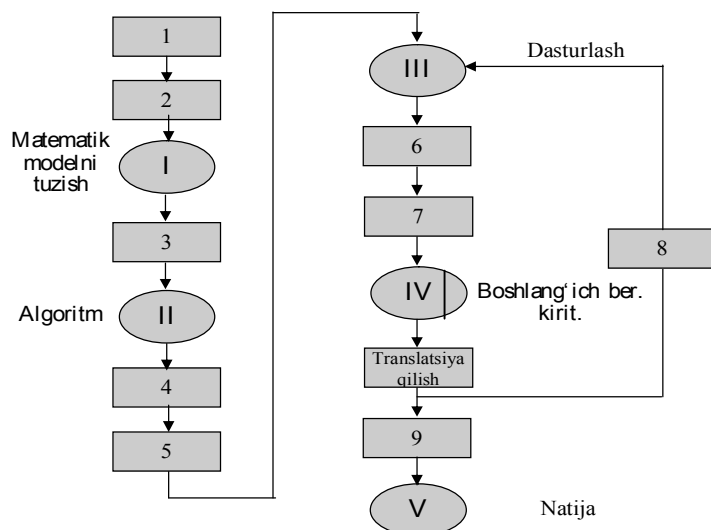
Nazariy qism

Kompyuterdan foydalanib masalani yechish, yaratilgan algoritmgaga asoslangan holda dastlabki ma'lumotlar ustida avtomatik tarzda amallar bajarilib, izlangan natija (natijalar) ko'rinishiga keltirish demakdir.

Kompyuterdan foydalanib ilmiy-texnik masalalarni yechish keng ma'nodagi ifoda bo'lib, quyidagi bosqichlardan iborat:

1. Masalaning qo'yilishi va maqsadning aniqlanishi.
2. Masalani matematik ifodalash.
3. Masalani yechish uslubini ishlab chiqish, sonli usullarni tanlash.
4. Masalani yechish algoritmini ishlab chiqish.
5. Ma'lumotlarni tayyorlash va tarkibini aniqlash (tanlash).
6. Dasturlash.
7. Dastur matnini va ma'lumotlarni axborot tashuvchiga o'tkazish.
8. Dastur xatolarini tuzatish.
9. Dasturning avtomatik tarzda kompyuterda bajarilishi.
10. Olingan natijalarni izohlash, tahlil qilish va dasturdan foydalanish uchun ko'rsatma yozish.

Yuqorida keltirilgan bosqichlarning grafik ko'rinishini 1.1-rasmdagi kabi tasvirlash mumkin.

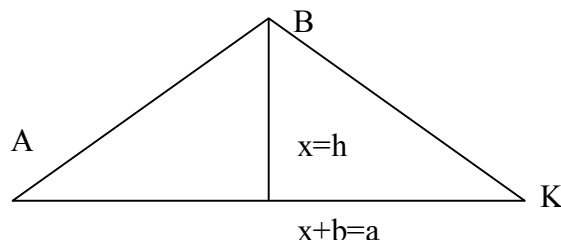


1.1-rasm. Masalalarni yechish bosqichlari

Masalani yechish uchun 3 ta bosqichni quyida keltirilgan aniq misolda ko'rib chiqamiz.

1. Masalaning qo'yilishi va maqsadni aniqlash.

Uchburchakning asosi - a balandligi - h dan b ga ko'p. Uchburchakning balandligini topish kerak ($x=h$)(1.2-rasm).



1.2-rasm. Masalaning qo'yilishining grafik ifodasi

2. Masalani matematik ifodalash.

Uchburchak yuzasi $c = \frac{1}{2}a \cdot h$ ga teng, bularni o'rniga qo'ysak,

$$c = \frac{1}{2}(x+b) \cdot x = \frac{x^2 + bx}{2}; \text{ yoki } 2c = x^2 + bx, \quad x^2 + bx - 2c = 0 \text{ bo'ladi.}$$

3. Masalani yechish usuli, ya'ni kvadrat tenglamani yechishga keltirildi.

4. Masalani yechishning algoritmlash bosqichi.

Algoritm deb, masalani yechish uchun bajarilishi lozim bo'lgan amallar ketma-ketligini aniq tavsiflaydigan qoidalar tizimiga aytiladi.

Algoritm quyidagi asosiy xossalarga ega:

Uzluqlilik, aniqlik, natijaviylik va ommaviylik.

Algoritmlarni ishlab chiqishda bir necha xil usullar bor. Shulardan uchta keng tarqalgan.

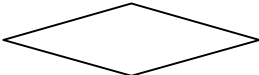
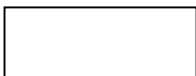
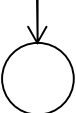
1. Algoritmni oddiy tilda tavsiflash.
2. Algoritmni sxema ko'rinishida ifodalash.
3. Algoritmni maxsus (algoritmik) tilda yozish.

Algoritmni ifodalashning eng keng tarqalgan shakli – bu oddiy tilda so'zlar bilan bayon qilishdir, lekin murakkab algoritmlarda ko'rgazmalilikni yaxshi ta'minlay olmaydi. Bundan tashqari, algoritmning so'zdagi tavsifi hisoblash mashinasiga kiritish uchun yaramaydi. Buning uchun algoritmni mashina tilida shunday bayon qilish kerak. Mashina tushunadigan shaklda yozilgan algoritm masalani yechish dasturidir.

Algoritm tizimi – bu berilgan algoritmni amalga oshirishdagi amallar ketma-ketligining oddiy tildagi tasvirlash elementlari bilan to'ldirilgan grafik tasvirlardir. Algoritmning har bir qadami biror-bir geometrik shakl-blok (blok simvol) bilan aks ettiriladi, bu GOST 19002-80da qat'iy belgilab berilgan.

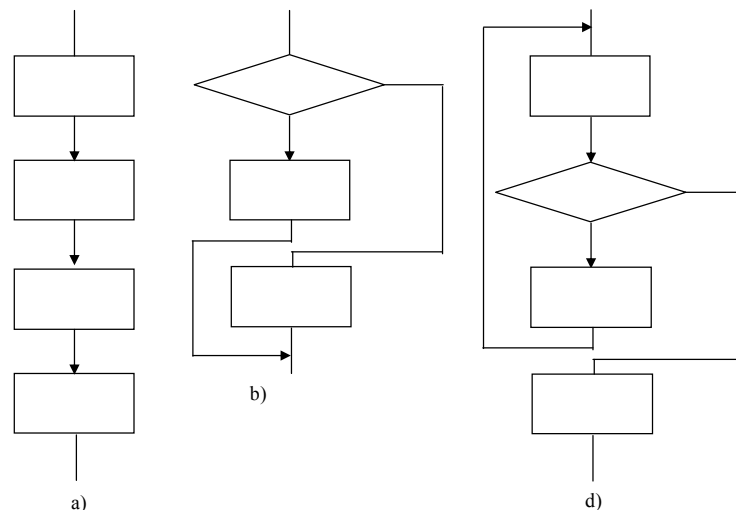
Algoritmnlarni tuzish simvollari 1-jadvalda keltirilgan.

1-jadval

Nomi	Belgilanishi	Bajaradigan vazifasi
Boshlash, Tamomlash		
Jarayon		Bir yoki bir nechta amallarni bajarish operatori
Qaror		Biron-bir shartga bogʻliq ravishda algoritm bajarilishi yoʻnalishini tanlash
Modifikatsiya		Dasturni oʻzgartiruvchi buyruq yoki buyruqlar turkumini oʻzgartirish amalini bajarish
Kiritish- chiqarish		Berilganlarni kompyuterga kiritish yoki olingan natijalarni tasvirlash
Qism- dastur		Avvaldan aniqlangan jarayonga murojaat
Bogʻlagich		Uzilib qolgan axborot oqimlarini ulash belgisi
Izoh		Va boshqalar

Murakkab masalalarning yechimini olish algoritmnlari yuqoridagi turlarining barchasini oʻz ichiga olishi mumkin.

Chiziqli turdagi algoritmnlarda bloklarning biri ketidan boshqasi joylashgan boʻlib, berilgan tartibda bajariladi (1.3-rasmga qarang).



1.3-rasm. a) chiziqli; b) tarmoqlanuvchi; d) takrorlanuvchi

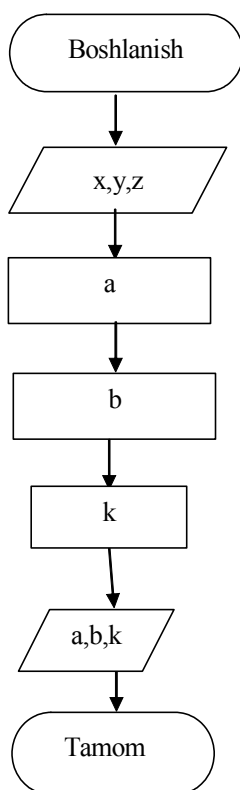
Misol.

$K = \sin a + \sin b$ ning qiymatini topish algoritm blok-sxemasi yaratilsin.

Bu yerda: $A = \ln(y - \sqrt{|x|}) (x - \frac{y}{2})$;

$B = \sin^2 \arctg z$; $x = -15,246$; $u = 4,642$; $Z = 20,001$;

Masalani yechish algoritmi 1.4-rasmida keltirilgan blok-sxema ko‘rinishida tuziladi.

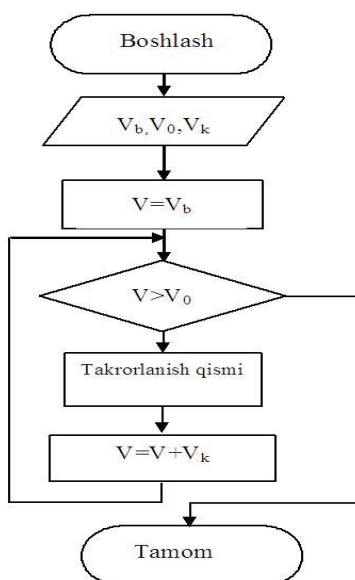


1.4-rasm. Masalani yechish blok-sxemasi

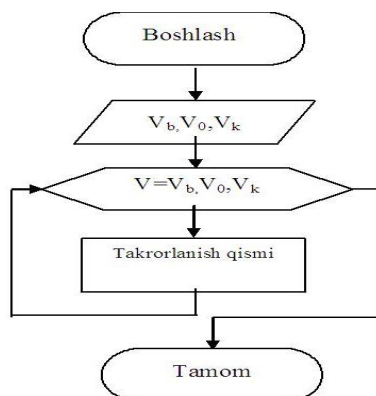
Amalda hamma masalalarni ham chiziqli turdagi algoritmgga keltirib yechib bo'lmaydi. Ko'p hollarda biron-bir oraliq natijaga bog'liq ravishda hisoblashlar u yoki boshqa ifodaga ko'ra amalga oshirilishi mumkin, ya'ni birorta mantiqiy shartning bajarilishiga bog'liq holda hisoblash jarayoni u yoki bu tarmoq bo'yicha amalga oshiriladi. Bunday tuzilishdagi hisoblash jarayonining algoritmi tarmoqlanuvchi turdagi algoritm deb ataladi.

Ko'pgina hollarda masalalarning yechimini olishda bitta matematik bog'lanishga ko'ra unga kiruvchi kattaliklarni turli qiymatlariga mos keladigan qiymatlarni ko'p martalab hisoblanadigan qismiga takrorlanishlar deb, takrorlanishlarni o'z ichiga olgan algoritmlar takrorlanuvchi turdagi algoritmlar deb ataladi.

Agar 5 va 6-rasmlardagi takrorlanuvchi algoritmlar blok-sxemalardagi V_b , V_o , V_k larni mos holda takrorlanish o'zgaruvchi-sining bosh, oxirgi va o'zgarish qadami desak, 1.5-rasmdagi algoritm tarmoqlanishdan foydalanib yaratilgan, 1.6-rasmdagi algoritm esa takrorlanishni boshlash bloki asosida yaratilgan takrorlanuvchi hisoblash jarayonlari algoritmiga misol bo'la oladi.



1.5-rasm. Tarmoqlanishdan foydalangan dastur blok-sxemasi



1.6-rasm. Takrorlanishdan foydalangan dastur blok-sxemasi

Topshiriqlar:

1. t- o'zgaruvchining qiymatini aniqlash algoritmini blok sxema ko'rinishida yarating.

$$t = \frac{2 \cos(x - \frac{\pi}{6})}{0.5 + \sin^2 y} (1 + \frac{z^2}{3 - z^2/5}),$$

bunda $x=14.26$, $y=-1.22$, $z=3.5 \times 10^{-2}$ $t=0.564849$.

2. u- o'zgaruvchining qiymatini aniqlash algoritmini blok sxema ko'rinishida yarating.

$$u = \frac{\sqrt[3]{8 + |x - y|^2 + 1}}{x^2 + y^2 + 1} - e^{|x-y|} (tg^2 z + 1)^x,$$

bunda $x=-4.5$, $y=0.75 \times 10^{-4}$, $z=0.845 \times 10^2$, $u=-55,6848$

3. c- o'zgaruvchining qiymatini aniqlash algoritmini blok sxema ko'rinishida yarating.

$$c = \begin{cases} f(x)^2 + y^2 + \sin(y), & x - y = 0 \\ (f(x) - y)^2 + \cos(y), & x - y > 0 \\ (y - f(x))^2 + tg(y), & x - y < 0. \end{cases}$$

4. d- o'zgaruvchining qiymatini aniqlash algoritmini blok sxema ko'rinishida yarating.

$$d = \begin{cases} (f(x) - y)^3 + arctg(f(x)), & x > y \\ (y - f(x))^3 + arctg(f(x)), & y > x \\ (y + f(x))^3 + 0.5, & y = x \end{cases}$$

5. S- o'zgaruvchining qiymatini aniqlash algoritmini blok sxema ko'rinishida

$$S = (1 + \frac{1}{1^2})(1 + \frac{1}{2^2}) + \dots (1 + \frac{1}{n^2})$$

6. Y- o'zgaruvchining qiymatini aniqlash algoritmini blok sxema ko'rinishida

$$Y = \sin x + \sin^2 x + \sin^3 x + \dots \sin^n x$$

Nazorat savollari:

1. Kompyuterdan foydalanib masalani yechish deganda nimani tushunasiz?
2. Masalani yechish bosqichlarini tushuntiring.
3. Algoritmizatsiyalash va algoritm ma'nosini tushuntiring.
4. Algoritm xossalari va uning asosiy turlari qanday?

2-amaliy mashg'ulot

Borland C++ Builder 6 integrallashgan muxiti bilan tanishish.

Oddiy jarayonlarni dasturlash.

Ishdan maqsad:

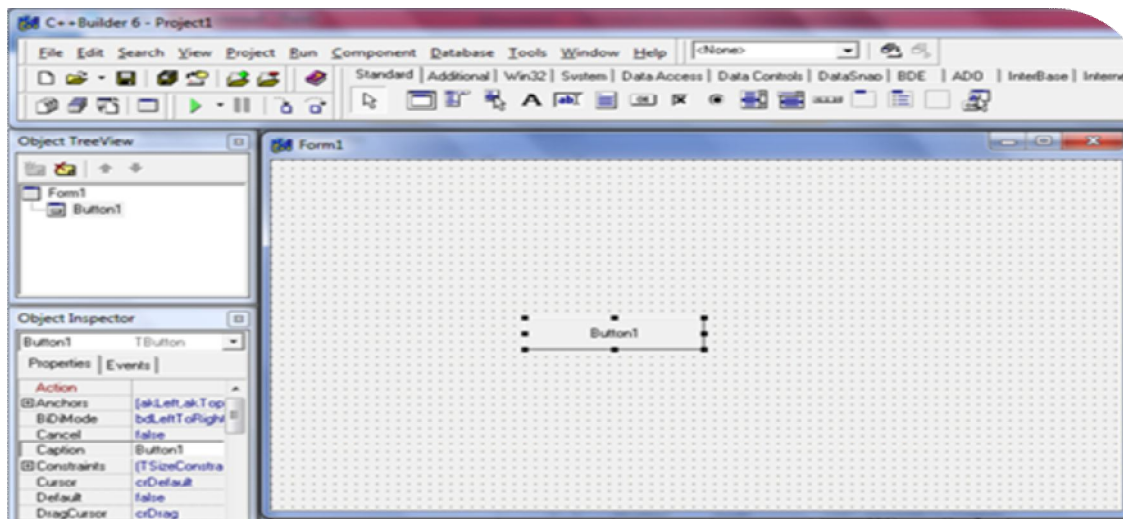
1. C++Builder 6 dasturlash muhitining asosiy elementlari bilan tanishish.
2. Chiziqli jarayonlarning yechish algoritmini tuzish.
3. Dastur tuzish asoslarini o'rganish.

Nazariy qism

C++Builder 6 muhitida dasturlash 2ta rejim orqali, konsol va visual rejimlarida amalga oshiriladi. Vizual rejimida dasturlash C++Builder 6 muhitining mahsus oynalari orqali amalga oshiriladi. 2.1-rasmda ko'rsatilganidek, C++Builder 6 muhitining oynasi WINDOWS AT tizimi oynasining uskunolari bilan mos keladi. C++Builder 6 muhiti tarkibiga asosan 5 ta oyna kiradi:

C++Builder 6 muhiti dasturini yuklash uchun quyidagi ketma-ketlikni amalga oshirish lozim: Пуск→Все программы→ C++Builder 6 muhiti→ Builder 6. Dastur yuklanganda ekranda bir vaqtning o'zida bir qancha oynalar vizual ravishda ochiladi va quyidagilardan iborat bo'ladi: 1) Bosh oyna (Project1) menyu qatori, uskunalar paneli, komponentalar paneli; 2) Forma oynasi (Forma 1); 3) Obyektlar daraxti oynasi; 4) Obyektlar inspektori (object inspector); 5) Dastur kodi oynasi (unit1).

Vizual rejimda dastur ko'rinishi 1 necha fayllardan iborat loyiha ko'rinishida tasvirlanadi (.cpp,.dfm,.h,.bpr va boshqalar).



2.1-rasm. C++Builder 6 muhitining ishchi oynasi.

C++Builder 6 tilining dasturlash ob'yektlari bo'lib quyidagilar xizmat qiladi: o'zgarmlar, o'zgaruvchilar, funktsiyalar, ifodalar va operatorlar. O'zgarmlar sonli, belgili, mantiqiy va qatorli bo'lishi mumkin.

Sonli o'zgarmlar:

56, -12, 526 - butun,
 0.43, -7.826, 0.2718e+1 - haqiqiy,
 's', 'G', '7', '!' - belgili,
 &&, ||, !! - mantiqiy,
 0, 1 - mantiqiy o'zgarmlar, yolg'on yoki rost.
 'Axborot texnologiyalari' - belgili o'zgarmlar.

O'zgaruvchilar:

I, g, x, Y- oddiy o'zgaruvchilar,
 A[i], d[I,j]- indeksli o'zgaruvchilar.

Har qanday o'zgaruvchi o'z nomiga ega (identifikator) va turi bilan harakterlanadi.

C++Builder 6 dasturlash tilining funktsiyalari.

Dasturlarda buyruqlar tarkibida turli ko'rinishdagi standart funktsiyalardan foydalanuvchi funktsiyalaridan foydalanish mumkin. Standart funktsiyalar maxsus jadvallarda beriladi. Bu funktsiyalar turli turdagi standart funktsiyalar kutubxonasini tashkil qiladi. Standart funktsiyalardan farqli foydalanuvchi funktsiyalari dasturda foydalanuvchi tomonidan berilib, faqat maxsus misollarda ishlatiladi.

Qiyida standart funktsiyalarni keltiramiz:

Standart funktsiyalar

$(1+x)^2$	pow(1+x,2)
Sinx	sin(x)
Cosx	cos(x)
Tgx	tan(x) sin(x)/ cos(x)

$\text{Ln}x$	$\log(x)$
e^x	$\exp(x)$
x^2	$x*x$
$ x $	$\text{abs}(x)$
$A^x e^{x \ln a}$	$\text{pow}(a, \exp(x * \log(a)))$
$\sqrt{x}$	$\text{sqrt}(x)$
$\cos^2 x^3$	$\text{pow}(\cos(\text{pow}(x, 3)), 2)$
X^4	$\text{pow}(x, 4)$

Ifodalar

C++Builder 6 tilida ifodalar bir-biri bilan o‘zaro bog‘langan operandlar birikmasidan iborat amallar bilan bog‘langan. Har bir qadam ifodaning turiga mos ravishda o‘zgarmas, o‘zgaruvchi yoki funksiya orqali ifodalanadi. Ifodalar sonli, belgili, mantiqiy, qatorli bo‘lishi mumkin. Arifmetik, mantiqiy va nisbiy amallar ishlatiladi.

$(x*x)+\sin(a-x)-1$ - sonli ifodalar.

$x \ \&\& \ y \ || \ ! \ z$ - mantiqiy ifodalar.

Operatorlar

C++Builder 6 tili operatorlari oddiy va strukturlashgan turlariga bo‘linadi:

- oddiy: o‘zlashtirish, o‘tish, kirish-chiqish operatorlari;
- strukturlashgan: shartli, tanlash, takrorlanish operatorlari.

Oddiy jarayonlar chiziqli strukturali algoritmgaga ega bo‘lib, ularda asosan ma‘lumotlarning oddiy toifalari, standart matematik va o‘zgartirish funktsiyalari va oddiy operatorlar ishlatiladi. C++ tili operatorlari ichida faqat o‘zlashtirish operatori o‘zgaruvchilar qiymatini o‘zgartira olish imkoniga ega.

$$Y=a*x+b;$$

Chiziqli jarayonda dasturlashga misol.

1-misol. A,B nuqtalari orasidagi masofani toping. X1,Y1 va X2,Y2 koordinatalari berilgan.

```
//-----
#include<iostream.h>    // 1
#include<conio.h>        // 2
#include <vcl.h>          // 3
#include<math.h>        // 4
#pragma hdrstop
//-----
#pragma argsused
int main(int argc, char* argv[])
{void main(); // tuzilayotgan dastur
```

```
float x1=4.6, x2=6.9, y1=7, y2=2.5, r;
r=pow(pow(x1-x2,2)+pow(y1-y2,2),2);
cout<<"r="<<r<<endl;
getch();    return 0;
}
```

//-----

Birinchi to'rt qatorda protsessor ko'rsatmalari yozilgan. **#include <fayl nomi>** ko'rinishidagi ko'rsatma fayldagi ma'lumotni faylning kerakli nuqtasiga qo'yadi (bu e'lon fayli deyiladi). Birinchi qatorda ko'rsatma **<iostream>** faylining ishtirokini ta'minlaydi. Bunda kirish-chiqish kutubxonasining standart elementlari haqidagi ma'lumotlar joylashgan. Ikkinchi qatorda ko'rsatma **<conio>** faylining ishtirokini ta'minlaydi. Uchinchi qatorda ko'rsatma **<vcl>** faylining ishtirokini ta'minlaydi. Bunda dasturda qatnashuvchi elementlarning standart kutubxonasidan olingan yozuvlarni ifodalaydi. To'rtinchi qatorda ko'rsatma **<math>**, faylining ishtirokini ta'minlaydi. Bu faylda dasturda ishlatiladigan standart funktsiyalarni saqlanadi.

2-misol. Quyidagi misolda a o'zgaruvchining b- darajasini topish dasturi konsol rejimda tuzilgan.

//-----

```
#include<math.h>
#include<conio.h>
#include <vcl.h>
#pragma hdrstop
//-----
#pragma argsused
int main(int argc, char* argv[])
{int a,b,c;
  cout<<"a=";
  cin>>a;
  cout<<"b=";
  cin>>b;
  c=pow(a,b);
  cout<<"c="<<c;
  getch();
  return 0;
}
```

//-----

Dastur kodi va uning bajarilishidan olingan natija(2.2-rasm):

```

Unit1.cpp
#include <math.h>
#include <vc1.h>
#pragma hdrstop

//-----

#pragma argsused
int main(int argc, char* argv[])
{
    int a,b,c;
    cout<<"a=";
    cin>>a;
    cout<<"b=";
    cin>>b;
    c=pow(a,b);
    cout<<"c="<<c;
    getch();

    return 0;
}
//-----

```

Output: a=4, b=5, c=1024_

2.2-rasm. Tuzilgan dastur natijasi

Topshiriqlar:

1. Quyidagi sonlarni C++Builder 6 tilida yozing:

5!	
LXIV	
6,37	
0,7(4)	
11/4	
-1/6	
II	
$5 \cdot 10^6$	
10^6	

2. Quyidagi ifodalarni C++Builder 6 tilida ifodalang:

$(1+x)^2$; $\sqrt{1+x^2}$; $|a+bx|$; $\sin 8$; $\cos^2 x^3$; $\operatorname{tg} x$;

$\log_2 \frac{x}{5}$; $\operatorname{ch} x$; $\operatorname{arctg} 10^3$; $\arcsin x$.

3. O'zlashtirish operatorlari ko'rinishida yozing.

$$Y = 8\sqrt{x^8} + 8^x; \quad Y = \frac{xyz - 3/3|x + 4\sqrt{y}|}{10^7 + \sqrt{\lg 4!}};$$

$$Y = \frac{\beta + \sin^2 \pi^4}{\cos 2 + |\operatorname{ctg} \lambda|}.$$

4. 3-misoldagi operatorlarni ishlatgan holda dastur tuzing.

5. $(1+x)^2 / \sqrt{1+x^2}$; $|a^2 + bx^2|$; $\sin 8$; $\cos^2 x^3$; $\operatorname{tg} x$; $\operatorname{arctg} 10^3$; $\arccos x$.

6. Quyidagi funktsiyani hisoblash dasturi tuzilsin:

$$a = \lg(\sqrt{e^{x-y}} + x^{|y|} + z), \quad b = e^{|x-y|}(\operatorname{tg}^2 x + 1)^x, \quad k = e^b \sin a,$$

bu yerda $x = 1,542$, $y = 0.75$, $z = 3.5$

Nazorat savollari:

1. C++Builder 6 dasturlash muhitining asosiy elementlari nimalardan tashkil topgan?
2. Standart funktsiyalar bilan foydalanuvchi tomonidan beriladigan funktsiyalarning farqi qanday?
3. C++Builder 6 tilining ifodalari qanday va ularni dasturda ishlatish tartibini ko'rsating?
4. C++Builder 6 tilining asosiy bo'limlari vazifalari qanday?
5. C++Builder 6 tilida oddiy dasturlashda qanday operatorlar ishlatiladi?

3-amaliy mashg'ulot

Tarmoqlanuvchi algoritmlarni dasturlash

Ishdan maqsad:

1. Nazariy qism bilan tanishish.
2. Shartli o'tish operatorlarini o'rganish.
3. Shartli o'tish operatorlari yordamida tarmoqlanuvchi jarayonlarni dasturlash ko'nikmalarini hosil qilish.

Nazariy qism

Amaliyotda shunday jarayonlar mavjudki, buyruqning ijrosi ma'lum bir shartning bajarilishiga bog'liq bo'ladi. Bunday jarayonlar tarmoqlanuvchi jarayonlar deb ataladi. Bu jarayonlarni dasturlash uchun shartli o'tish va tanlash operatorlari foydalaniladi. Bu algoritmnining blok-sxemasini biz yuqorida ko'rib o'tdik.

Shartli o'tish operatorining to'liq ko'rinishi quyidagicha ifodalanadi:

if (Ifoda) operator_S1; [else operator_S2;]

bu yerda **if**, **else** – bo'g'lovchi so'zlar, **S1, S2**- C++Builder 6 tilining operatorlari.

Bu yerda ifoda mantiqiy shartni ifoda etib, uning qiymati algoritm yo'nalishini aniqlab beradi. Mantiqiy ifoda oddiy va murakkab bo'lishi mumkin. U o'zida mantiqiy ifodalar (**!!**, **&&**, **||**)dan tashqari munosabat operatsiyalarini ham o'z ichiga olishi mumkin. Agar ifoda rost 1 (True) qiymatga ega bo'lsa, S1 aks holda yolg'on 0 (False) S2 operatorlari bajariladi.

Avvalo arifmetik turga ega bo'lgan ifoda hisoblanadi. Agar u 0 ga teng bo'lmasa, birinchi operator bajariladi, aks holda ikkinchi operator bajariladi. Shundan keyin

boshqaruv If operatoridan keyingi operatorga beriladi. Bunda operator bitta bo‘lishi ham mumkin.

Agar biror bir tarmoqda bir necha operatorlarning ishlashi zarur bo‘lib qolsa, u holda ularni blok ichida ifodalash zarur. Blok hoxlagan operatorlarni o‘z ichiga olishi mumkin. Bundan tashqari izoh va o‘tish operatorlarini ham o‘z ichiga oladi.

Misollar:

```
if (a<0) b = 1; // 1
if (a<b && (a>d || a==0)) b++;
                        else {b= a; a = 0;} // 2
if (a<b) {if (a<c) m = a; else m = c;}
else {if (b<c) m = b; else m = c;} // 3
```

Birinchi misolda else tarmog‘i yo‘q, shartning bajarilishiga qarab o‘zlashtirish operatori yoki ishlaydi yoki ishlamaydi.

Agar bir nechta shartni tekshirish lozim bo‘lib qolsa, ularni mantiqiy operatsiyalar belgilari bilan birlashtiriladi. Misol uchun, ikkinchi misoldagi **mantiqiy ifodasi** rost bo‘ladi, qachonki bir vaqtning o‘zida ham $a < b$ sharti, ham qavs ichidagi shartlardan biri bajarilsa.

Uchinchi misolda 3ta o‘zgarmasdan eng kichik qiymat topiladi. Bu yerda figurali qavsni qo‘yish shart emas.

Switch operatori (tanlash) tarmoqlanishni bir necha yo‘nalishga uzatish uchun ishlatiladi.

Operatorning ko‘rinishi quyidagicha:

```
switch (ifoda ){
case o‘zgarmas ifoda_1: 1-operator; break;
case o‘zgarmas ifoda_2: 2-operator ; break;
...
case o‘zgarmas ifoda_n: n-operator ; break;
[default: operatorlar ;}]
```

Operatorning bajarilishi ifodaning hisoblanishi bilan boshlanadi (u butun son bo‘lishi shart), undan so‘ng boshqaruv operatorlarga uzatiladi, qachonki belgilangan o‘zgarmas ifodaning qiymati hisoblanayotgan qiymat bilan mos tushsa. Shundan keyin, agar o‘gartiruvchidan chiqish ko‘rsatilmagan bo‘lsa tarmoqning hamma yonalishi ketma-ket bajariladi.

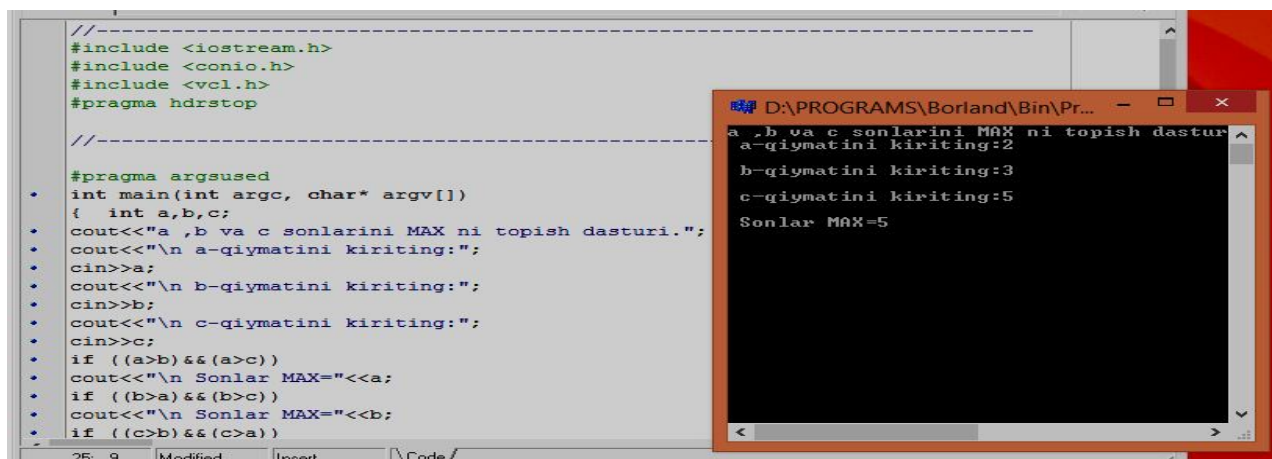
Hamma o‘zgarmas ifodalar har xil qiymatga ega bo‘lishlari shart, lekin ular butun turda bo‘lishlari kerak. Agar solishtirish amalga oshmasa, u holda **default** so‘zidan so‘ng turgan operatorlar bajariladi (default so‘zi bo‘lmasa boshqaruv **switch** dan so‘ng turgan operatorga uzatiladi).

Chiqish operatori **break** case tarmog‘idan qaytish uchun ishlatiladi. **Break, continue** operatorlari C++ tilida o‘tish operatorlari deb yuritiladi.

1-misol. Quyidagi misolda X, Y, va Z sonlarini maksimalini topish dasturi tuzilsin.

Quyida dastur kodi va olingan natija keltirilgan(3.1-rasm):

```
#include<math.h>
#include<conio.h>
#include <vcl.h>
#pragma hdrstop
//-----
#pragma argsused
int main(int argc, char* argv[])
{
    int a,b,c;
    cout<<"a ,b va c sonlarini MAX ni topish dasturi.";
    cout<<"\n a-qiymatini kiriting:";
    cin>>a;
    cout<<"\n b-qiymatini kiriting:";
    cin>>b;
    cout<<"\n c-qiymatini kiriting:";
    cin>>c;
    if ((a>b)&&(a>c))
        cout<<"\n Sonlar MAX="<<a;
    if ((b>a)&&(b>c))
        cout<<"\n Sonlar MAX="<<b;
    if ((c>b)&&(c>a))
        cout<<"\n Sonlar MAX="<<c;
    getch(); return 0;}
```



3.1-rasm. Tuzilgan dastur natijasi quyidagicha

2-misol. Dekabr oyi hafta kunlarini chiqarish dasturi tuzilsin:

```
//-----
#include<iostream.h>
#include<math.h>
#include<conio.h>
#include <vcl.h>
#pragma hdrstop
```

```
//-----
#pragma argsused
int main(int argc, char* argv[])
{
switch(hafta kunlari){
case 1:cout<<"chorshanba \n"; break;
case 2:cout<<"payshanba \n"; break;
case 3:cout<<"juma \n";break;
case 4:cout<<"shanba \n"; break;
case 5:cout<<"yakshanba \n"; break;
case 6:cout<<"dushanba \n"; break;
case 7:cout<<"seshanba \n"; break;
case 8:cout<<"chorshanba \n"; break;
case 9:cout<<"payshanba \n"; break;
case 10:cout<<"juma \n"; break;
case 11: cout<<"shanba \n"; break;
case 12: cout<<"yakshanba \n"; break;
case 13: cout<<"dushanba \n"; break;
case 14:cout<<"seshanba \n"; break;
case 15:cout<<"chorshanba \n"; break;
case 16:cout<<"payshanba \n"; break;
case 17:cout<<"juma \n"; break;
case 18:cout<<"shanba \n"; break;
case19: cout<<"yakshanba \n"; break;
case 20: cout<<"dushanba \n"; break;
case 21: cout<<"seshanba \n"; break;
case 22: cout<<"chorshanba \n"; break;
case 23:cout<<"payshanba \n"; break;
case 24: cout<<"juma \n"; break;
case 25:cout<<"shanba \n"; break;
case 26: cout<<"yakshanba \n"; break;
case 27: cout<<"dushanba \n"; break;
case 28: cout<<"seshanba \n"; break;
case 29: cout<<"chorshanba \n"; break;
case 30: cout<<"payshanba \n"; break;
case 31: cout<<"juma \n"; break;
default: cout;
getch();
return 0;}

```

Quyida dastur kodi va olingan natija keltirilgan(3.1-rasm):

The screenshot shows a C++ IDE with a file named 'Unit1.cpp'. The code is a switch statement that takes an input 'D' and prints corresponding day names. The console window shows the output for 'D=8', which is 'chorshanba'.

```

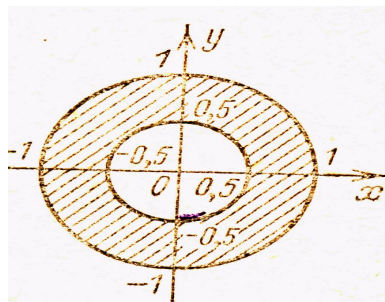
Unit1.cpp
#pragma argsused
using namespace std;
int main()
{ unsigned int D;cout<<" D=";<<cin>>D;
switch(D){
case 1:cout<<"chorshanba \n"; break;
case 2:cout<<"payshanba \n"; break;
case 3:cout<<"juma \n";break;
case 4:cout<<"shanba \n"; break;
case 5:cout<<"yakshanba \n"; break;
case 6:cout<<"dushanba \n"; break;
case 7:cout<<"seshanba \n"; break;
case 8:cout<<"chorshanba \n"; break;
case 9:cout<<"payshanba \n"; break;
case 10:cout<<"juma \n"; break;
case 11: cout<<"shanba \n"; break;
case 12: cout<<"yakshanba \n"; break;
case 13: cout<<"dushanba \n"; break;
case 14:cout<<"seshanba \n"; break;
case 15:cout<<"chorshanba \n"; break;
case 16:cout<<"payshanba \n"; break;
case 17:cout<<"juma \n"; break;
case 18:cout<<"shanba \n"; break;
}
}
15: 38 Modified Insert \Code/
D:\PROGRAMS\Borland\Bin\Project2.exe
D=8
chorshanba

```

3.2-rasm. Tuzilgan dastur natijasi.

Topshiriqlar:

1. Koordinata o'qida X va Y nuqtalarining shtrixlangan maydonga tegishli ekanligini aniqlang.



2. Berilgan X,Y,Z sonlarining minimal va maksimal qiymatlarining yig'indisini topish dasturi tuzilsin.

3. Yil oylarining tartib raqamiga asosan oy nomlarini bosmaga bosib chiqarish dasturi tuzilsin. Misol uchun, 3 sonini kiritsak «Mart» so'zi chiqsin.

4. Funksiya qiymatini topish dasturi konsol rejimda tuzilsin:

$$a = \begin{cases} (f(x) + y)^2 - \sqrt{f(x)y}, & xy > 0 \\ (f(x) + y)^2 + \sqrt{|f(x)y|}, & xy < 0 \\ (f(x) + y)^2 + 1, & xy = 0 \end{cases}$$

5. Funksiya qiymatini topish dasturi konsol rejimda tuzilsin:

$$b = \begin{cases} \ln(f(x)) - (f(x)^2 + y)^3, & x/y > 0 \\ \ln|f(x)/y| + (f(x) + y)^3, & x/y < 0 \\ (f(x)^2 + y)^3, & x = 0 \\ 0, & y = 0 \end{cases}$$

6. Funksiya qiymatini topish dasturi konsol rejimda tuzilsin:

$$c = \begin{cases} f(x)^2 + y^2 + \sin(y), x - y = 0 \\ (f(x) - y)^2 + \cos(y), x - y > 0 \\ (y - f(x))^2 + \lg(y), x - y < 0. \end{cases}$$

Nazorat savollari:

1. C++ Builder tilida necha turdagi o'tish operatorlari mavjud?
2. C++ Builder dasturlash tilida qanday jarayonlarga tarmoqlanuvchi jarayonlar deyiladi?
3. Qanday tanlash operatorlari mavjud va ular qachon ishlatiladi?
4. Shartli o'tish operatorining to'liq ko'rinishi qanday yoziladi?

4-amaliy mashg'ulot Takrorlanuvchi algoritmlarni dasturlash.

Ishdan maqsad:

1. Nazariy qism bilan tanishish.
2. Takrorlanuvchi tarkibli algortmlar dasturini tuzishni o'rganish.
3. Takrorlanish operatorlaridan foydalanish ko'nikmalarini egallash.

Nazariy qism

Ayrim masalalarda bir yoki bir necha parametrlarning o'zgarishiga qarab ma'lum hisoblashlar bir necha marotaba takrorlanib bajarilishi mumkin.

C++Builder 6 tilida takrorlanishni asosan 3 xil ko'rinishda tashkil etish mumkin: parametrli, oldshartli (takrorlanish sharti oldindan qo'yilgan) va ketshartli (takrorlanish sharti keyin qo'yilgan) takrorlanish. Bu turdagi takrorlanuvchi jarayonlarni dasturlashda mos ravishda **For**, **While Do**, **Do While** operatorlari ishlatiladi.

Har qanday takrorlanish *takrorlanish tanasidan* tashkil topadi, ya'ni bir necha marta bajariladigan operatorlardan, boshlang'ich ko'rsatmalardan, takrorlanish parametri modifikasiyasidan va takrorlanish shartini tekshirib takrorlanishdan chiqish blokidan tashkil topgan bo'ladi. **For** operatori dastur tarkibida takrorlanish soni ma'lum bo'lgan takrorlanish jarayonlarini tashkil qilish uchun ishlatiladi.

While Do, **Do While** operatorlari takrorlanish soni noma'lum bo'lgan holda takrorlanishni tashkil etish uchun xizmat qiladi, yoki operator takrorlanish soni ma'lum bo'lmagan takrorlanish jarayonida ishlatiladi.

Takrorlanishni bir marta bajarilishi *iteratsiya* deb ataladi. Takrorlanish parametrlari deb, takrorlanishdan chiqish shartini tekshiradigan o'zgaruvchilarga aytiladi. Har bir iteratsiyada ishlatiladigan butun sonli takrorlanish parametrlari, *takrorlanish hisoblaguvchisi* deb ataladi.

Tashqaridan takrorlanish tanasiga boshqaruvni uzatish mumkin emas. Takrorlanishdan chiqish, faqat chiqish shartlarini bajargan holda amalga oshirilishi mumkin. Bu **break**, **return** yoki shartsiz o'tish operatorlariga ham tegishli.

Old **shartli takrorlanish (while)** quyidagi ko'rinishga ega:

while (ifoda) operator;

while (ifoda) {tashkil qiluvchi operator};

Oddiy yoki murakkab shartlardan tashkil topgan mantiqiy ifoda takrorlanish shartini tashkil etadi. Agar ifoda 0 (rost) ga teng bo'lmasa, u holda takrorlanish operatori bajariladi va yana ifoda hisoblanadi. Agar birinchi tekshiruvda ifoda 0 ga teng bo'lsa, u holda takrorlanish tanasi bir marta ham bajarilmaydi.

Ket shartli takrorlanish (do while) quyidagi ko'rinishga ega:

do operator while ifoda;

yoki

do { tashkil qiluvchi operator } while ifoda;

Takrorlanish tanasini tashkil etuvchi avval oddiy yoki murakkab operator, keyin ifoda hisoblanadi. Agar u 0 ga teng bo'lmasa (rost), takrorlanish tanasi yana bir marta bajariladi va hokazo, qachonki ifoda 0 ga teng bo'lguncha yoki takrorlanish tanasida boshqaruvni uzatuvchi biron - bir operator bajarilmaguncha takrorlanish davom etadi.

Parametrli takrorlanish (for) quyidagi ko'rinishga ega:

for (initsializatsiya; ifoda; modifikatsiyalar) operator;

Initsializatsiya, takrorlanishda qatnashadigan kattaliklar va qiymqtlarni e'lon qilish, boshlang'ich qiymatni o'zlashtirish uchun ishlatiladi. Bu qismda vergul bilan ajratilgan bir nechta operatorlarni yozish mumkin.

Ifoda takrorlanish tanasi shartini aniqlaydi, agar ifoda 0 ga teng bo'lmasa (rost), takrorlanish bajariladi.

Modifikatsiyalar har bir takrorlanish iteratsiyasidan keyin bajarilib, odatda u takrorlanish parametrlarini o'zgartirish uchun xizmat qiladi. Modifikatsiya bo'limida bir necha operatorlarni vergul orqali yozish mumkin.

Oddiy yoki murakkab operatorlar takrorlanish tanasini tashkil etadi, misol uchun:

for (int i = 1, s = 0; i<=100; i++) s += i; // sonlar yig'indisi 1 dan 100 gacha.

1-misol. Quyidagi misolda a va b sonlarining Ekubini topish dasturi tuzilsin:

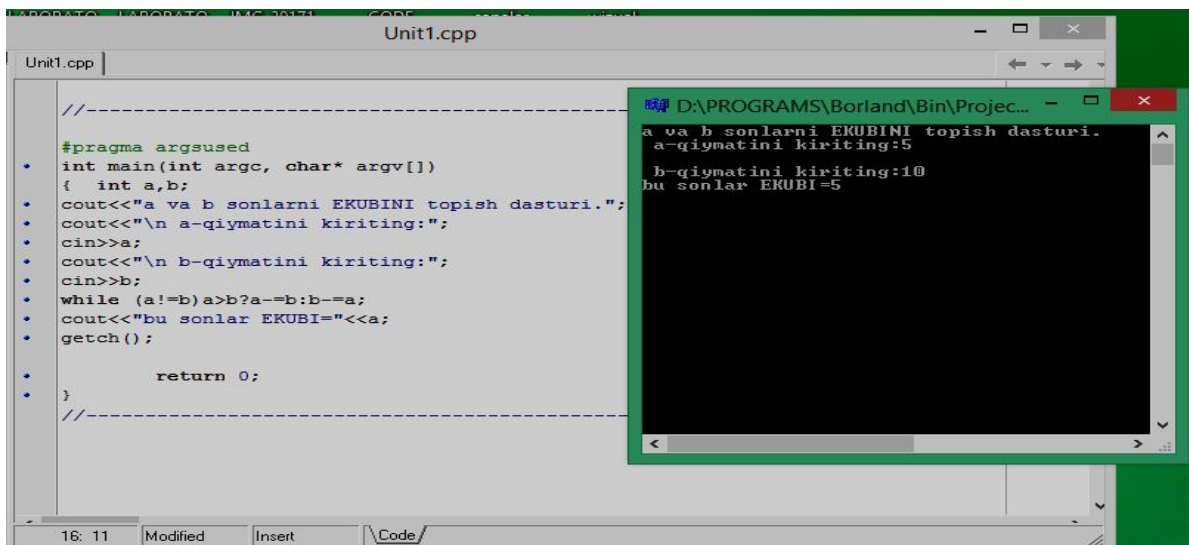
```
/-----  
#include<iostream.h>  
#include<math.h>
```

```

#include<conio.h>
#include <vcl.h>
#pragma hdrstop
//-----
#pragma argsused
int main(int argc, char* argv[])
{
int a,b;
cout<<"SONLAR EKUBINI TOPISH DASTURI";
cout<<"\n a-qiymatini kiriting:";
cin>>a;
cout<<"\n b-qiymatini kiriting:";
cin>>b;
while (a!=b)a>b?a-=b:b-=a;
cout<<"\n BU SONLAR EKUBI="<<a;
getch(); return 0;}
//-----

```

Quyida dastur kodi va olingan natija keltirilgan(4.1-rasm):



4.1-rasm. Tuzilgan dastur natijasi.

2-misol. Hisoblang
$$S=1+\frac{x}{1}+\frac{x^2}{2}+\dots+\frac{x^n}{n}$$

```

#include <math.h>
#include<iostream.h>
#include<conio.h>
#include <vcl.h>
#pragma hdrstop
//-----

```

```
#pragma argsused
int main(int argc, char* argv[])
{ float s,x; int i,n;
  cout<<"n="; cin>>n;
  cout<<"x="; cin>>x;
  s=1;
  while (i<=n) {s=s+pow(x,i)/i; i+=i;}
  cout<<"s= "<<s<<endl;
  getch();    return 0; }
//-----
```

Topshiriqlar:

1. Funksiyani hisoblash dasturi tuzilsin: $S = (1 + \frac{1}{1^2})(1 + \frac{1}{2^2}) + \dots (1 + \frac{1}{n^2})$
2. Funksiyani hisoblash dasturi tuzilsin: $Y = \sin x + \sin^2 x + \sin^3 x + \dots \sin^n x$
3. Funksiyani hisoblash dasturi tuzilsin: $F = n!$
4. Funksiyani hisoblash dasturi tuzilsin:

$$Y = \begin{cases} \sum_{i=1}^{50} \frac{x_i}{2^i} & x \geq 0 \\ \prod_{i=1}^{50} \frac{i^2 x^2}{i+2} & x < 0 \end{cases}$$
5. 'A' dan..... 'Z' gacha bo'lgan qator harflarining tartib raqami yig'indisi topish dasturi tuzilsin.
6. 1 dan 100 gacha bo'lgan hamma toq natural sonlar bosmaga chiqarish dasturi tuzilsin

Nazorat savollari:

1. C++Builder 6 tilida qanday jarayonlarga takrorlanish jarayoni deyiladi?
2. Necha turdagi takrorlanish jarayonlarining operatorlari mavjud?
3. Parametrli takrorlanish operatori qachon ishlatiladi?
4. Nima uchun oldshartli va ket shartli takrorlanish operatorlari deyiladi?
5. Takrorlanish ichida takrorlanish nima?

5-amaliy mashg'ulot

Ma'lumotlarning strukturlashgan toifalari. Massivlardan foydalangan holda dasturlash.

Ishdan maqsad:

1. Nazariy qism bilan tanishish.

2. Muntazam toifadagi ma'lumotlarni dasturlarda ishlatish usullari bilan tanishish.
3. Massivlar yordamida dasturlar tuzishni o'rganish.

Nazariy qism

Massiv bu – bir xil toifadagi elementlarning tartiblangan to'plamidir. Massiv ma'lum nomga ega bo'lib, massivning har bir elementi shu nom bilan nomlanadi va u o'z tartib raqamiga (indeksiga) ega. Massiv elementlari xotirada ketme-ket joylashib, ularga murojaat massiv nomi va kavadrat qavs ichida indeksni ko'rsatish orqali amalga oshiriladi. Massivlar bir o'lchamli yoki ko'p o'lchamli bo'lishi mumkin.

Bir o'lchamli massivni tavsiflashda massiv nomidan so'ng kvadrat qavs ichida uning o'lchami ko'rsatiladi:

int a[10]; // 10 ta butun sonda iborat massiv
float b[3]; // 3 ta xaqiqiy sonda iborat massiv

Massivning qiymatlarini uni tavsiflash jarayonida berish mumkin:

int a[3] = {1, 2, 3};

Ikki o'lchamli massivlar quyidgi ko'rinishda e'lon qilinadi:

int matrix[3][5]; // 3 ta qator, 5 ta ustundan tashkil topgan butun sonlardan iborat massiv;
int array2d[5][5];

1-misol. N ta elementdan tashkil topgan massivning eng katta va eng kichik elementlari yig'indisini topish dasturi tuzilsin.

```
//-----
#include<iostream.h>
#include<conio.h>
#include <vcl.h>
#pragma hdrstop
//-----
#pragma argsused
int main(int argc, char* argv[])
{
float min,max,s=0;
for(i=0;i<n;i++)
float a,b,d,x[100];  {
while(1)
s+=x[i];

{
printf("\n n=");
if(max<x[i]) max=x[i];
```



```

scanf("%d",&n);
if(n>0 && n<=100) break;
if(min>x[i]) min=x[i];

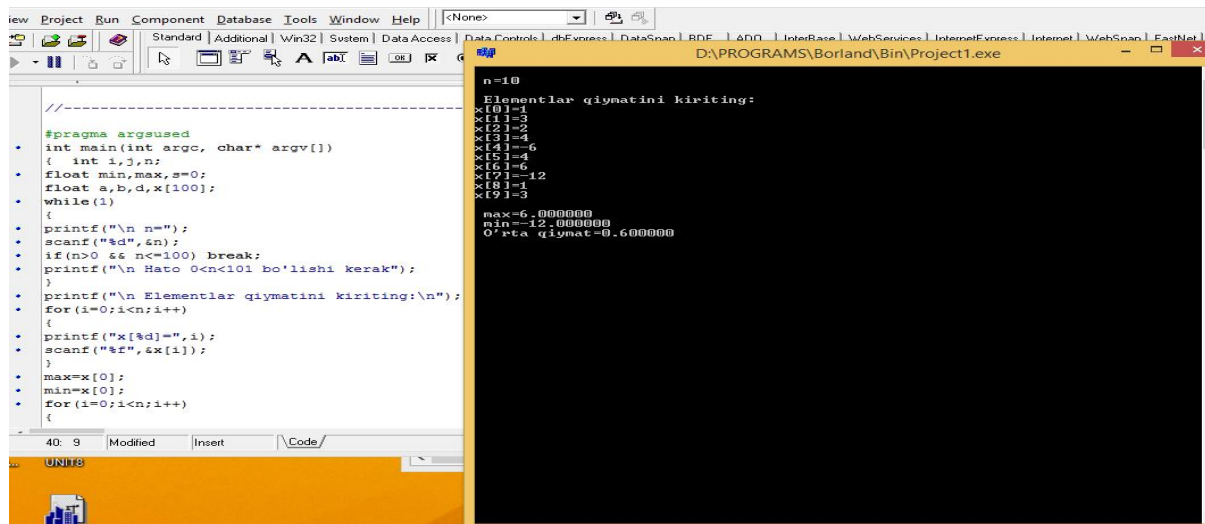
printf("\n Hato 0<n<101 bo'lishi kerak");
}

}
s/=n;
printf("\n Elementlar qiymatini kiriting:\n");
printf("\n max=%f",max);
for(i=0;i<n;i++)
printf("\n min=%f",min);
{printf("\n O'rta qiymat=%f",s);
printf("x[%d]=",i);
getch();
scanf("%f",&x[i]);
}
max=x[0];
min=x[0]

return 0;}
//-----

```

Quyida dastur kodi va olingan natija keltirilgan(5.1-rasm):



5.1-rasm. Tuzilgan dastur natijasi.

2-misol. $B(n,n)$ - ikki o'lchamli massiv berilgan. Agar $b=3$ bo'lsa massiv quyidagi ko'rinishda bo'ladi

$$\begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix}.$$

Massivning izi (Sl) va elementlari yig'indisi (St) topilsin.

Matritsa izi bu asosiy diagonaldagi elementlar yig'indisidir.

```
//-----
#include<iostream.h>
#include<conio.h>
#include <vcl.h>
#pragma hdrstop
//-----
#pragma argsused
int main(int argc, char* argv[])
{ const k=20;
float b[k][k]; float sl=0, st=0;
int n,m,i,j;
cout<<"satrlar soni "; cin>>n;
cout<<"ustunlar soni "; cin>>m;
for (i=0; i<n; i++)
for (j=0; j<m; j++) cin>>b[i][j];
sl+=b[0][0];
for (i=1; i<n; i++)
for (j=1; j<m; j++) if (i==j) sl += b[i][j];
for (i=0; i<n; i++)
for (j=0; j<m; j++) if (i<=j) st=st+b[i][j];
cout<<"matritsa izi Sl="<<sl<<endl;
cout<<"elementlar yigindisi St= "<<st;
getch(); return 0;
}
//-----
```

Forma ilovasida massivlarni ishlatish

Forma ilovasida bir o'lchamli va ko'p o'lchamli massivlar bilan ishlash uchun bir nechta komponentalari mo'jallangan: ListBox, ComboBox, StringGrid komponentalari.

ListBox va ComboBox komponentalari bir o'lchamli massiv elementlarini Formaga kiritish va chiqarish uchun foydalaniladi, StringGrid komponentasi ikki o'lchamli massivlar bilan ishlashda qo'llaniladi, massivlar jadval ko'rinishida kiritiladi va chiqariladi.

StringGrid komponentasi

StringGrid jadval komponentasi ikki o'lchovli matritsa elementlari qiymatini ekranda jadval ko'rinishda tasvirlash, ular qiymatini kiritish va tahrirlash uchun ishlatiladi. Standart holda jadval qator va ustun raqamlari noldan boshlanadi. Jadval ustun va qatorlar sonini uning xossasi yordamida o'rnatiladi. *StringGrid* – jadval komponentasining asosiy xossalari:

ColCount – jadvaldagi ustunlar sonini aniqlaydi;

RowCount – jadvaldagi satrlar sonini aniqlaydi;

FixedCols – fiksirlangan ustunlar sonini aniqlaydi;

FixedRows – fiksirlangan satrlar sonini aniqlaydi;

Options – jadval holatini aniqlaydi (aniqlash asosan uning parametrlarida bajariladi, masalan, *GoEditing* parametr true qiymatga ega bo'lsa, yacheykani tahrirlash mumkin, aks holda mumkin emas);

ColWidths – jadvaldagi har bir ustun kengligini aniqlaydi;

DefaultColWidth – jadvalning boshlang'ich ustunlar kengligini aniqlaydi;

DefaultRowHeight – jadval satrining boshlang'ich balandligini aniqlaydi;

FixedColor – fiksirlangan yacheyka rangini aniqlaydi;

RowHeights – jadval satri balandligini aniqlaydi;

Cells – simvol qatorli ikki o'lchamli massivni aniqlaydi.

3-misol. Tasodifiy butun sonlar bilan to'ldirilgan butun sonlardan iborat matritsa berilgan. Matritsaning maksimal, minimal elementlari va elementlar yig'indisi topilsin.

Forma dizaynini yaratib olamiz:

1) yangi forma yaratamiz. Yaratilgan formaga 1 ta *StringGrid*, 1 ta *label*, 4 ta *Button* va 1 ta *Edit* komponentalarini o'rnatamiz. *Object inspector* bo'limida *StringGrid* komponentasining xususiyatlarini quyidagicha o'zgartiramiz:

FixedCols -0,

FixedRows -0,

ColCount -5,

RowCount -5;

2) *Option* xossasiga kirib, *GoEditing* parametrini *True* qiymatiga o'zgartiramiz;

3) dastur kodini quyidagicha yozamiz:

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
#include "Unit1.h"  
//-----
```

```

#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
int a[5][5];
    int i,j,s,max,min;
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    for (i=0;i<=4;i++)
        for (j=0;j<=4;j++)
        {
            a[i][j]=random(50);
            StringGrid1->Cells[i][j]=IntToStr(a[i][j]);
        }
}

    void __fastcall TForm1::Button2Click(TObject *Sender)
    {
        max=a[0][0];
        for (i=0;i<=4;i++)
            for (j=0;j<=4;j++)
                if(a[i][j]>max) max=a[i][j];
        Edit1->Text=IntToStr(max);
    }

.....
    void __fastcall TForm1::Button2Click(TObject *Sender)
    {
        min=a[0][0];
        for (i=0;i<=4;i++)
            for (j=0;j<=4;j++)
                if(a[i][j]<min) min=a[i][j];
        Edit2->Text=IntToStr(min);
    }
//-----

```

Dastur quyidagi ilovada bajariladi(5.2-rasm) :

The screenshot shows a software interface titled "Form1" with a dark blue background. It contains several input fields and labels:

- MAX:** 40 (with a red label)
- ELEMENTLAR MAX** (with a grey label)
- MIN:** 0 (with a blue label)
- ELEMENTLAR MIN** (with a grey label)
- O'RTACHA QIYMAT:** 20 (with a yellow label)
- O'RTACHA QIYMAT** (with a green checkmark label)
- BAJARDI:** (with a black label)
- QABUL QILDI:** (with a green label)
- CHIQUISH** (with a purple icon label)
- MASSIV ELEMENTLARI:** (with a grey label)

In the center, there is a large, stylized 3D text "DELL". To the right, there is a grid of numbers:

49	33	11	29	11
4	6	44	13	41
36	34	0	48	42
44	36	32	6	37

5.2-rasm. Dastur ilovasi

Topshiriqlar

1. A massivining musbat elementlari yig'indisi va manfiy elementlarining ko'paytmasi hisoblansin.
2. A massivi musbat elementlarining eng kichigi topilsin.
3. A massiv elementlarini teskari tartibda Y massiviga yozing.
4. Butun sonlardan tashkil topgan ($a_1, \dots, a_n$) massivi berilgan. Massivning juft o'rinlarida joylashgan manfiy elementlarini nolga almashtiring.
5. X massivi manfiy elementlarining o'rtacha arifmetik qiymati hisoblansin. Agar manfiy elementlar mavjud bo'lmasa, $S=0$ degan ma'lumot bosmaga chiqarilsin.
6. Kvadrat matritsa haqiqiy sonlardan tashkil topgan. Matritsaning eng kichik qiymati joylashgan qator elementlari yig'indisi topilsin.
7. Haqiqiy sonlardan tashkil topgan kvadrat matritsa berilgan. Matritsaning asosiy diagonal elementlari yig'indisi hisoblansin va diagonal elementlarning eng kattasi aniqlansin.
8. X massivining juft o'rinlarida joylangan elementlarining o'rtacha arifmetik qiymati hisoblansin.

Nazorat savollari:

1. Massiv deb nimaga aytiladi?
2. Necha xil massivlar mavjud?
3. StringGrid komponentasining qanday xossalari mavjud?
4. Massivlar qaysi bo'limda e'lon qilinadi?

6 - amaliy mashgʻulot

Aralash va toʻplam toifasidagi maʼlumotlar va ularni dasturlarda ishlatish

Ishdan maqsad:

1. Nazariy qism bilan tanishish
2. Maʼlumotlarning aralash (struktura) toifasidagi berilganlarni xotiraga kiritish va xotiradan chiqarishni tashkil qilishni oʻrganish.
3. Yozuvlar qatnashgan masalalarning yechish dasturini tuzishni oʻrganish.

Nazariy qism

Iqtisod va axborotni qayta ishlash masalalarini yechishda maʼlum turdagi hujjatlar, kataloglar, roʻyhatlar, vedomostlar ishlatiladi. Masalan, talabalarning anketa maʼlumotlari: familiyasi, ismi, otasining ismi, turar joyi, tugʻilgan yili, mutaxassisligi, guruh raqami va hokazo. Bu hollarda turli toifadagi maʼlumotlarni bir guruhga birlashtirish zaruriyati tugʻiladi. Bizning misolimizda bu maʼlumotlarni talaba guruhiga birlashtirish mumkin. Koʻrinib turibdiki, bu guruhdagi maʼlumotlarning toifalari turlicha: familiya, ism - qator (soʻz), tugʻilgan yili, guruh raqami - butun toifaga tegishli.

C++ tilida bunday maʼlumotlarni struktura(aralash) toifasi yordamida ifodalash imkoniyati berilgan. Struktura toifasining qiymati sifatida yozuvni olish mumkin.

Yozuv deb turli toifadagi maʼlumotlarning cheklangan toʻplamiga aytiladi.

Dasturda struktura toifasi bir necha usullarda tavsiflanishi mumkin:

1. ***struct<toifa nomi>***

```
{  
<toifa><1element>;  
<toifa><2 element>;
```

...

```
};
```

Masalan:

```
struct Sana
```

```
{  
    int day;  
    int month;  
    int year;  
}; Sanatav;// Sana toifasidagi oʻzgaruvchi
```

2. ***struct***

```
{  
<toifa><1 element>;
```

<toifa><2 element>;

...

};<o‘zgaruvchilar ro‘yhati>;

Masalan:

```
struct
{
int min;
int sec;
int msec;
} Vaqt, second;
```

3. Struktura toifasini Typedef xizmatchi so‘zi orqali ham tavsiflash mumkin.

Masalan:

typedef struct

{
float re;

float im;

}Complex;

Complex a[100]; //100 ta complex sondan iborat a massiv.

Dasturda struktura toifasi elementlariga qiymatlarni kiritish oqimi cin yoki initsializatsiyalash yordamida kiritish mumkin. Masalan:

```
struct Student
{ char name[20];
int kurs;
float baho;
};
Student s={"Axmedov",1,5.0};
```

Yoki

```
cout<<"familiya"<<endl;
cin>>s.name;
```

Bir variantning yechimi.

1-masala. Berilgan massivda kitoblar haqida ma‘lumotlar berilgan. Har bir kitob to‘g‘risida: - kitobning nomi, muallifning ism-sharifi, kitob bosilgan sanasi. Shunday dastur tuzish kerakki, unda kitob chiqqan sanasi 2008yilga teng yoki 2008 yildan katta bo‘lsin va bu kitoblar soni aniqlansin.

```
//-----
#include <iostream.h>
#include <conio.h>
#include <vcl.h>
```

```

#pragma hdrstop
//-----
#pragma argsused
int main(int argc, char* argv[])
{ typedef struct {
    char title[40];
    char author[20];
    int entry; }book ;
int sum=0;
book k;
book b[10];
int i;
for (i=1;i<=5; i++)
{ cout<<"KITOB NOMI"<<endl;
  cin>>b[i].title;
  cout<<"MUALLIF"<<endl;
  cin>>b[i].author;
  cout<<"CHIQQAN YILI"<<endl;
  cin>>b[i].entry;
}
for (i=1; i<=5; i++)
  if (b[i].entry<=2008) sum=sum+1;
  cout<<"Kitoblar yigindisi Sum="<<sum;
  getch();
  return 0;
}
//-----

```

2-masala. Talabalarining familiyasi va ta'lim olayotgan bosqichi haqida ma'lumotlar berilgan. Bu ma'lumotlar asosida 2-bosqich talabalari ro'yhatini chiqaring.

```

//-----
#include <iostream.h>
#include <conio.h>
#include <vcl.h>
#pragma hdrstop
//-----
#pragma argsused
int main(int argc, char* argv[])
{ typedef struct {
    char fam[20];
    int kurs;
    }talaba ;

```



```

talaba stud;
talaba b[100];
int i,n;
cout<<"talabalar sonini kiriting";
cin>>n;
for (i=1;i<=n; i++)
{ cout<<i<<"-talaba familiyasi"<<endl;
  cin>>b[i].fam;
  cout<<i<<"-talaba kursi"<<endl;
  cin>>b[i].kurs;
}
cout<<"2-bosqich talabalari"<<endl;
for (i=1; i<=n; i++)
{if (b[i].kurs==2)
  cout<<b[i].fam<<endl;}
getch();
return 0;
}
//-----

```

Topshiriqlar:

1. Kitob mualliflari asosida kerakli kitoblarni izlab topish dasturini tuzing.
2. Kafedra xodimlari familiyasi, staji, lavozimi kiritilgan. Bu ma'lumotlar ichidan assistentlar ro'yhatini chiqaring.
3. Talabalarning 1-semestrda fanlardan olgan ballari berilgan. Bu ma'lumotlar asosida talabalarga beriladigan stipendiya miqdori aniqlansin.
4. Talabalarning familiyasi, guruxi, fanlardan olgan ballari berilgan. Bu ma'lumotlar asosida a'lochi talabalar familiyasi va guruxini aniqlang.
5. Avtomashina egasining familiyasi, adresi, mashina nomeri ma'lum. Shu ma'lumotlar asosida mashina nomeri orqali avtomashina egasining familiyasi va adresi izlash yordamida chiqarilsin.
6. Informatika fanidan olimpiadada ishtirok etgan talabalar guruhi ro'yxati berilgan. Shunday dastur tuzilsinki, natijada olimpiadada birinchi to'rtta o'rinni egallagan talabalar ro'yxati chiqsin.

Nazorat savollari

1. Ma'lumotlarning aralash toifasi - yozuvlardan dasturda foydalanish zaruriyati.
2. Yozuvlarni dasturda tavsiflashni qanday usullari mavjud?
3. Aralash toifadagi ma'lumotlarni kiritish va bosmaga chiqarish qanday tashkil etiladi?

7-amaliy mashg'ulot

Borland C++ Builder 6 da foydalanuvchining protsedura va funksiyalarida ma'lumotlarning faylli toifasini tavsiflash va ularni qo'llash.

Ishdan maqsad:

1. Nazariy qism bilan tanishish.
2. Fayllar bilan ishlashni o'rganish.
3. Ma'lumotlarning faylli toifasi yordamida dasturlar tuzish.

Nazariy qism

C++ ixtiyoriy dasturida o'zida tur, o'zgarmas, makros, funktsiya va sinflar aniqlangan standart kutubxonaga murojaat mavjud bo'ladi.

Ular dasturda qo'llash uchun **#include** direktivasi yordamida dasturning boshlang'ich matniga mos e'lonlardan tashkil topgan sarlavha fayllarni kiritish lozim. **Fayl** - bu bir xil toifadagi qiymatlar joylashgan tashqi xotiradagi nomlangan sohadir. Operatsion sistema nuqtai-nazaridan fayl hisoblangan har qanday fayl C++ tili uchun *moddiy fayl* hisoblanadi. C++ tilida *mantiqiy fayl* tushunchasi bo'lib, u fayl turidagi o'zgaruvchini anglatadi. Fayl turidagi o'zgaruvchilarga boshqa turdagi o'zgaruvchilar kabi qiymat berish operatori orqali qiymat berib bo'lmaydi. Boshqacha aytganda fayl turidagi o'zgaruvchilar ustida hech qanday amal aniqlanmagan. Ular ustida bajariladigan barcha amallar funksiyalar vositasida bajariladi.

Fayllar bilan ishlash quyidagi bosqichlarni o'z ichiga oladi:

- fayl o'zgaruvchisi albatta diskdagi fayl bilan bog'lanadi;
- fayl ochiladi;
- fayl ustida yozish yoki o'qish amallari bajariladi;
- fayl yopiladi;
- fayl nomini o'zgartirish yoki faylni diskdan o'chirish amallarini bajarilishi mumkin.

Kutubxona funktsiyalarini ularni qo'llanilishiga qarab quyidagi guruhlariga bo'lish mumkin: kiritish/chiqarish, qatorlarni qayta ishlash, matematik funktsiyalar, dinamik xotira bilan ishlash, qidirish, saralash va boshqalar.

C++da oqimlar orqali kiritish/chiqarish funktsiyalarini qo'llash uchun dasturda quyidagi sinflarni ishlatish kerak:

- <iostream.h>- kiritish oqimi
- <ostream.h> - chiqarish oqimi
- <iostream.h> - kiritish/chiqarish.

Kiritish/chiqarishda ma'lumotlar baytlar okimi sifatida karaladi. Fizik jixatdan oqim faylni tashkil qiladi.

Quyidagi faylli oqimlar fayllar bilan ishlash uchun xizmat qiladi:

1) ***ofstream name("path\file_name");*** - faylli chiqarish. Oqimni aniq fayl bilan bog'laydi, faylni yozish uchun ochadi. Bu yerda name- oqim nomi(ixtiyoriy identifikator), faylga ma'lumotlarni yozish va yoki undan o'qishda shu nomdan foydalaniladi. path\file_name- fizik faylning yo'li.

Masalan: ***ofstream book("c:\kitob\bob\bo'lim1.dat");***

Agar faylning yo'li ko'rsatilmasa, u holda fayl joriy katalogda yaratiladi:

ofstream books("bo'lim1.dat");

2) ***ifstream name("path\file_name");*** - faylni undan ma'lumotlar o'qish uchun ochadi.

Masalan: ***ifstream book("c kitob\bob\bo'lim1.dat");***

ifstream books("\bo'lim1.dat");

O'qish va yozish uchun ochilgan fayllarni albatta yopish kerak - ***name.close();***

Book.close(): books.close();

1-masala. Berilgan 10 ta butun sondan iborat A massivning musbat elementlarini F faylga yozing. (dastur konsol rejimida bajariladi).

```
#include <conio.h>
#include <iostream.h>
#include <fstream.h>
#include <vcl.h>
#pragma hdrstop
//-----
#pragma argsused
int main(int argc, char* argv[])
{ int a[10], i;
  ofstream file("musbat.txt");
  file<<"A massivning musbat elementlari"<<endl;
  for(i=1;i<=10;i++)
  {
    cout<<"a["<<i<<"]="";    cin>>a[i];
    if (a[i]>0) file<<" "<<a[i];
  }
  file.close();getch();
```

```

        return 0; }
//-----

```

2-masala. Matritsa va vektorlar berilgan. Son qiymatlari ixtiyoriy. Ushbu qiymatlardan foydalanib, matritsani vektorga ko‘paytirish, matritsaning izini hisoblash va vektorning yigindisini hisoblash dasturini tuzing.

Avval matritsa va vektorlarning son qiymatlarini o‘zida saqllovchi fayl xosil qilamiz. So‘ngra bu ma‘lumotlardan foydalanamiz.

```

#include <iostream.h>
#include <fstream.h>
#include <stdlib.h>
#include <time.h>
void main ( )

{ srand (time (0));
int a [3][3], b[3], i, j;
ofstream said ("akbar.txt");
for ( i=0; i<3; i++)
    { for (j=0; j<3; j++)
        { a[i][j] = rand( );
          said <<a[i][j]; } }
    for (i=0; i<3; i++)
        { b[i] = rand( );
          said << b[i]; }
said.close ( );
}

```

```

#include <iostream.h>
#include <fstream.h>
void main ( )
{
int a[3][3], b[3], i, j, c[3], s1=0, s2=0;
ifstream said ("akbar.txt");
for ( i=0; i<3; i++)
    for (j=0; j<3; j++)
        said >>a[i][j];
for ( i=0; i<3; i++) said >> b[i];
for ( i=0; i<3; i++)
    { c[i] = 0;

```

```

    for (j=0; j<3; j++)
        c[i] = c[i] + a[i][j] * b[i];
    cout << "c="<<c[i]<<endl; }
for ( i=0; i<3; i++)
    s1 = s1 + a[i][i];
for ( i=0; i<3; i++)
    s2=s2 + b[i];
cout << "s1="<<s1<<"  s2="<<s2<<endl;}

```

Topshiriqlar:

1. 2ta butun sonni va ularning yig'indisini o'zida saqlovchi fayl hosil qiling va undan keyingi dasturda foydalaning.
2. Joriy disk va joriy papkada gurux.cpp nomli fayl ochib, unga gurux talabalari familiyasi va 1-semestrda to'plagan o'rtacha ballari yozilsin.
3. D diskda son.dat nomli fayl ochib, unga 50 ta butun sonlardan iborat massivning juft sonlari yozilsin.
4. Haqiqiy sonlar yozilgan f fayli berilgan. f fayldagi elementlarning o'rtacha arifmetigidan kichik bo'lgan elementlar miqdorini aniqlansin.
5. Talabalar familiyasi va fanlardan olgan ballari berilgan. Joriy disk va joriy papkada "talaba" nomli fayl yaratib unga a'lochi talabalar familiyasini yozish dasturini tuzing.
6. Kitoblar xaqida ma'lumotlardan tashkil topgan fayl yarating

Nazorat savollari:

1. Fayl nima?
2. C++i tilida fayllarning necha xil turi mavjud?
3. Fayl bilan ishlash uchun qaysi funktsiya va protseduralar mavjud?
4. Fayllar o'zgaruvchilar bo'limida qanday qilib e'lon qilinadi?

8-amaliy mashg'ulot

Borland C++ Builder 6ning grafik imkoniyatlarini ilovalarda qo'llash.

Ishdan maqsad:

1. Ilovada grafik tasvirlarni yaratishda C++ Builder 6 muhitining asosiy imkoniyatlari bilan tanishish
2. Grafik tasvirlarni yaratishda komponentalardan foydalanishni o'rganish.

Nazariy qism

C++ Builder 6 muhitida ilovaga turli ko‘rinishdagi grafik tasvirlarni chiqarish imkonini beradi. Bunda xotiradagi mavjud rasmlarni ilovaga qo‘yish yoki dasturda chizish, bo‘yash, harakatlantirish kabi amallarni bajarish uchun standart protseduralardan foydalanish mumkin.

C++ Builder 6 grafik tasvirlarni yaratish uchun asosan ikkita sinf – **TGraphic** va **TPicture**ning xossa va protseduralaridan foydalanadi. **TGraphic** sinfi 2 turdagi fayllarni (.vmf va .bmp) yaratadi. **TPicture** sinfi esa boshqa grafik muharrirlarda yaratilgan tayyor tasvirlardan foydalanish uchun maxsus xossa va protseduralardan foydalanish imkonini beradi.

Grafik instrumentlarni boshqarish va sozlash uchun TCanvas, TFont, TPen va h.k. kabi sinflar belgilangan. Ularning ba’zi birlarini ko‘rib chiqamiz.

TCanvas sinfi ilovaga aylana, to‘g‘rito‘tburchak, ellips kabi figuralarni chizish, ularni bo‘yash, rasmga math qo‘shish va hokazo protseduralarini (usullarini) tavsiya etadi. Ularning asosiylari quyidagilar:

- Arc(x1,y1,x2,y2,x3,y3) – yoy chizish;
- Ellips(x1,y1,x2,y2) – ellips chizish;
- LineTo(x,y) – joriy nuqtadan (x,y) nuqtaga chiziq o‘tkazish;
- MoveTo(x,y) – (x,y) nuqtani joriy qilish;
- Ract(x1,y1,x2,y2) – to‘g‘rito‘rtburchak chizish;
- FloodFill(x,y,Color, Style) – sohani bo‘yash;
- Polygon(x,y:array of Tpoint) – ko‘pburchak chizish;
- TextOut(x,y,str) – (x,y) nuqtada Str qatorini yozish va h.k.

Figuralarni chizish, matnlarni yozishda quyidagi standart ranglarni ishlatish mumkin:

- ClBlack – qora rang;
- ClRed – qizil rang;
- ClYellow – sariq rang;
- ClBlue – ko‘k rang;
- ClGreen – yashil rang;
- ClGray – kulrang;
- ClWhite – oq rang va h.k.

Ilovaga xotiradagi mavjud tasvirlarni o‘rnatish uchun **TPicture** sinfiga mansub bo‘lgan **TImage** va **TShape** komponentalarini ishlatish tavsiya etiladi.

TImage komponentasi Additional instrumentlar palitrasida joylashgan bo‘lib,  piktogrammasiga ega. Bu komponenta tanlanib, Formadan rasm uchun joy belgilanadi, so‘ngra Object Inspektor jadvalidan **Picture** xossasi ochiladi. Ochilgan muloqot oynasida quyidagi amallar bajariladi:

Load – fayldan rasmni yuklash;
Save – rasmni saqlash;
Clear – rasmni olib tashlash;
Ok – tanlangan rasmni ajratilgan joyga qo'yish;
Cancel – amallarni bekor qilish.

TShape komponentasi Formaga aylana, to'rtburchak, ellips va boshqa shakllarni joylashtirish uchun ishlatiladi. Uning quyidagi xossalari mavjud:

Brush – shaklni bo'yash uchun mo'yqalam;
Pen – shakl chetini chizish uchun qalam;
Shape – ekranga chiqadigan shaklni aniqlaydi:
StRectangle – to'rtburchak;
StSquare – kvadrat;
StRoundRect – chetlari aylanasiimon to'rtburchak;
StRoundSquare – chetlari aylanasiimon kvadrat;
StEllipse – ellips;
StCircle – aylana.

Shape komponentasi ham Additional palitrasida joylashgan bo'lib, u  ko'rinishdagi piktogrammaga ega. Bu tugmachani bosib, formadan shakl uchun joy ajratiladi, so'ngra xossalar bo'limidan *Shape* xossasiga kirib, kerakli shakl tanlanadi.

1-misol. Paint Box visual komponentasi yordamida rangli kvadrat yasang.

Vizual rejimda dastur kodi:

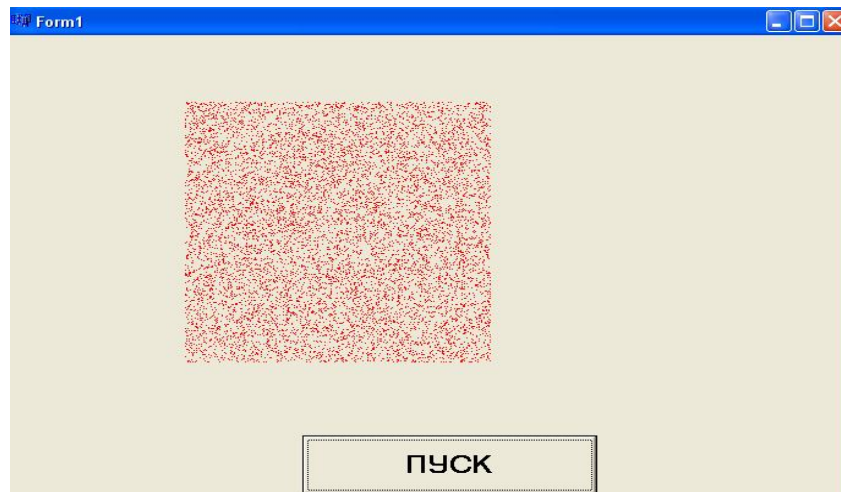
```
#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner) : TForm(Owner)
{
}
//-----
int i,x,y;
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    randomize;
    for (i=1;i<=300;i++)
```

```

{ x=random(250);
y=random(250);
PaintBox1->Canvas->Pixels[x][y]=RGB(255,0,0); }
}
//-----

```

Quyida forma ilovasida olingan natija keltirilgan(8.1-rasm):



8.1- rasm. Dastur natijasi

2-misol. Sohani turli qalinlikdagi rangli chiziqlar bilan to'ldiring

Vizual rejimda 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)
{
}
//-----
int i,x,y;
void __fastcall TForm1::Button1Click(TObject *Sender)
{
PaintBox1->Canvas->Pen->Color=RGB(random(255),random(255),random(255));
}

```

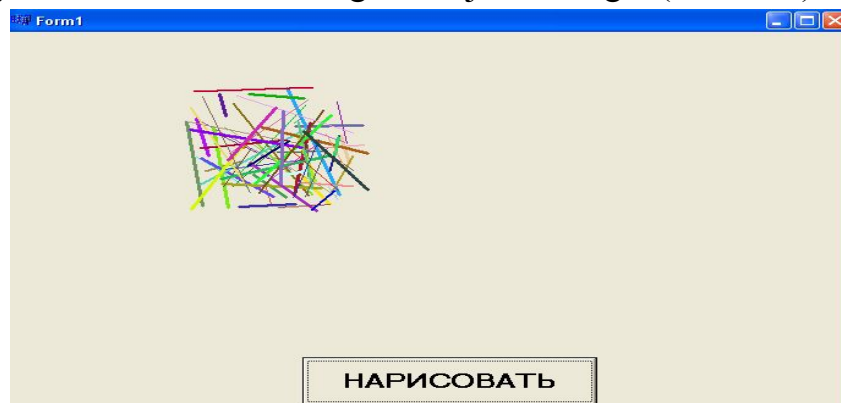


```

PaintBox1->Canvas->Pen->Width=random(3)+1;
x=random(150);
y=random(150);
PaintBox1->Canvas->MoveTo(x,y);
x=random(150);
y=random(150);
PaintBox1->Canvas->LineTo(x,y);
}
//-----

```

Quyida frma ilovasida olingan natija keltirilgan(8.2-rasm):



8.2-rasm. Dastur natijasi

3-misol. Sohani tasodifiy holda ellipslar bilan to'ldiring.

Vizual rejimda 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)
{
}
//-----
int i,x,y;
void __fastcall TForm1::Button1Click(TObject *Sender)

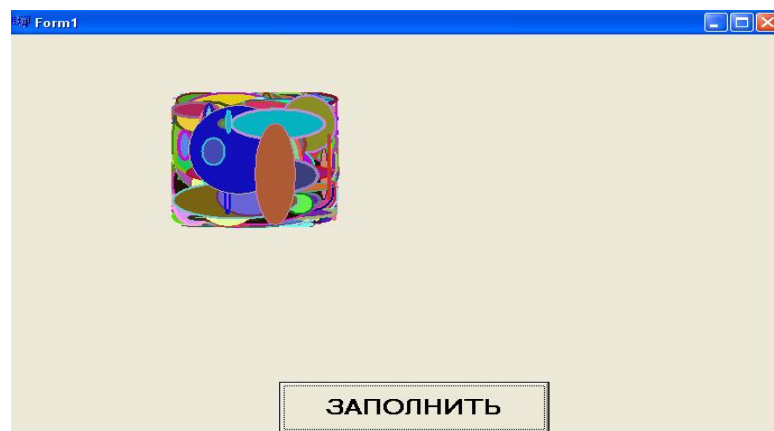
```

```

{int i,x,y,x2,y2;
for (i=1;i<=300;i++) {
PaintBox1->Canvas->Pen->Color=RGB(random(255),random(255),random(255));
PaintBox1->Canvas->Pen->Width=random(3)+1;
PaintBox1->Canvas->Brush->Color=RGB(random(255),random(255),random(255));
x=random(150);
y=random(150);
x2=random(150);
y2=random(150); PaintBox1->Canvas->Ellipse(x,y,x2,y2); }
}
//-----

```

Quyida forma ilovasida olingan natija keltirilgan(8.3-rasm):



8.3-rasm. Dastur natijasi

Topshiriqlar:

1. Sohani to‘g‘ri to‘rtburchaklar bilan to‘ldiring.
2. Sohani rangli aylana shakllari bilan to‘ldiring.

Nazorat savollari:

1. C++ dasturida qanday yo‘llar bilan grafika yaratish mumkin?
2. *Image* komponentasining vazifasi nimalardan iborat?
3. C++ dasturida avtofiguralarni chizishda qanday metodlardan foydalaniladi?
4. *TChart* komponentasining vazifasi nimalardan iborat?

9-amaliy mashg'ulot

Borland C++ Builder 6 muhitida ma'lumotlar bazasi bilan ishlash

Ishdan maqsad:

1. Ma'lumotlar bazasiga qo'yiladigan talablar bilan tanishish.
2. C++da ma'lumotlar bazasi bilan ishlovchi komponentlarni o'rganish.
3. Ma'lumotlar bazasiga murojaat qiluvchi komponentalar bilan ishlash.

Bu darsda biz C++ Builder 6 muhitida ma'lumotlar bazasi bilan ishlash asoslarini ko'rib o'tamiz. Ma'lumotlar bazasining maqsadi ma'lum strukturada kiritilgan, ma'noga ega bo'lgan ma'lumotlarni bir yoki bir nechta joyda ortiqcha sarflarsiz (xotira va zaxiralarni iqtisod qilib) saqlashdir. Ma'lumotlar bazasi ikkita asosiy maqsadlarni ko'zda tutadi: ma'lumotlarni yo'qolishdan saqlash va ularning ishonchliligini oshirish.

Ma'lumotlar bazasiga qo'yiladigan talablar. Yaxshi loyihalangan ma'lumotlar bazasi:

– foydalanuvchilarning ma'lumotlar bazasiga bo'lgan barcha talablarini qondiradi. Shuning uchun ma'lumotlar bazasini loyihalashdan oldin foydalanuvchilarning ma'lumotlar bazasiga bo'lgan talablarini keng miqyosda o'rganib chiqish zarur.

Ma'lumotlarning mosligi va ishonchliligini kafolatlaydi. Jadvallarni tuzishda foydalanuvchi tomonidan mumkin bo'lmagan ma'lumotlar kiritilishining oldini olish uchun ma'lum shartlarni kiritish kerak.

C++ Builder6 muhitida ma'lumotlar bazasini yaratish uchun bir qancha usullardan foydalanish mumkin:

– ma'lumotlar bazasini yaratish va qayta ishlash uchun mo'ljallangan tizimlarda ma'lumotlar bazasini yaratib, C++ Builder vositalari yordamida ularni qayta ishlash, masalan, ma'lumotlar bazasining maydonlari ustida amallar bajarish, ma'lumotlarni yangilash va hokazo;

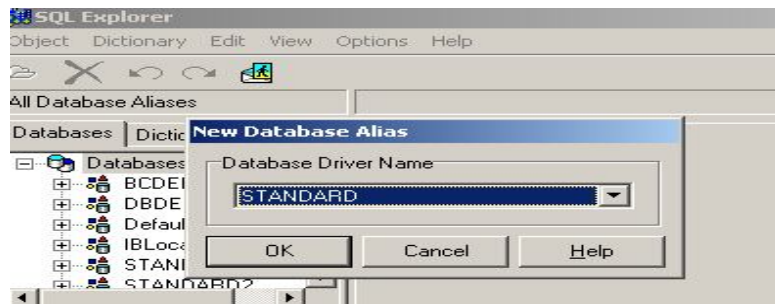
– C++ Builder 6 muhitining vositalari yordamida MB yaratish va qayta ishlash.

Amaliy qism

C++da ma'lumotlar bazasi bilan ishlash:

C++ da ma'lumotlar bazasi bilan ishlaganda dastlab **Alias** hosil qilinadi. Alias ma'lum bir guruhga tegishli bo'lgan jadvallarni qaysi toifaga tegishlilikini (jadval hosil qiladigan drayverni), jadvallarni yo'lini o'zida saqlaydi. C++ da Alias hosil qilishning bir nechta usullari mavjud.

1. Menyuning **DataBase** bo'limidan **SQLExplorer** tanlanadi.
2. **Object -->New...** [Ctrl+N] tanlanadi yoki **SQLExplorer** darchasining o'ng qismida shichqonchaning o'ng qismi bosiladi va New tanlanadi.(9.1-rasm)

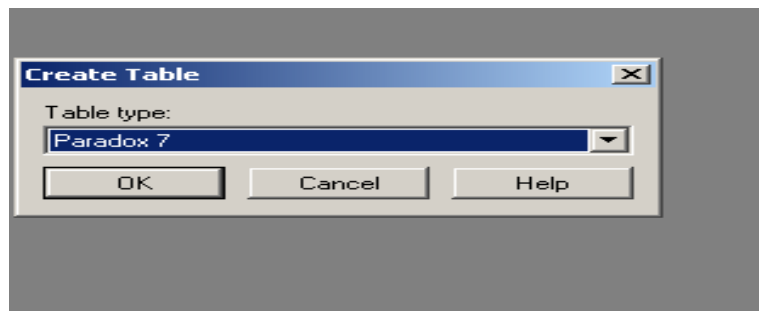


9.1-rasm

3. Yuqoridagi darchadan drayver tanlanadi. Standart holatda **Paradox** tanlanadi
4. Path bo‘limidan jadvallar saqlanadigan catalog ko‘rsatiladi.
5. Hosil qilingan Aliasni saqlash uchun sichqonchaning o‘ng tugmasi bosiladi va Apply [Ctrl+A] tanlanadi.

Jadvallar hosil qilish uchun quyidagi amallar ketma-ketligi bajariladi:

1. C++ menyusining **Tools** bo‘limidan **DatabaseDesktop** tanlanadi;
2. **DatabaseDesktop** dan **File→New→Table** tanlanadi(9.2-rasm)



9.2-rasm

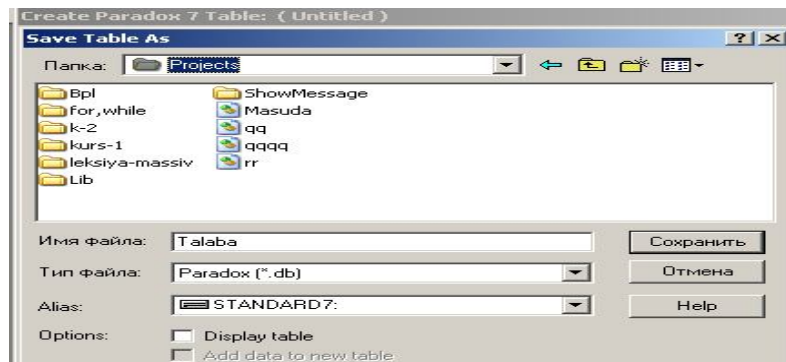
3. Jadval toifasi tanlanadi(9.3-rasm);



9.3-rasm

4. **Field Name** – jadval maydoni nomi; **Type** – maydon toifas; **Size** – hajmi; **Key** – kalit maydonini bildiradi .

5. Jadval maydonlari kiritilgandan so'ng Save As... tugmasi bosiladi;
6. Alias bo'limidan o'zimiz hosil qilgan alias nomi tanlanadi;
7. Jadvalga nom beriladi(9.4-rasm);
- 8.



9.4-rasm

9. **Options** bo'limida **Display Tablega** bayroqcha o'rnatib **Сохранить** tugmasi bosiladi. Ekranda **Database Desktop** da jadval ochiladi..

Maydonlar haqida ma'lumot

Maydon nomi 25 ta simvoldan iborat bo'lishi mumkin. Maydon nomini yozishda probel, apostrovdan foydalanmagan ma'qul. Chunki SQL so'rovlaridan foydalanganda muammo chiqishi mumkin(Zarurat bo'lsa Shift minus “_” belgisidan foydalanish maqsadga muvofiq).

Maydon toifasini tanlash uchun Type maydoniga o'tib, sichqonchanaing o'ng tugmasi bosiladi. Paradox uchun maydon toifalari quyidagicha bo'lishi mumkin:

1. **A** 1-255 Alpa satrli maydon. ASCII kodini barcha simvollarda qabul qiladi.
2. **N** number – 1-10308 butun son.
3. **\$** money- pul birligini bildiradi.
4. **S** short 1...32767 oralig'idagi butun sonlar.
5. **I** long integer- 1... 2147483648
6. **#** 0-32 BCD Binary Coded Decival formatdagi son.
7. **D** date- sanani bildiruvchi son.
8. **T** time- vaqt.
9. **@** Time Stamp- vaqt va sanani bildituvchi maydon.
10. **M** 1-240 Memo cheklanmagan satr ma'lumotlarini saqlashga mo'ljallangan.
11. **G** grafik tasvir ma'lumotlarni saqlash.
12. **O** OLE- tasvir, ovoz, xujjat va xokazolarni saqlash.
13. **L** logical- mantiqiy maydon.
14. **+** Autoincrement- avtomat ravishda yozuvni bittaga oshirib boradi. Yozuvlar o'chirilsa oldindagilar o'zgarishsiz qoladi.

Amaliy qism.

Yuqorida yaratilgan jadvalni C++Builder bilan bog'laymiz. Bu ish quyidagi ketma-ketlikda bajariladi:

1. **File->New->Application** tanlanadi;
2. Komponentalarning **BDE** bo'limidan **Table** komponentasi formaga qo'yiladi;
3. Table komponentasining **DatabaseName** xususiyatida **Alias** ko'rsatiladi;
4. **TableName** xususiyatida jadval nomi keltiriladi;
5. **Active** xususiyati **True** ga o'zgartiriladi;
6. Komponentalarning **DataAccess** bo'limidan **DataSource** komponentasi formaga qo'yiladi;
7. **DataSet** xususiyatida **Table1** belgilanadi;
8. Komponentalarning **DataControls** bo'limidan **DBNavigator** komponentasi formaga qo'yiladi;
9. **DataSource** xususiyatida **DataSource1** belgilanadi;
10. Komponentalarning **DataControls** bo'limidan **DBGrid** komponentasi formaga qo'yiladi;
11. **DataSource** xususiyatida **DataSource1** belgilanadi;
12. Dastur kodi ishga tushiriladi va jadvalga kerakli ma'lumotlar kiritiladi, taxrirlanadi, saqlanadi.

Topshiriq:

1. C++ tizimida baza bilan ishlovchi komponentalardan foydalangan holda yo'nalishingizga oid ma'lumotlarni kiritish va tahrirlash imkoniga ega bo'lgan baza yarating.

Nazorat savollari:

1. Ma'lumotlar bazasi nima?
2. C++ Builder 6 muhitida ma'lumotlar bazasi bilan ishlovchi qanday komponentalarni bilasiz?
3. *AdoConnection* komponentasining vazifasi?
4. *DBNavigator* komponentasining vazifasi?

Adabiyotlar:

1. Х.М.Дейтел, П. Дж. Дейтел. Как программировать на С++:Пятое малое издание. -М.: ООО «Бином- Пресс», 2007г.
2. Alex Allain. Jumping into C++. e.book.-USA, 2014.
3. Назиров Ш.А., Қобулов Р.В. Объектга мўлжалланган дастурлаш.Ўқув қўлланма. -Тошкент: Алоқачи, 2007.
4. Информатика. Фани бўйича лаборатория ишларини бажариш учун ўқув-услугий қўлланма 1-қисм – Т.: ТАТУ, 2009.
5. Каримова Д.К., Равилов Ш.М. Информатика. Информационные технологии. Учебное пособие. 1,2- часть. –Ташкент: ТашГТУ, 2004.
6. Kjell Backman “Structured Programming with C++”. 2012.
7. Кадиров М.М. Ахборот хавфсизлиги бўйича атама ва тушунчаларнинг рус, ўзбек ва инглиз тилларидаги изоҳли луғати. Ўқув қўлланма. – Тошкент: Иқтисод-Молия,2017.
8. Кадиров М.М. Ахборот технологиялари. Ўқув қўлланма. –Тошкент: Иқтисод-Молия,2017.

Elektron resurslar

1. <http://www.intuit.ru/department/informatics/intinfo/>
2. <http://www.dstu.edu.ru/informatics/mtdss/index.html>
3. <http://www.ziynet.uz>

Mundarija

	Soʻz boshi.....	3
1-amaliy mashgʻulot	Ilmiy-texnik masalalarni kompyuterda yechish bosqichlari. Algoritmash asoslari.....	4
2-amaliy mashgʻulot	Borland C++ Builder 6 integrallashgan muxiti bilan tanishish. Oddiy jarayonlarni dasturlash.....	10
3-amaliy mashgʻulot	Tarmoqlanuvchi algoritmlarni dasturlash.....	15
4-amaliy mashgʻulot	Takrorlanuvchi algoritmlarni dasturlash.....	20
5-amaliy mashgʻulot	Maʼlumotlarning strukturlashgan toifalari. Massivlardan foydalangan holda dasturlash.....	23
6-amaliy mashgʻulot	Aralash va toʻplam toifasidagi maʼlumotlar va ularni dasturlarda ishlatish.....	28
7-amaliy mashgʻulot	Borland C++ Builder 6 da foydalanuvchining protsedura va funksiyalarida maʼlumotlarning faylli toifasini tavsiflash va ularni qoʻllash.....	32
8-amaliy mashgʻulot	Borland C++ Builder 6ning grafik imkoniyatlarini ilovalarda qoʻllash.....	37
9-amaliy mashgʻulot	Borland C++ Builder 6 muxitida maʼlumotlar bazasi bilan ishlash.....	43
	Adabiyotlar.....	47

Muharrir: Sidikova K.A.
Musahhih: Adilhodjayeva Sh.M.