

**O‘ZBEKISTON RESPUBLIKASI OLIY TA’LIM FAN VA
INNOVATSIYALAR VAZIRLIGI**

**ISLOM KARIMOV NOMIDAGI
TOSHKENT DAVLAT TEXNIKA UNIVERSITETI**



DASTURLASH ASOSLARI

fanidan amaliy mashg‘ulotlarni bajarish uchun

USLUBIY KO‘RSATMALAR



TOSHKENT – 2023

Tuzuvchilar: Kadirov M.M., Zokirova F.R. “Dasturlash asoslari” fanidan amaliy mashg‘ulotlarni bajarish uchun Uslubiy ko‘rsatmalar. -Toshkent: ToshDTU, 2023. 44 b.

Ushbu amaliy mashg‘ulotlar bakalavriat talabalarining 60711400 – “Texnologik jarayonlar va ishlab chiqarishni avtomatlashtirish va boshqarish (tarmoqlar bo‘yicha)” ta’lim yo‘nalishiga mo‘ljallangan bo‘lib, unda C++ dasturlash tili yordamida muhandislik masalalarini yechish uchun dasturlar yaratish, matematik modellarni algoritmlarda ifoda etish, muxandislik masalalarini yechishda c++ konstruksiyalaridan foydalanish, mantiqiy dasturlashni misollarda qo‘llash, texnik masalalrni yechishda takrorlanish operatorlaridan foydalanish, muxandislik texnologiyalari amaliyotida strukturaviy masalalarni yechishda massivlarni qo‘llash, C++ da fayl toifasidagi ma’lumotlardan foydalanish hamda C++ tizimidagi grafik imkoniyatlardan foydalanib ilovalar yaratish kabi mavzulari bo‘yicha amaliy mashg‘ulotlarni bajarish uchun mo‘ljallangan.

Islom Karimov nomidagi Toshkent davlat texnika universiteti ilmiy-uslubiy kengashi qaroriga muvofiq chop etildi

Taqrizchilar: Abdurashidova K.T.

TATU “Kompyuter tizimlari”
kafedrasi dotsenti.

Karimova N.O.

ToshDTU “Axborot texnologiyalari”
kafedrasi dotsenti.

1-AMALIY MASHG‘ULOT

MATEMATIK MODELLARNI ALGORITMLARDA IFODA ETISH

Ishning maqsadi: Ushbu amaliy mashg‘ulotda matematik modellarni dasturlash uchun algoritmlash asoslari o‘rganiladi. Ilmiy-texnik masalani yechish bosqichlari bilan tanishiladi. Qo‘yilgan masalani dasturini tuzishda algoritmlardan foydalangan holda amalga oshiriladi. Algoritmlar turlari o‘rganiladi.

Reja:

1. Ilmiy-texnik masalani yechish bosqichlari.
2. Masala yechishning birinchi 3 ta bosqichini aniq misolda ko‘rib chiqish.
3. Algoritmlash, algoritm va uning xossalari, tavsiflash usullari, turlari to‘g‘risidagi tushunchalar.

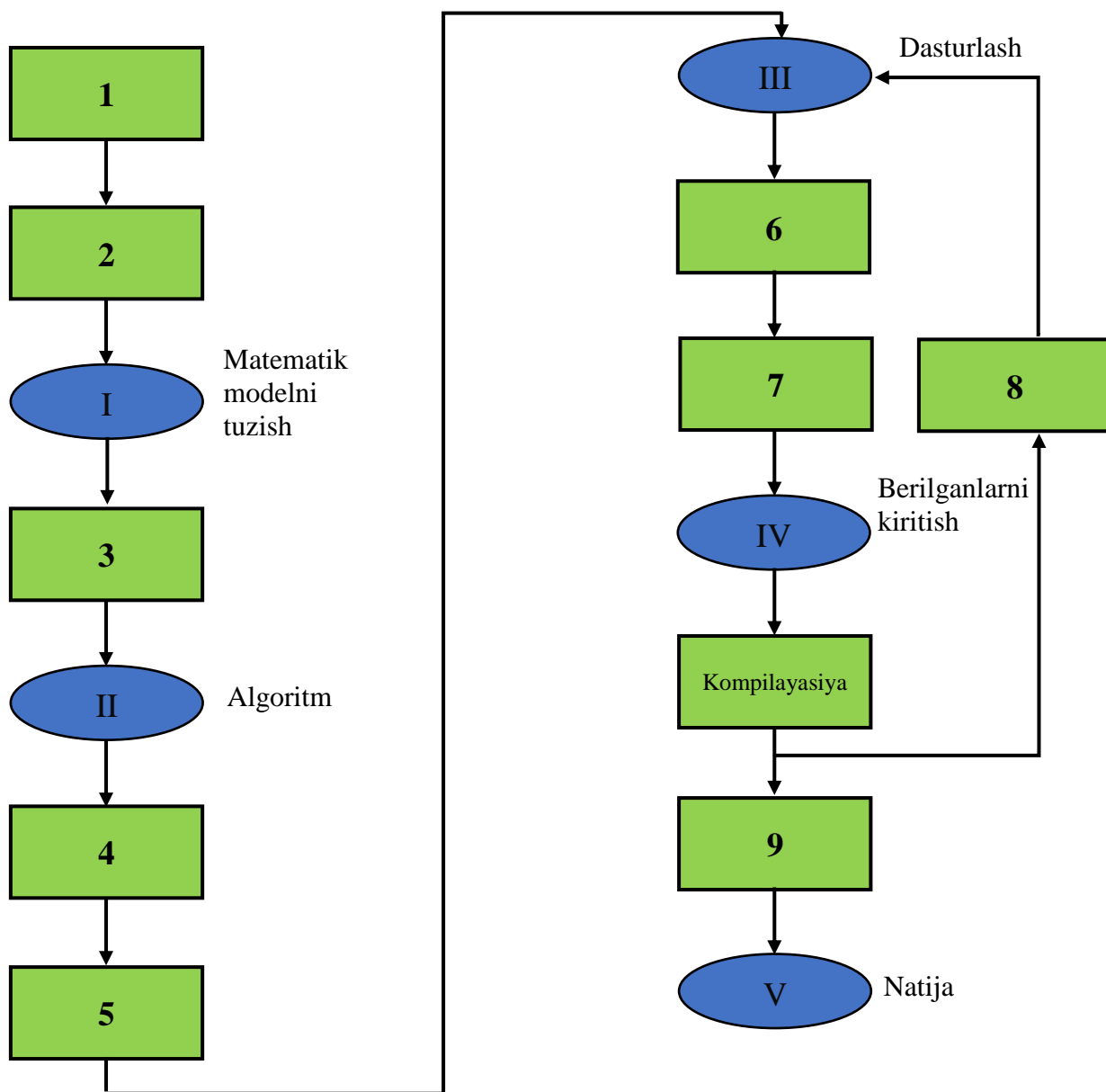
Nazariy qism

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‘t-kazish.
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.

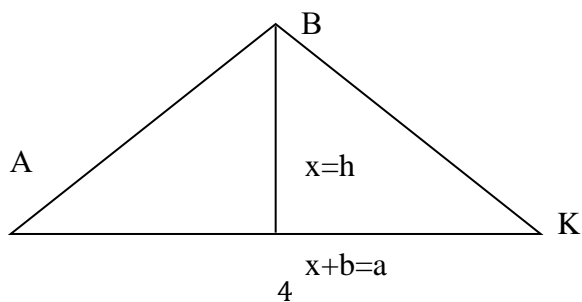


1.1- rasm. Masalalarni yechish bosqichlari

II. 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,

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

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:

Uzluklilik, aniqlik, natijaviylik va ommaviylik.

Algoritmni ishlab chiqishda bir necha xil usullar bor. Shulardan uchasi keng tarqalgan.



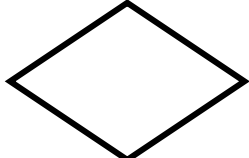
1. Algoritmni oddiy tilda tavsiflash.
2. Algoritmni sxema ko'rinishida ifodalash.
3. Algoritmni maxsus (algoritmlik) tilda yozish.




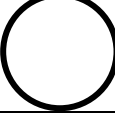
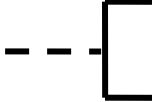
Algoritmni ifodalashning eng keng tarqalgan shakli – bu oddiy tilda so'zlar bilan bayon qilishdir, lekin murakkab algoritmlarda ko'rgazmalilikni yaxshi ta'minlay olmaydi. Bundan tashqari, algoritmning so'zdagi tavsifi hisoblash mashinasiga kiritish uchun yaramaydi. Buning uchun algoritmni mashina tilida 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.

Algoritmni tuzish simvollarini 1-jadvalda keltirilgan.

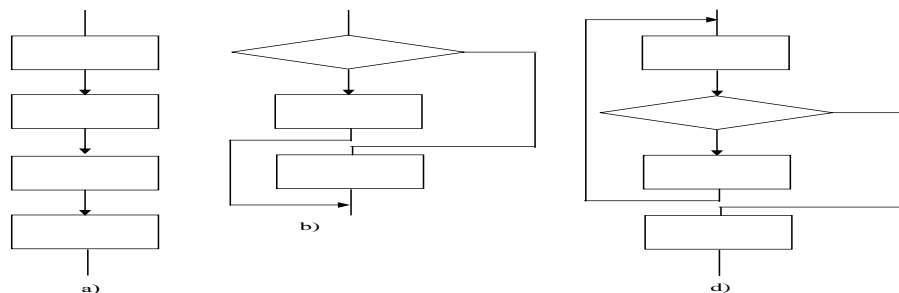
1-jadval

Nomi	Belgilanishi	Bajaradigan vazifasi
Boshlash, tamomlash		Axborotni qayta ishlashni boshlash yoki uni to'xtatib qo'yish
Jarayon		Bir yoki bir nechta amallarni bajarish operatori
Qaror		Biron-bir shartga bog'liq ravishda algoritm bajarilishi yo'nalishini tanlash

Modifikatsiya		Dasturni o'zgartiruvchi buyruq yoki buyruqlar turkumini o'zgartirish amalini bajarish
Kiritish-chiqarish		Berilganlarni kompyuterga kiritish yoki olingan natijalarni tasvirlash
Qism- dastur		Avvaldan aniqlangan jarayonga murojaat
Bog'lagich		Uzilib qolgan axborot oqimlarini ulash belgisi
Izoh		Va boshqalar

Murakkab masalalarning yechimini olish algoritmlari yuqoridagi turlarining barchasini o'z ichiga olishi mumkin.

Chiziqli turdagi algoritmlarda bloklar biri ketidan boshqasi joylashgan bo'lib, berilgan tartibda bajariladi (1.3-rasmga qarang).



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

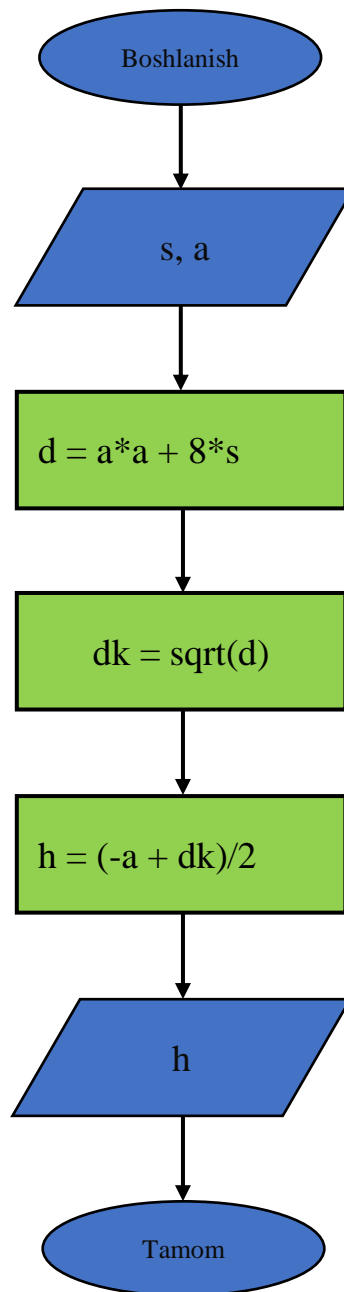
Misol.

$K = \sin a + \sin b$ 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-rasmda keltirilgan blok-sxema ko'rinishida tuziladi.



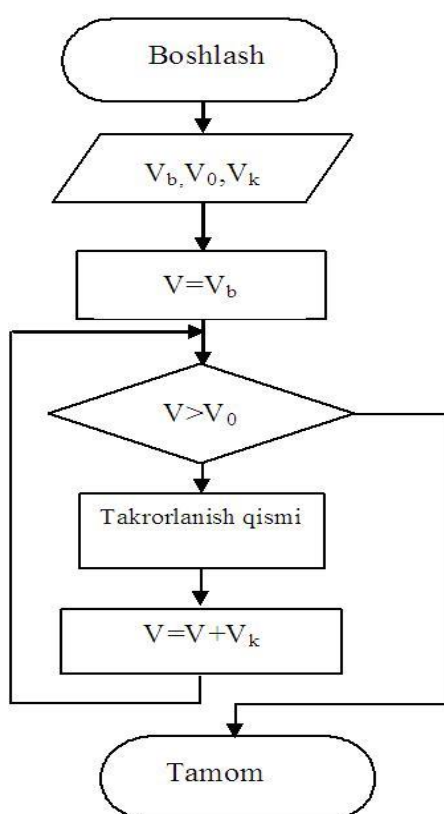
1.4-rasm. Masalani yechish blok-sxemasi

Amalda hamma masalalarni ham chiziqli turdagi algoritmgaga 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 algoritmlar deb ataladi.

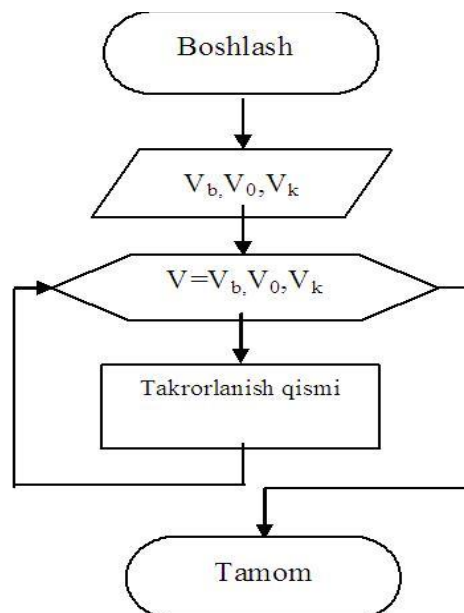
Ko‘pgina hollarda masalalarning yechimini olishda bitta matematik bog‘lanishga ko‘ra unga kiruvchi kattaliklarni turli qiymatlariga mos keladigan qiymatlarni ko‘p marta hisoblanadigan qismiga takrorlanishlar deb, takrorlanishlarni o‘z ichiga olgan algoritmlar takrorlanuvchi turdagi algoritmlar

deb ataladi.

Agar 1.5 va 1.6-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, 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

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

MUHANDISLIK MASALALARINI YECHISHDA C++ KONSTRUKSIYALARIDAN FOYDALANISH

Ishning maqsadi: C++Builder 6 dasturlash muhitining asosiy elementlari bilan tanishish. Chiziqli jarayonlarning yechish uchun visual komponentalar bilan tanishish. Dastur tuzish asoslarini o‘rganish.

Reja:

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.

7-rasmda ko‘rsatilganidek, C++Builder 6 muhitining oynasi WINDOWS AT tizimi oynasining uskunalari bilan mos keladi. C++Builder 6 muhiti tarkibiga asosan 5 ta oyna kiradi:

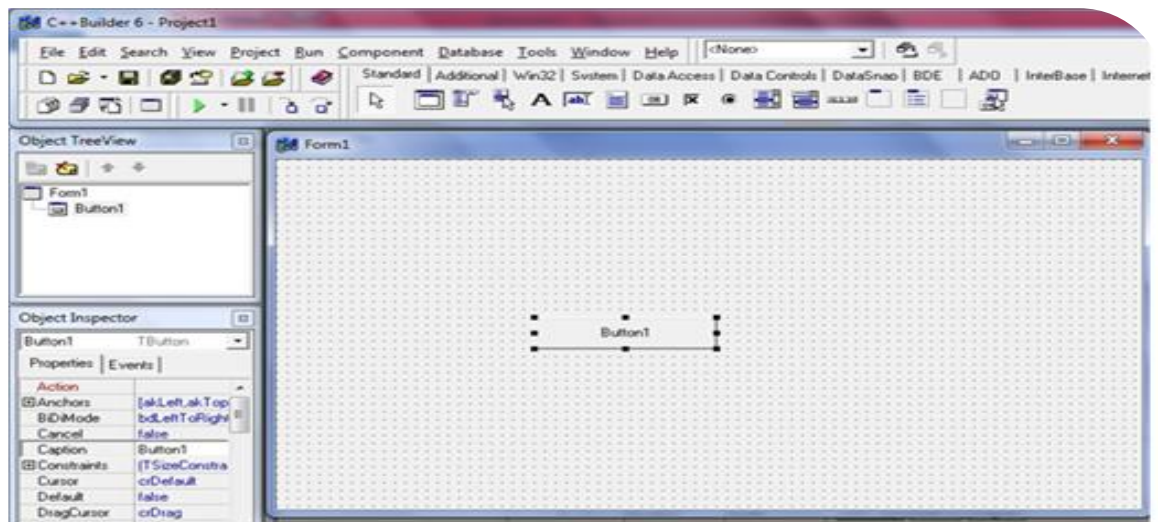
C++Builder 6 muhiti dasturini yuklash uchun quyidagi ketma-ketlikni amalga oshirish lozim:

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 ifodat loyiha ko‘rinishida tasvirlanadi (.cpp,.dfm,.h,.bpr va boshqalar).



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

C++Builder 6 tilining dasturlash ob'yektlari bo'lib quyidagilar xizmat qiladi: o'zgarmlar, o'zgaruvchilar, funksiyalar, 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 funksiyalari

Dasturlarda buyruqlar tarkibida turli ko'rinishdagi standart funksiyalardan foydalanuvchi funksiyalaridan foydalanish mumkin. Standart funksiyalar maxsus jadvallarda beriladi. Bu funksiyalar turli turdagi standart funksiyalar kutubxonasini tashkil qiladi. Standart funksiyalardan farqli foydalanuvchi funksiyalari dasturda foydalanuvchi tomonidan berilib, faqat maxsus misollarda ishlatiladi.

Qiyida standart funksiyalarni keltiramiz:

Standart funksiyalar

$(1+x)^2$	<code>pow(1+x,2)</code>
$\text{Sin}x$	<code>sin(x)</code>
$\text{Cos}x$	<code>cos(x)</code>
Tgx	<code>tan(x)</code> <code>sin(x)/cos(x)</code>
Lnx	<code>log(x)</code>
e^x	<code>exp(x)</code>
x^2	<code>x*x</code>
$ x $	<code>abs(x)</code>
$A^x e^{x \ln a}$	<code>pow(a,exp(x*log(a)))</code>
\sqrt{x}	<code>sqrt(x)</code>
$\text{Cos}^2 x^3$	<code>pow(cos(pow(x,3),2))</code>
X^4	<code>pow(x,4)</code>

Ifodalar

C++Builder 6 tilida ifodalar bir-biri bilan o‘zaro bog‘langan operandlar birikmasidan ifodat 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 funksiyalari 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.

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 funksiyalarni saqlanadi.

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

5!	
LXIV	
6,37	
0,7(4)	
11/4	
-1/6	
Π	
5*10⁶	
10⁶	

2-misol. Quyidagi ifodalarni C++Builder 6 tilida ifodalang

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

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

3- misol. 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-misol. 3-misoldagi operatorlarni ishlatgan holda dastur tuzing.

Topshiriqlar:

1. $(1+x)^2 / \sqrt{1+x^2}; |a^2 + bx^2|; \sin 8; \cos^2 x^3; \operatorname{tg} x; \operatorname{arcctg} 10^3; \arccos x.$
C++Builder 6 tilida chiziqli jarayonni ishlatgan holda dastur tuzilsin.

2. To‘g‘ri uchburchak piramidaning yuzasini hisoblash dasturi tuzilsin.

3. Quyidagi funktsiyani hisoblash dasturi tuzilsin:

$$a = \lg(\sqrt{e^{x-y}} + x^{|y|} + z), \quad b = e^{|x-y|} (\operatorname{tg}^2 x + 1)^x, \quad k = e^b \sin a, \quad \text{bu yerda } x = 1,542, \\ y = 0.75, z = 3.5$$

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

MANTIQUIY DASTURLASHNI MISOLLARDA QO'LLASH

Ishning maqsadi: Texnik masalalarni yechishda mantiqiy dasturlash asoslarini o'rganish. Shartli o'tish operatorlari yordamida tarmoqlanuvchi jarayonlarni dasturlash ko'nikmalarini hosil qilish.

Reja:

1. Nazariy qism bilan tanishtirish.
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'lish ham mumkin.

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

Misollar:

1) if (a<0) b = 1;

2) if (a<b && (a>d || a==0)) b++;

```
else {b= a; a = 0;}
```

```
3) if (a<b) {if (a<c) m = a; else m = c;}
```

```
else {if (b<c) m = b; else m = c;}
```

Birinchi misolda else operatori 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;
```

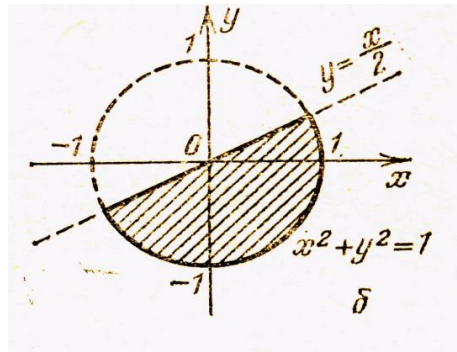
```
[default: operatorlar ;}]
```

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 algoritm uchun dastur tuzing, ya‘ni nuqtaning berilgan shtrixlangan maydonga tegishli ekanligini aniqlang.



```
//-----
#include<iostream.h>
#include<conio.h>
#include <vcl.h>
#pragma hdrstop
//-----
#pragma argsused
int main(int argc, char* argv[])
{
float x, y, r; int b;
cin>>x>>y>>r;
b=(x*x)+(y*y) <=r && (y<=x/2);
if (b=1) cout<<" nuqta strixlangan maydonga tegishli ";
else cout<<" nuqta strixlangan maydonga tegishli emas "<<endl;
getch(); return 0;
}
```

2-misol.

$$y = \begin{cases} \frac{ax^2 + bx}{\sin x + 1}, 0 \leq x \leq 2 \\ \frac{x^3 + \cos x}{\sqrt{1+x}} e^x, x > 2 \end{cases}$$

```
/-----
#include<iostream.h>
#include<math.h>
#include<conio.h>
#include <vcl.h>
#pragma hdrstop
//-----
#pragma argsused
int main(int argc, char* argv[])
{
float x, y, a, b;
cin>>a>>b>>x;
if ((x<=2) && (x>=0)) { y= (a*(x*x)+b*x) / (sin(x)+1);
```



```

cout<<"1-shart bajarildi y="<<y<<endl;};
    else if (x>2) { y=(x*x*x+cos(x)) / sqrt(1+x)*exp(x);
cout<<"2-shart bajarildi y="<<y<<endl;}
getch(); return 0;
}
//-----

```

3-misol. Kvadrat tenglama berilgan. Javob 3 hil variant yechimi ko‘rinishida olinsin.

```

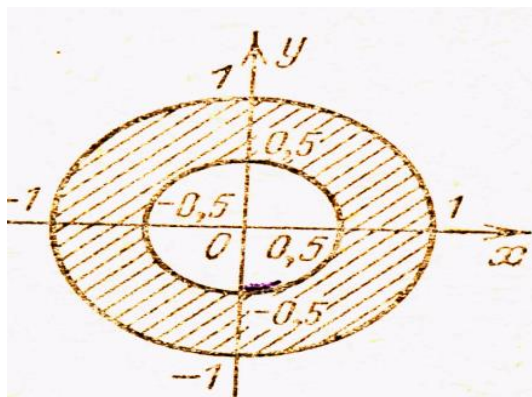
/-----
#include<iostream.h>
#include<math.h>
#include<conio.h>
#include <vcl.h>
#pragma hdrstop
//-----
#pragma argsused
int main(int argc, char* argv[])
{
float a,b,c,d,x, y, x1, x2;
int v;
cin>>a>>b>>c;
d = (b*b - 4*a*c);
if (d > 0) v = 1;
else if (d = 0) v = 2; else v = 3;
switch (v)
{
case 1: {x1 = (b - sqrt(d))/(2*a); x2 = (b + sqrt(d))/(2*a);
cout<<"pri d>0 x1="<<x1<<" x2="<<x2; break;
}
case 2: {x = b/(2*a); cout<<"pri d=0 x="<<x; break;
}
case 3: {cout<<"d<0 taqribiy ildizlar"; break;}

default : cout<<"echim aniqlanmadi" ;
}
getch(); return 0;
}
//-----

```

Topshiriq:

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.

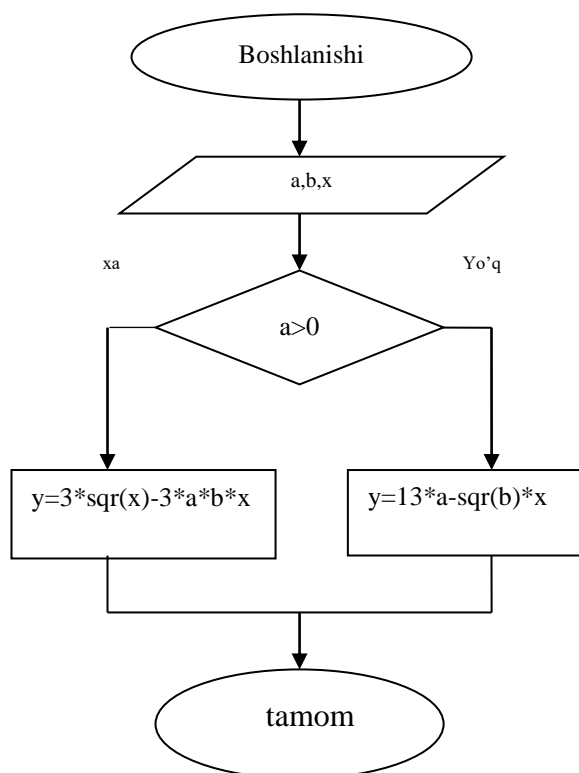
3. Yil oylarining tartib raqamiga asosan oy nomlarini bosmaga bosib chiqarish dasturi tuzilsin. Misol uchun, 3 sonini kiritsak «Mart» so'zi chiqsin.

Quyidagi misolni vizual rejim ko'rinishida ishlab ko'ramiz.

1- misol.

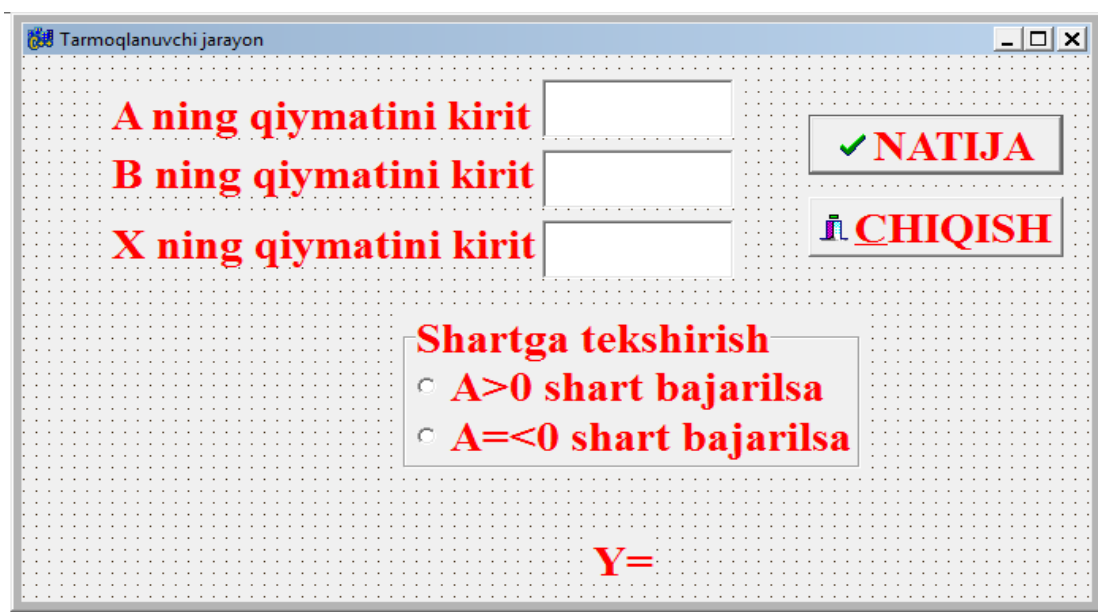
	Misolning berilishi	Berilgan qiymatlar
	$y = \begin{cases} 3x^2 - 3abx & \text{agar } a > 0 \\ 13a - b^2x & \text{agar } a \leq 0 \end{cases}$	$x=2,7$ $b=3,2$ $a=1,(-1)$

Berilgan misolning blok-sxemasi (3.1-rasm):



3.1-rasm. Masalani yechish algoritmi

Formaning ko'rinishini shakllantiramiz. Formada quyidagi komponentalarni o'rnatamiz: 4 ta Label, 3ta Edit va 2 ta Button (Natija va chiqish). Formaga TRadioCroup1 komponentasini joylashtirib, Object inspectordan Items xossasini o'rnatib, qator uchun shartni o'rnatamiz(3.2-rasm).



3.2-rasm. Formaning ko'rinishi.

Unit1.H fayl oynasida tanlangan komponentalar quyidagi ko'rinishda bo'ladi:

```
#ifndef Unit1H
#define Unit1H
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
//-----
class TForm1 : public TForm
{
__published:      // IDE-managed Components
    TLabel *Label1;
    TLabel *Label2;
    TLabel *Label3;
    TLabel *Label4;
    TEdit *Edit1;
    TEdit *Edit2;
    TEdit *Edit3;
    TButton *Button1;
    TRadioGroup *RadioGroup1;
    TButton *Button2;
    void __fastcall Button1Click(TObject *Sender);
    void __fastcall Button2Click(TObject *Sender);
private:          // User declarations
public:           // User declarations
    __fastcall TForm1(TComponent* Owner);
};
//-----
extern PACKAGE TForm1 *Form1;
//-----
#endif
```

Dastur kodi:

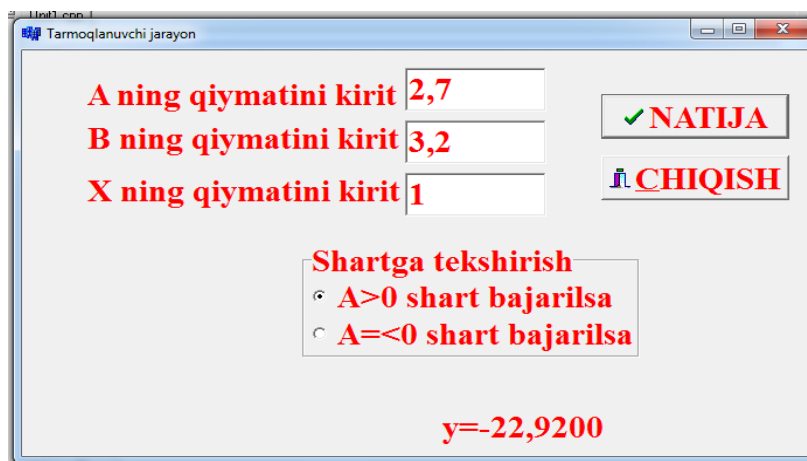
```
//-----
#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
```

```

TForm1 *Form1;
//-----
fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm1::Button1Click(TObject *Sender)
{
    float a,b,x,y;
    a=StrToFloat(Edit1->Text);
    b=StrToFloat(Edit2->Text);
    x=StrToFloat(Edit3->Text);
    if(a>0) { y=3*x*x-3*a*b*x; RadioGroup1->ItemIndex=0;
        Label4->Caption="y="+FloatToStr(y);};
    else {y=13*a-b*b*x; RadioGroup1->ItemIndex=1;
        Label4->Caption="y="+FloatToStr(y);}
}
//-----
void __fastcall TForm1::Button2Click(TObject *Sender)
{
    Close();
}
//-----

```

Natija:



3.4-rasm. Formada natijalar ko'rinishi

Topshiriqlar:

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

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

$$3. c = \begin{cases} f(x)^2 + y^2 + \sin(y), & x - y = 0 \\ (f(x) - y)^2 + \cos(y), & x - y > 0 \\ (y - f(x))^2 + \operatorname{tg}(y), & x - y < 0. \end{cases}$$

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 TEXNIK MASALALARNI YECHISHDA TAKRORLANISH OPERATORLARINI QO'LLASH

Ishning maqsadi: Texnik masalalarni yechishda foydalaniladigan takrorlanish operatorlari bilan tanishib chiqish. Takrorlanuvchi tarkibli algortmlar dasturini tuzishni o'rganish. Takrorlanish operatorlaridan foydalanish ko'nikmalarini egallash.

Reja:

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 (inializasiya; ifoda; modifikasiyalar) operator;

Inisializasiya, 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.

Modifikasiyalar har bir takrorlanish iteratsiyasidan keyin bajarilib, odatda u takrorlanish parametrlarini o'zgartirish uchun xizmat qiladi. Modifikatsiya bo'limida bir nechta 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.

Hisoblang $y = \sum_{i=1}^{50} i^3 + 1$, bu yerda i-natural son.

```
//-----
#include <math.h>
#include <vcl.h>
#include<iostream.h>
#include<conio.h>
#pragma hdrstop
//-----
#pragma argsused
int main(int argc, char* argv[])
{ float y;
  y=0;
  for (int i=1;i<=50;i++)
    y=y+pow(i,3)+1;
cout<<"y="<<y<<endl;
  getch(); return 0; }
//-----
```

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 qatro 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. Parametrlilik takrorlanish operatori qachon ishlatiladi?
4. Nima uchun oldshartli va ket shartli takrorlanish operatorlari deyiladi?
5. Takrorlanish ichida takrorlanish nima?

5-AMALIY MASHG‘ULOT

MUHANDISLIK TEXNOLOGIYALARI AMALIYOTIDA STRUKTURAVIY MASALALARNI YECHISHDA MASSIVLARNI QO‘LLASH

Ishning maqsadi: Massivlar bilan ishlash tushunchasini xosil qilish. Muntazam toifadagi ma’lumotlarni dasturlarda ishlatish usullari bilan tanishish. Massivlarni qo‘llash orqali dasturlar tuzishni o‘rganish

Reja:

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 ketma-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 unung 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.

5 ta elementdan iborat bir o‘lchamli massiv berilgan. Massiv elementlarining yig‘indisi topilsin.

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

```

#include<conio.h>
#include <vcl.h>
#pragma hdrstop
//-----
#pragma argsused
int main(int argc, char* argv[])
{
int i, sum;
int a [5] = {7, 6, 3, 2, 6 }; sum = 0;
for (i = 0; i<5; i++) sum += a[i];
cout<<"summa elementov sum= "<<sum;
getch(); return 0;
}
//-----
Dastur konsol rejimida tuzildi.

```

2- misol.

Bir o'lchamli A nomli massiv berilgan. Uning maksimal elementi aniqlansin.

```

//-----
#include<iostream.h>
#include<conio.h>
#include <vcl.h>
#pragma hdrstop
//-----
#pragma argsused
int main(int argc, char* argv[])
{
const n=20;
float a[n]; int k,i, max;
cout<<"kolichestvo elementov massiva ";
cin>>k;
for (i=0; i<k; i++) cin>>a[i];
max=a[0];
for (i=1; i<k; i++) if (a[i]>max) max=a[i];
cout<<"maksimalniy element massiva="<<max;
getch(); return 0;
}
//-----

```

Forma ilovasida massivlarni ishlatish

Forma ilovasida bir o'lchamli va ko'po'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.

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?

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.

6-AMALIY MASHG'ULOT

DASTURLARGA STRUKTURAVIY YONDASHGAN XOLDA

DASTUR YARATISH USLUBIYATINI O'RGANISH

Ishning maqsadi: Masalalarni kompyuterda yechish dasturini ixchamlashtirish maqsadida funksiyalar bilan ishlashni o'rganish. Funksiyalardan foydalanib dasturlar tuzish.

Reja:

1. Nazariy qism bilan tanishish.
2. Funksiyalar bilan ishlashni o'rganish.
3. Funksiyalar yordamida dasturlar tuzish.

Nazariy qism

Amaliyotda shunday masalalar ham uchraydiki, ularni yechish jarayonida birgina funktsiyani hisoblash emas, balki bir nechta amallar ketma-ketligini o'zgaruvchi kattalikning turli qiymatlarida bir necha marotaba takrorlashga to'g'ri keladi.

Bunday masalalarni kompyuterda yechish dasturini ixchamlashtirish maqsadida takrorlanayotgan buyruqlarni dasturda bir marotaba yozib, kerak bo'lganda unga istalgancha murojaat qilish mumkin.

Dasturning turli joyidan murojaat qilinishi natijasida ko'p marotaba bajarilishi mumkin bo'lgan operatorlarning to'plami "protsedura" (qism dastur), protsedura joylashgan dastur "asosiy dastur" deb yuritiladi.

C++ tilida qism dasturlarni ikki xil ko'rinishda tashkil qilish mumkin: protsedura va funksiya ko'rinishida. Bu ko'rinishdagi qism dasturlar asosiy dastur tarkibida joylashgan bo'lib, ular ham o'z sarlavhasiga va tarkibiga (operatorlar to'plamiga) ega bo'ladi.

Funksiya – bu mantiqan to'g'ri tugatilgan dasturiy qismidir. Ular yordamida katta va murakkab hisoblashlarni qayta-qayta yozish mashaqqatidan halos bo'linadi va dastur bajarilishi yengillashadi. Uni bir marta tashkil etib yozib qo'yiladi va unga dasturning istalgan yeridan murojaat qilish mumkin bo'ladi. Funksiyani tashkil qilishda funksiyaning toifasi, uning nomi va tashkil etuvchi

parametrlari haqida axborot keltiriladi. Bu parametrlar rasmiy parametrlar deb yuritiladi, ularning qiymati funksiyaga murojaat qilish vaqtida aniqlanadi.

Funksiya umumiy ko‘rinishda quyidagicha yoziladi:

```
    funksiya toifasi funksiya nomi (rasmiy parametrlar)
    {
        funksiya tanasi;
    }
```

Funksiya nomi ixtiyoriy lotincha so‘z bo‘lishi mumkin, rasmiy parametrlar – ixtiyoriy o‘zgaruvchilardir.

Funksiyaga murojaat qilishdan oldin rasmiy parametrlar o‘rniga keladigan haqiqiy parametrlar aniqlanishi lozim. Unga murojaat qilish quyidagicha bo‘ladi:

o‘zgaruvchi = funksiya nomi (haqiqiy parametrlar);

Rasmiy va haqiqiy parametrlar soni, ularning toifasi va kelish tartibi albatta bir biriga mos bo‘lishi lozim. Rasmiy va haqiqiy parametrlar nomlari bir xil bo‘lishi mumkin. Funksiyani bosh funksiya ichida e‘lon qilinganida haqiqiy parametrlar nomlarini ko‘rsatmasdan, faqat ularning toifalarini keltirish ham mumkin, masalan: float max (float, float);

Asosiy dasturda ham funksiyada ishlatish mumkin bo‘lgan o‘zgaruvchilar global o‘zgaruvchilar deyiladi. Global o‘zgaruvchilar asosiy dasturda e‘lon qilinishi shart. Faqat funksiyaning ichida ishlatish mumkin bo‘lgan o‘zgaruvchilarga lokal o‘zgaruvchilar deyiladi. Ular funksiyada e‘lon qilinadi.

Funksiyalar main () funktsiyasidan avval ham, keyin ham aniqlanishi mumkin. Agar bosh funksiyadan avval aniqlangan bo‘lsa, uni main () funksiyasi ichida alohida e‘lon qilish shart emas, agar bosh funksiyadan keyin keladigan bo‘lsa, uni main () funksiyasi ichida albatta e‘lon qilish kerak. Masalan quyidagi dasturda “funk” nomli funksiya main bosh funktsiyasidan keyin e‘lon qilingan va aniqlangan:

```
#include <vcl.h>
#pragma hdrstop
#include <math.h>
# include <iostream.h>
# include <conio.h>
//-----
#pragma argsused
int main(int argc, char* argv[])
{ int k, n, funk (int n);
  cin >>n;
  k=funk(n);
  cout << "k="<<k<<endl;
  getch( );
}
int funk(int a) // funksiyani aniqlash
{ int c;
```

```

    c=1+sin(a)+cos(a);
    return c; // funksiyaga natijani qaytarish
}

```

Yuqorida keltirilgan funksiya tanasini quyidagicha ham yozish mumkin:

```

int funk(int a)
{
return 1+sin(a)+cos(a);
}

```

Funksiya bosh funksiyadan oldin e'lon qilinsa dastur quyidagi ko'rinishda yozilishi mumkin:

```

//-----
#include <vcl.h>
#pragma hdrstop
#include <math.h>
# include <iostream.h>
# include <conio.h>
//-----
#pragma argsused
int funk (int a)
{
    return 1+sin(a)+cos(a);
}
int main(int argc, char* argv[])
{ int k, n ;
  cin>>n;
  k=funk(n);
  cout << "k="<<k<<endl;
  getch( );   return 0;
}
//-----

```

Demak, funksiyalarga murojaat qilish jarayonida amallar quyidagi tartibda bajariladi:

1. Funksiya bajarilayotganda rasmiy parametrlar uchun xotiradan joy ajratiladi, ya'ni ular funksiyaning ichki parametrlariga aylantiriladi. Bunda parametr toifasi o'zgartiriladi: float toifasi double toifasiga, char va shortint toifalari int toifasiga aylantiriladi.

2. Haqiqiy parametrlar qiymatlari beriladi yoki hisoblanadi.

3. Haqiqiy parametrlar rasmiy parametrlar uchun ajratilgan xotira qismiga yoziladi.

4. Funksiya tanasi ichki parametrlar yordamida bajariladi va qiymat qaytarish joyiga yuboriladi.

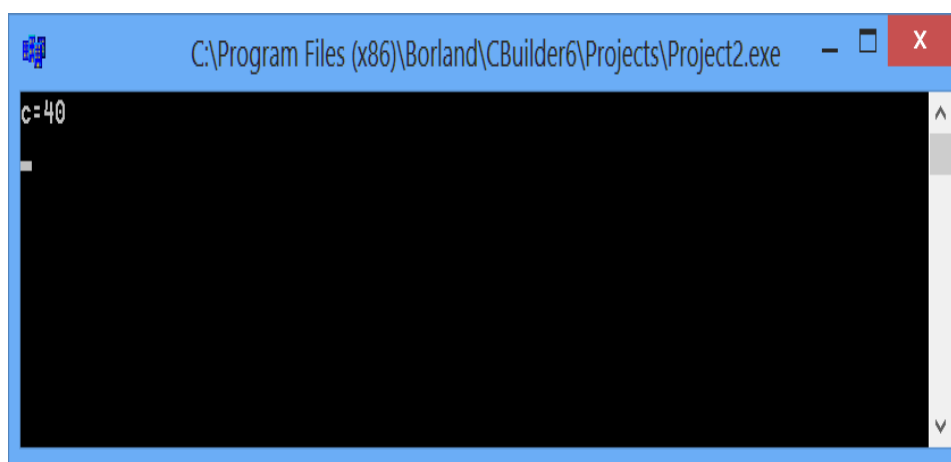
5. Funksiyadan chiqishda rasmiy parametrlar uchun ajratilgan xotira qismi bo'shatiladi.

1- misol.

Ikkita ixtiyoriy sonlar ichidan kattasini topish uchun funksiya tashkil qilinsin. Dasturning konsol rejimidagi ko‘rinishi quyidagicha:

```
//-----  
#include <vcl.h>  
#pragma hdrstop  
#include <math.h>  
# include <iostream.h>  
# include <conio.h>  
//-----  
#pragma argsused  
int main(int argc, char* argv[])  
{  
float x=20, b=40, c, max(float x, float y);  
    c = max(x, b);  
    cout << "c="<<c<<endl;  
    getch( );  
}  
float max(float x, float y)  
{  
if (x > y) return x;  
    else return y;  
    return 0;  
}  
//-----
```

Dastur matni kiritib bo‘lingandan so‘ng F9 tugmasi bosiladi va natijada quyidagi dastur ko‘rinishi paydo bo‘ladi:



6.1-rasm. Natija oynasi

Ushbu misolda max funksiyasini e'lon qilishda rasmiy parametrlar ko'rsatilmagan, funksiya tanasida 2 ta return operatori ishlatilgan.

Sinov savollari:

1. C++ tilida qism-dastur tushunchasi va ulardan foydalanish.
2. Funksiyaning dasturdagi o'rni.
3. Funksiyalarga qanday murojaat qilinadi?
4. Haqiqiy parametrlar va ulardan foydalanish.
5. Funksiya va uning xususiyatlari.

7-AMALIY MASHG'ULOT C++ DA FAYL TOIFASIDAGI MA'LUMOTLARDAN FOYDALANISH

Ishning maqsadi: C++ da fayl toifasidagi ma'lumotlardan foydalanish bo'yicha tushunchalarga ega bo'lish. Fayllar bilan ishlashni o'rganish va ma'limotlarning faylli toifasi yordamida dasturlar tuzishni amaliy bajarish.

Reja:

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

Nazariy qism

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

Ularni dasturda qo'llash uchun *#include* direktivasi yordamida dasturning boshlang'ich matniga mos e'lonlardan tashkil topgan sarlavha fayllarni (zagolovochno`y fayl) kiritish lozim.

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

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

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

Kiritishni boshqarishda ma'lumotlar baytlar oqimi sifatida karaladi. Fizik jihatdan 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 xolda 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).

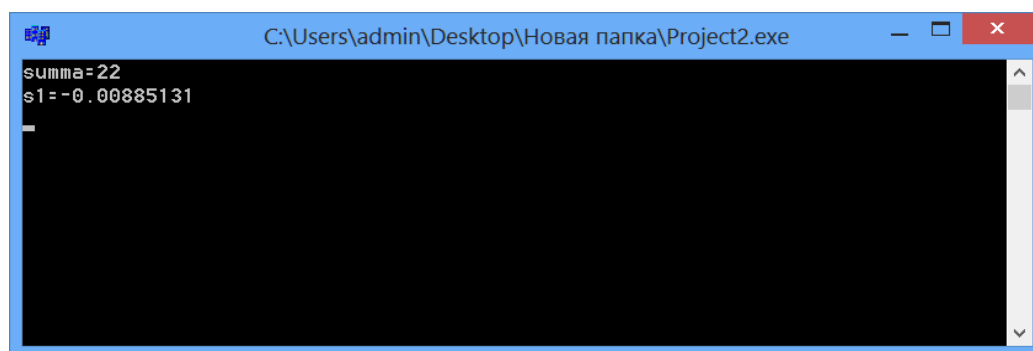
```
#include <conio.h>
#include <iostream.h>
#include <fstream.h>
#include <vcl.h>
#pragma hdrstop
//-----
#pragma argsused
int main(int argc, char* argv[])
{ int a[10], i;
  ofstream file("musbat.txt");
  file<<"A massivning musbat elementlari"<<endl;
  for(i=1;i<=10;i++)
  {
    cout<<"a["<<i<<"]=";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

2-misol. 2 ta butun sonlarning yig'indisini o'zida saqlovchi fayl hosil qilinsin va olingan natijadan foydalanuvchi dastur tuzilsin.

Dasturning konsol rejimidagi ko'rinishi quyidagicha:

```
//-----  
#include <vcl.h>  
# include <iostream.h>  
# include <fstream.h>  
# include <conio.h>  
# include <math.h>  
#pragma hdrstop  
//-----  
#pragma argsused  
int main(int argc, char* argv[])  
{int a=12, b=10;  
int summa;  
float s1;  
ofstream mk ("hujjat.txt");  
summa = a + b;  
cout <<"summa=" << summa << endl;  
mk << summa <<endl;  
mk.close ();  
ifstream mk1 ("hujjat.txt");  
mk1 >>summa;  
s1 = sin (summa);  
cout <<"s1="<<s1<<endl;  
mk1.close ();  
getch ();  
    return 0;  
}  
//-----
```

Dastur matni kiritib bo'lingandan so'ng F9 tugmasi bosiladi va natijada quyidagi dastur ko'rinishi paydo bo'ladi:



7.1-rasm. Natija oynasi

3-misol. Matrisa va vektorlar berilgan. Son qiymatlari ixtiyoriy. Ushbu qiymatlardan foydalanib, matrisani vektorga ko‘paytirish, matrisaning izini hisoblash va vektorning yig‘indisini hisoblash dasturini tuzing.

Dasturning konsol rejimidagi ko‘rinishi quyidagicha:

```
//-----  
#include <vcl.h>  
# include <iostream.h>  
# include <fstream.h>  
# include <conio.h>  
# include <math.h>  
#pragma hdrstop  
//-----  
#pragma argsused  
int main(int argc, char* argv[])  
{  
int a [3][3], b[3], c[3];  
int i, j;  
int s1=0, s2=0;  
ofstream said ("mk.txt");  
for ( i=0; i<3; i++)  
{  
for (j=0; j<3; j++)  
{  
a[i][j] = random (9);  
said <<a[i][j]<<endl;  
}  
}  
for (i=0; i<3; i++)  
{  
b[i] = random (11);  
said << b[i]<<endl;  
}  
said.close ( );  
ifstream said1 ("mk.txt");  
for ( i=0; i<3; i++)  
for (j=0; j<3; j++)  
said1 >>a[i][j];  
for ( i=0; i<3; i++)  
said1 >> 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<<endl;
cout <<"s2="<<s2<<endl;
getch ( );
return 0;
}
//-----

```

Dastur matni kiritib bo‘lingandan so‘ng F9 tugmasi bosiladi va natijada quyidagi dastur ko‘rinishi paydo bo‘ladi:



```

C:\Users\admin\Desktop\Новая папка\Project2.exe
c=72
c=20
c=80
s1=10
s2=18

```

7.2-rasm. Natija oynasi

Nazorat savollari:

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

8-AMALIY MASHG'ULOT

C++ TIZIMIDAGI GRAFIK IMKONIYATLARDAN FOYDALANISH

Ishning maqsadi: Ilovada grafik tasvirlarni yaratishda C++ Builder 6 muhitining asosiy imkoniyatlarini o'rganish. Grafik tasvirlarni yaratishda komponentalardan foydalanish va dasturlar tuzish ko'nikmalarini o'rganish.

Reja:

1. Ilovada grafik tasvirlarni yaratishda C++ Builder 6 muhitining asosiy imkoniyatlari
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;
- Poligon(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 aylanasimon to‘rtburchak;
- StRoundSquare – chetlari aylanasimon 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.

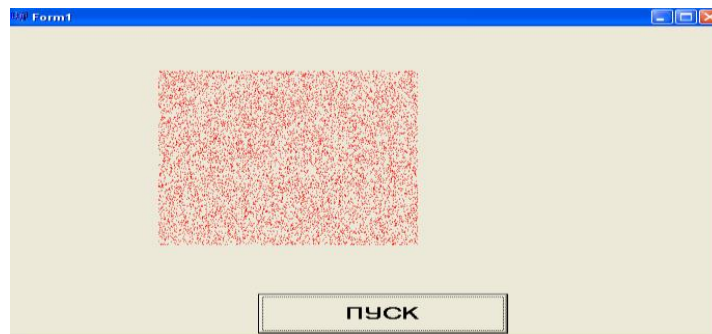
Vizual rejimda dastur kodi:

```
#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); }
}
//-----

```



8.1-rasm. Natija oynasi

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)
{
}

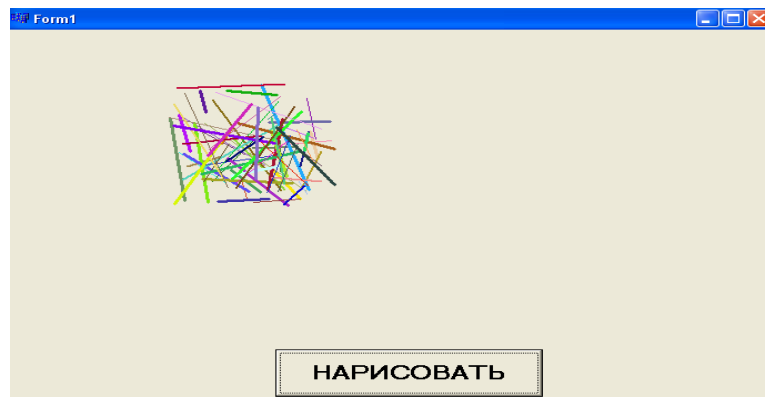
```



```

//-----
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);
}
//-----

```



8.2-rasm. Natija oynasi

1-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); }
    }
//-----

```

Nazorat savollari:

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

Foydalanilgan adabiyotlar

1. Kadirov M.M. Axborot texnologiyalari. O‘quv qo‘llanma, 1-qism. – T.: Sano-standart, 2018. - 320 b.
2. Kadirov M.M. Texnik tizimlarda axborot texnologiyalari. Darslik, 2-qism. –T.: Fan va texnologiya, 2018. - 288 b.
3. Kunwoo Lee. Principles of CAD/CAM/CAE: The Computer Aided Engineering Design Series. 5st Edition. Addison Wesley Longman, USA, 2015.
4. Alex Allain. Jumping into C++. USA, 2014. p 340.
5. Nazirov Sh.A., Qobulov R.V., Bobojonov M.R., Raxmanov Q.S. C va C++ tili. Darslik. –T.: Voris, 2013. - 488 b.
6. Ganiyev S.K., Karimov M.M., Tashev K.A. Axborot xavfsizligi. Oliy o‘quv yurti talabalari uchun darslik. –T.: Fan va texnologiya, 2017. - 372 b.
7. Каримова Н.О. Информационные технологии в технических системах , Часть 1. Fan va texnologiya matba uyi, 2020. – 340 стр.
8. Xaljigitov A.A., Madrahimov Sh.F., Adamboyev U.E. Informatika va programmash. O‘quv qo‘llanma., O‘zMU , 2005-yil. - 145 bet.
9. “O‘zbekiston Respublikasini yanada rivojlantirish bo‘yicha Harakatlar strategiyasi to‘g‘risida” O‘zbekiston Respublikasi Prezidentining Farmoni. 2017-yil 7-fevral, PF-4947-son.
10. Kenjabayev A.T., Ikramov A., Allanazarov A,Sh. Axborot-kommunikatsiya texnologiyalari. O‘quv qo‘llanma. –T.: O‘zbekiston faylasuflari milliy jamiyati nashriyoti, 2017. - 408 b.
11. Nazirov Sh.A., Ivanova G.S., Gaynazarov S.M. Dasturlash texnologiyalari. Darslik. –T.: O‘zbekiston faylasuflari milliy jamiyati nashriyoti, 2014. - 280 b.
12. Akbarova SH.A. Texnik tizimlarda axborot texnologiyalari. Darslik. - T.: Fan va texnologiyalar nashriyot matbaa uyi, 2022.-310 b.

Mundarija:

1	Matematik modellarni algoritmlarda ifoda etish.....	3
2	Muxandislik masalalarini yechishda c++ konstruksiylaridan foydalanish.....	9
3	Mantiqiy dasturlashni misollarda qo‘llash.....	14
4	Texnik masalarni yechishda takrorlanish operatorlarini qo‘llash.....	22
5	Muxandislik texnologiyalari amaliyotida strukturaviy masalalarni yechishda massivlarni qo‘llash.....	26
6	Dasturlarga strukturaviy yondashgan xolda yaratish uslubiyatini o‘rganish.....	29
7	C++ da fayl toifasidagi ma’lumotlardan foydalanish..	33
8	C++ tizimidagi grafik imkoniyatlarida foydalanish...	38
	Foydalanilgan adabiyotlar.....	43

Muharrir: Alimova S. A.
Musahhih: Adilkhadjayeva Sh. M.