

O‘ZBEKISTON RESPUBLIKASI
OLIY VA O‘RTA MAXSUS TA‘LIM VAZIRLIGI
QARSHI MUHANDISLIK – IQTISODIYOT INSTITUTI
“AXBOROT TEXNOLOGIYALARI” KAFEDRASI



ENERGETIKA FAKULTETI
INFORMATIKA VA AXBOROT TEXNOLOGIYALARI
FANIDAN

KURS ISHI

MAVZU: Delphi Dasturlashtirish Tilining Asosiy Operatorlari
Bajaruvchi: *IE-134 guruh talabasi*
Z.Qurbonov
Ilmiy rahbar: *B.Xoliqulov*

Qarshi-2015 y

KURS ISHI

Fannig nomi: “Informatika va axborot texnologiyalari”

Talaba: *Qurbonov Zafar Zikir o'g'li* **IE-134** guruh

1. Kurs ishining mavzusi: Delphi dasturlashtirish tilining asosiy operatorlari

2. Kurs ishi uchun boshlang'ich ma'lumotlar:

2.1. Asosiy qisim

- *Delphi dasturlash muhiti. Operator tushunchasi*
- *Ma'lumotlarni kiritish va chiqarish operatorlari*
- *Delphida konsol ilovasini yaratish*
- *Shartli o'tish operatori*
- *Shartsiz o'tish va tanlash operatorlari*
- *Sikl operatorlari*
- *Massivlar*

2.2 Amaliy hisoblash qism.(Ilovada keltirilgan)

3. Adabiyotlar:

1. Karimov I.A. Yuksak ma'naviyat-yengilmas kuch.T: Ma'naviyat, 2008 y.
2. Ochilov M., Jo'rayev F. “Kurs ishlarini bajarish bo'yicha uslubiy ko'rsatma” .2015y
3. G'ulomov S.S va boshqalar. Axborot tizimlari va texnologiyalari. T: Sharq, 2000y
4. Mamatov A.Z. Delphi tilida dasturlash asoslari. O'quv qo'llanma, Toshkent, 2006 y
5. Aripov M.M. va boshqalar. Axborot texnologiyalari. Toshkent: 2009y
6. Abduqodirov A.A. Turbo Paskal tilida programmalash. O'quv qullanma. Toshkent, 1999y
7. Faysman A. Profetsialnoe programmirovaniye na yazыke Paskal. T: 1992

Elektron resurslar.

1. <http://www.dials.ru>
2. <http://www.cotfrum.ru>
3. <http://www.ziyonet.uz>
4. <http://www.dastur.uz>

4. Ishlab chiqilishi lozim bo'lgan savollar ro'yxati:

(Delphi dasturlash muhiti haqida boshlang'ich ma'lumotlar, Operatorlar, Ma'lumotlarni kiritish va chiqarish operatorlari, Delphida konsol ilovasini yaratish, Shartli o'tish operatori, Shartsiz o'tish va tanlash operatorlari, Sikl operatorlari, Massivlar)

5. Bajarilishi shart bo'lgan chizma va grafiklar:

(Masalani yechishning algortmini bilok sxema ko'rinishda tasvirlash, Pascal dasturi muhitida tuzilgan dastur matni tasviri, tuzilgan dastur natijasining dasturlash tili oynasidagi tasviri, Matlab va MathCad dasturida bajarilgan topshiriqning natijaviy tasviri, Jadvalli miqdorlarning MS Excel dastur oynasida aks etgan tasviri)

O'ZBEKISTON RESPUBLIKASI OLIIY VA O'RTA MAXSUS TA'LIM VAZIRLIGI

QARSHI MUHANDISLIK-IQTISODIYOT INSTITUTI

Fakultet: ENERGETIKA FAKULTETI 1-KURS

Talaba: QURBONOV ZAFAR ZIKIR O'G'LI IE-134 guruh

Fanning nomi: INFORMATIKA VA AXBOROT TEXNOLOGIYALARI

Kurs ishi olingan sana « » yil

Kurs ishi qaytarilgan sana « » yil

Ball _____
(sinovdan o'tdi, sinovdan o'tmadi)

TAQRIZ

[illegible]

KIRISH

Kompyuter dunyosida ko'plab dasturlash tillari mavjud. Bir xil turdagi ishni bajaradigan dasturlarni Basic, Pascal, C++, kabi tillarda yozish mumkin. Lekin, qaysi dasturlash tili yaxshi? Bu savolga javob berish oddiy emas. Biroq shuni ishonch bilan aytish mumkinki, Pascal tili boshqa tillarga qaraganda dastur tuzishni o'rganish uchun ancha qulay til bo'lib hisoblanadi.

Pascal tili Shvesariyalik olim N.Virt tamonidan yaratilib, keyinchalik Borland korporasiyasi tamonidan rivjlantirildi. Bu til rivjlantirilib Turbo Pascal, Borland Pascal va keyinchalik esa Object Pascal nomini oldi. Hozirgi kunda Object Pascal tili asosi bo'lgan Windows muhitida ishlovchi Delphi dasturiy vositasida murakkab professional dasturlar ishlab chiqilmoqda.

Kompyuterda dasturlash oxirgi yillarda juda tez rivojlanib dastur tuzushga qiziquvchilar soni oshib bormoqda. 10-15 yil oldin o'z dasturlarini Windows muhitida yaratish ko'pgina dasturchilarning orzusi edi. Delphi dasturlash vositasining yaratilishi esa nafaqat professional dasturchilar, balki oddiy dastur tuzuvchilar uchun ham keng yo'l ochib berdi.

2.1.Ma'lumotlarni kiritish va chiqarish operatorlari

Biror bir masalani yechishning chiziqli bo'lgan algoritmgiga dastur tuzishda algoritmdagi keltirilgan ketma-ketliklar asosida operatorlar yoziladi. Bunday dasturlarni tuzushda asosan o'zgaruvchilar qiymatini kiritish, natijalarni chiqarish va shu bilan birga o'zlashtirish operatorlari ishlatiladi.

Dasturdagi o'zgaruvchilar qiymatlarini dastur ichida o'zlashtirish operatori yordamida ham berish mumkin. Lekin dasturda o'zgaruvchi qiymatini tashqaridan kiritish qulaylik tug'diradi va umumiylikni ta'minlaydi.

Read operatori o'zgaruvchilar qiymatlarini ekrandan kompyuter xotirasiga kiritish uchun ishlatiladi. U quyidagi ko'rinishlarga ega.

Read(c1,c2,...,cn);

Readln(c1,c2,...,cn);

Readln;

bu erda c1,c2,...,cn - o'zgaruvchilar nomi; ln - qo'shimchasi qiymatni kiritib keyingi qatorga o'tishni bildiradi.

Misollar: Read(Sm1,Sm2);

Readln(x1,x2,x3);

Readln;

Bu erda birinchi operator Sm1 va Sm2 o'zgaruvchilar qiymatini ekrandan kiritadi. Ikkinchi operator esa x1,x2,x3 o'zgaruvchilar qiymatini ekrandan kiritadi va kiritishni keyingi qatorga o'tkazadi. Oxirgi operator esa kiritishni kutadi va qator o'tkazadi.

Write operatori oddiy ma'lumotlarni va o'zgaruvchilar qiymatlarini kompyuter ekraniga chiqarish uchun ishlatiladi. U quyidagi ko'rinishlarga ega.

Write(c1,c2,...,cn);

Writeln(c1,c2,...,cn);

Writeln;

bu erda c1,c2,...,cn - oddiy matnlar yoki o'zgaruvchilar nomi; ln - qo'shimchasi chiqarishni keyingi qatorga o'tishni bildiradi.

Misollar: Write(Summa);

Write('Natija yuk');

Write('Tenglama echimi x1=', x1, 'x2=', x2);

Oddiy ma'lumotlarni chiqarishda ular matn deb qaraladi va u qo'shtirnoq ichida yoziladi. CHiqarish operatori yordamida o'zgaruvchilar qiymatini format ko'rinishda ham berish mumkin:

Write(c:m:n);

bu erda s-o'zgaruvchi; m-shu o'zgaruvchi qiymati uzunligi; n-qiymatning kasr qismi va unda $n-1 < m$ bo'lishi kerak.

Misol. Write(x:8:4);

Agar $x=155.01021$ bo'lsa, quyidagi yozuv chiqadi 115.0102.

Write('Maxsulot soni:', kol:5);

Agar kol=15 bo'lsa, quyidagi yozuv ekranga chiqadi,

Maxsulot soni: 15

Dastur matnini tushuntirish maqsadida ko'pincha dasturda izohlar keltiriladi. Dasturda izohlar istalgan joyda berilishi mumkin. Izoh katta qavs ichida yoziladi.

Masalan: { Bu matn dasturga izoh beradi }

{ Bu joyda echim aniqlanmoqda }

Dasturda ma'lum hisoblashlarning natijalarini biror bir o'zgaruvchida saqlash uchun o'zlashtirish (yuborish) operatori ishlatilib, u «:=» belgisi yordamida qiymat yuborilishi kerak bo'lgan o'zgaruvchidan keyin qo'yiladi.

Masalan: $i:=0$; i-qiymati nolga tenglashadi, ya'ni i o'zgaruvchiga nol yuboriladi deb tushuniladi. Bunda mashina i o'zgaruvchi uchun ajratilgan xotirasiga nol yozib saqlaydi.

Misol: $B:=5$; $C:=4$; $A:=(B+C)/2$;

Bu erda, agar A butun identifikator bo'lsa, uning qiymati 4 ga, aks holda esa 4.5 qiymatga ega bo'ladi.

CHiziqli strukturali algoritmlarni dastur shaklida yozish uchun oldin ishlatiladigan o'zgaruvchilar ruyxati keltirilib, keyin algoritmdagi bajarilishlar ketma-ket ravishda amalga oshirilishi kerak.

Misol: Tekislikdagi ikki nuqta orasidagi masofani topish dasturi.

Program Misol_AT;

Var

x1,y1,x2,y2,d: Real;

Begin

Write('Nukta koordinatalarini kiriting:');

Read(x1,y1,x2,y2);

d:=Sqrt(Sqr(x1-x2)+Sqr(y1-y2));

Writeln;

Writeln('Nukta koordinatalari:',x1,y1,x2,y2);

Writeln('Masofa=',d);

Readln;

End.

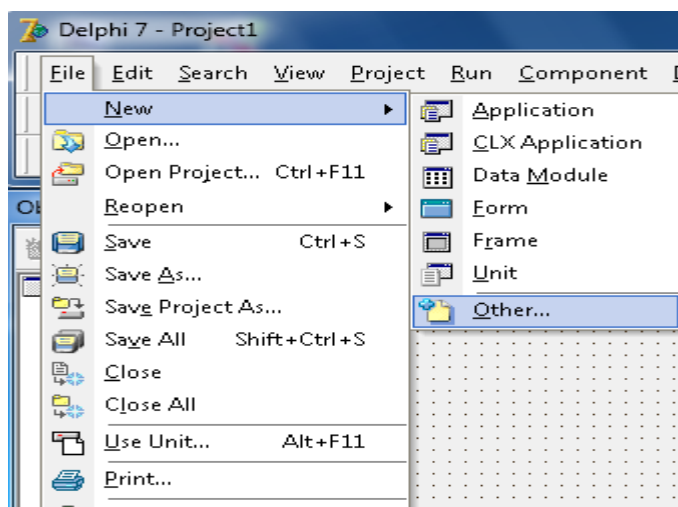
2.2.Delphida konsol ilovasini yaratish

Delphida konsol ilovalarini har xil usullarda yaratish mumkin. Ulardan eng oson usuli quyidagicha:

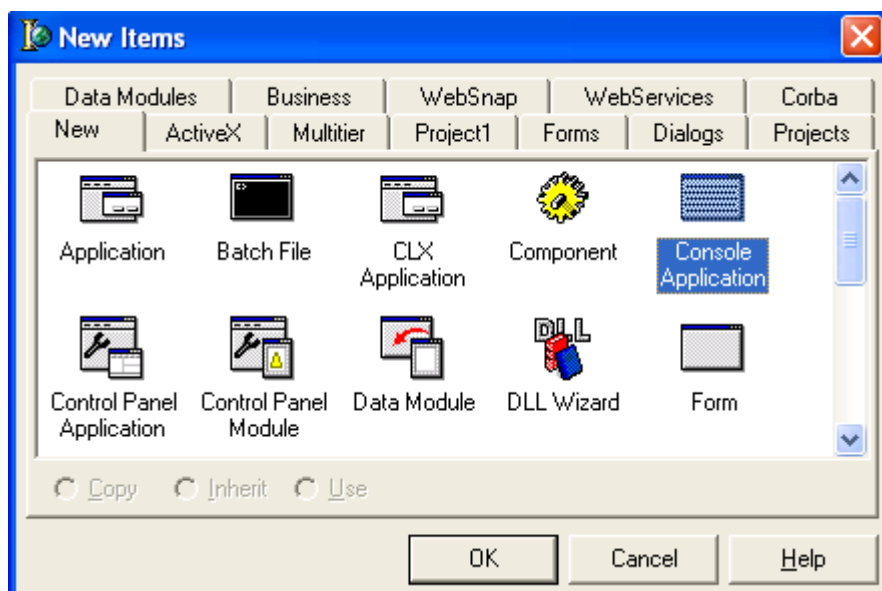
1.Delphi muhiti ishga tushiriladi.

Pusk=>Программы=>Borland Delphi 7=> Delphi 7

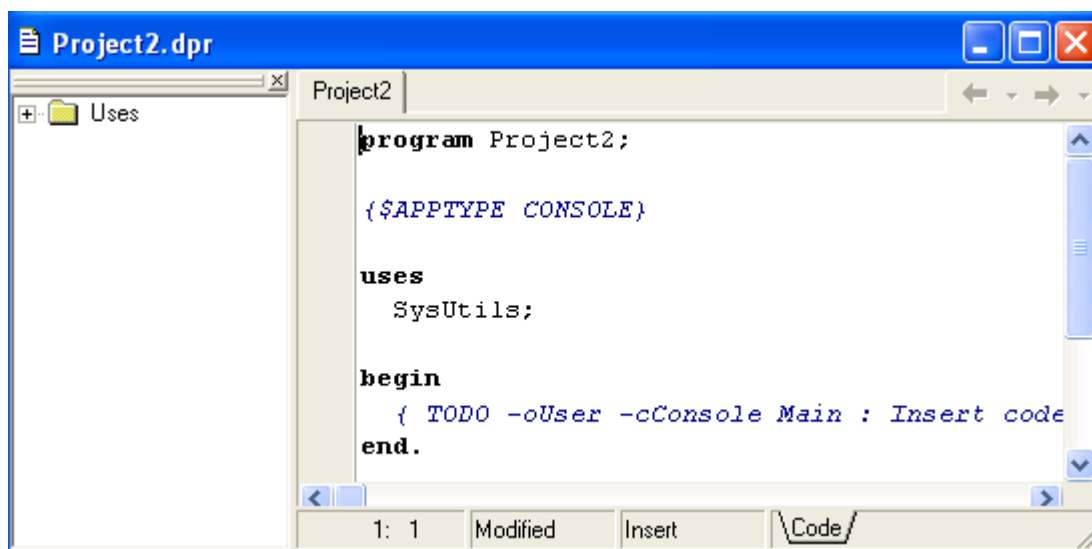
2.Bosh menyudan File punktini ochib u erdan New, keyin esa Other buyruqlari beriladi. **File=> New=> Other**



3.Forma va loyihalarni saqlash uchun ochilgan maxsus oynadan (bu oynaga Delphi arxiv oynasi deyiladi) “Console Application” piktogrammasi tanlanadi va Ok tugmasi bosiladi.



4. Natijada ekranda loyiha oynasi ochiladi (.dpr kungaytmali nom bilan).



Begin – end ichiga olingan

{ TODO -oUser -cConsole Main : Insert code here }

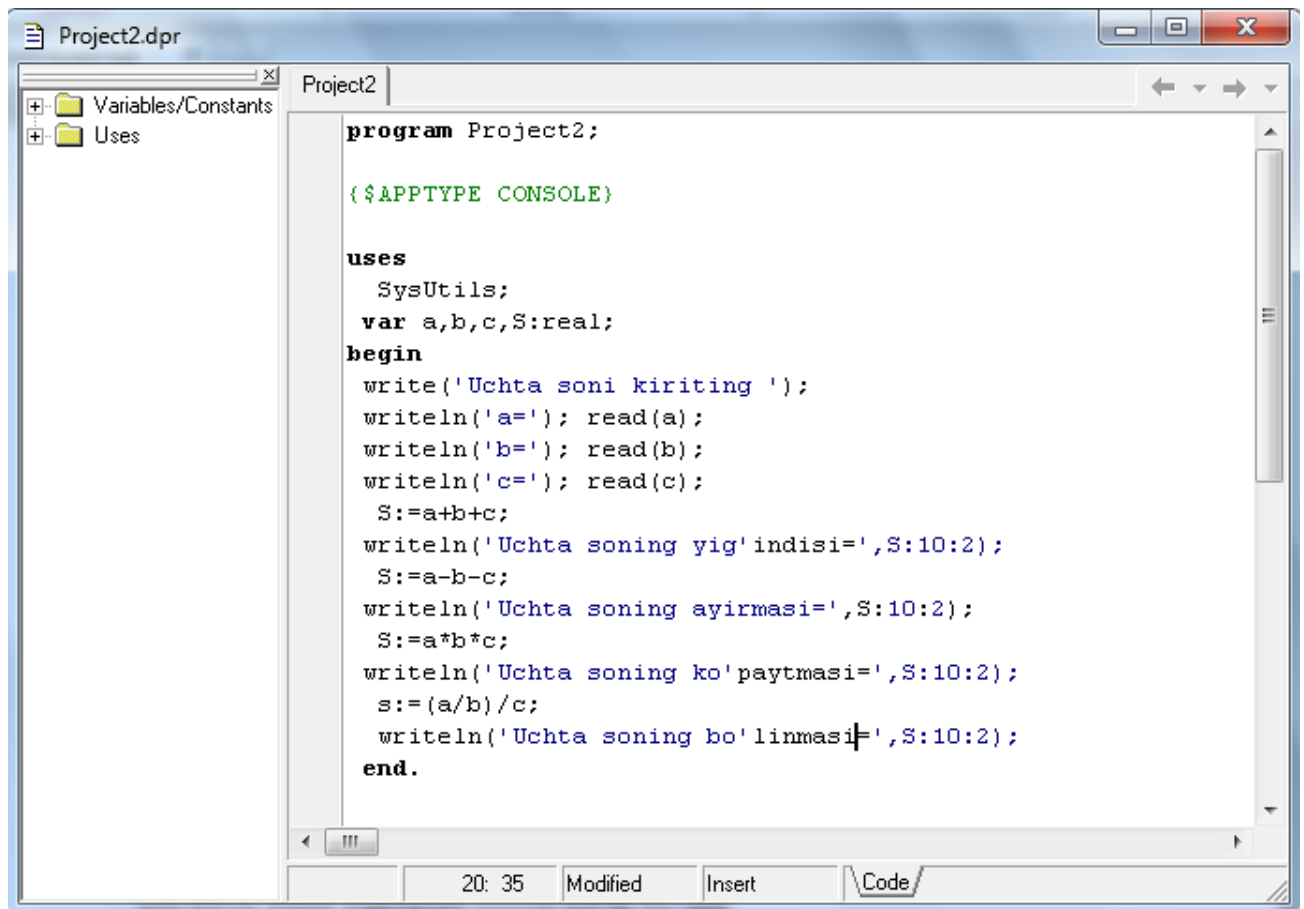
izoh o‘rniga loyiha faylining dastur matni kiritiladi.

Tuzilgan dasturni ishga tushirishdan oldin uni saqlash kerak bo‘ladi. Uni saqlash file=>Save All buyrug‘ini berish lozim. Har bir loyiha alohida yangi papkaga saqlanishni tavsiya beradi. Loyiha faylini saqlashda alohida kursatilmagan holatida ProjectN.dpr nomli fayl nomini tavsiya qiladi. Bu erda N har bir ketma ket nomlanadigan loyiha nomeri (son, masalan 1,2,3,..). Lekin biz loyiha faylini istalgan nom bilan saqlashimiz mumkin. Masalan MyProgram.dpr. Bu nom avtomatik ravishda chiqadi.

Loyihani saqlab bo‘lgandan so‘ng, uni bajarishga beramiz. Buning uchun bosh menyudan quyidagi buyruqni berish lozim: Run=>Run yoki F9 funksional tugmachasini bosish kerak bo‘ladi. Dastur normal ishga tushgandan so‘ng ekranda DOSning standart dastur oynasi namayon bo‘ladi.

Misol. Ucha sonning yig‘indisi, ayirmasi, ko‘paytmasi va bo‘linmasini hisoblash dasturini yarating.

Bu misolni yechish uchun yuqorida keltirilgan to‘rtta ketma ketlikni bajaramiz va dastur kodini kiritamiz.

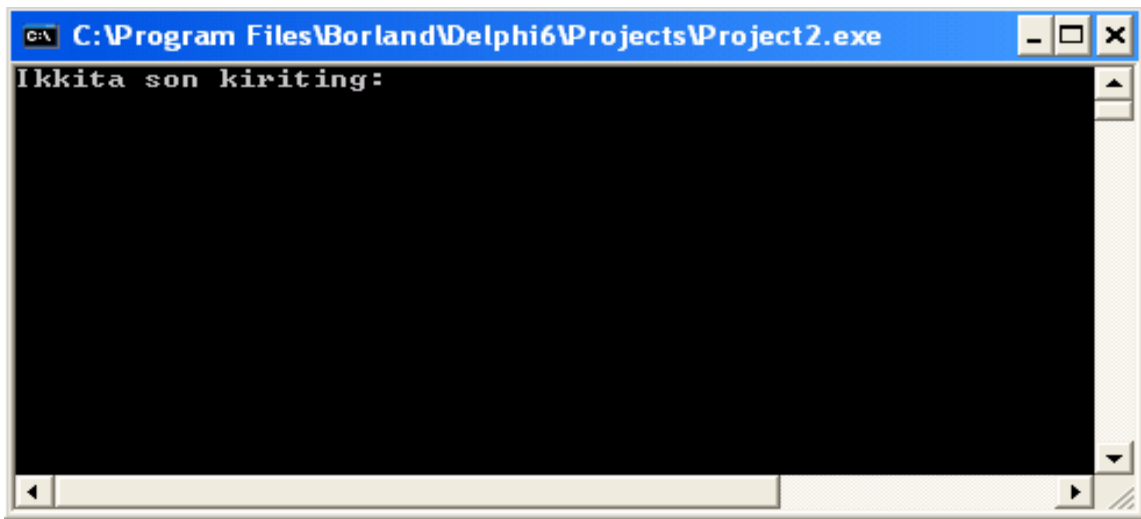


```
program Project2;

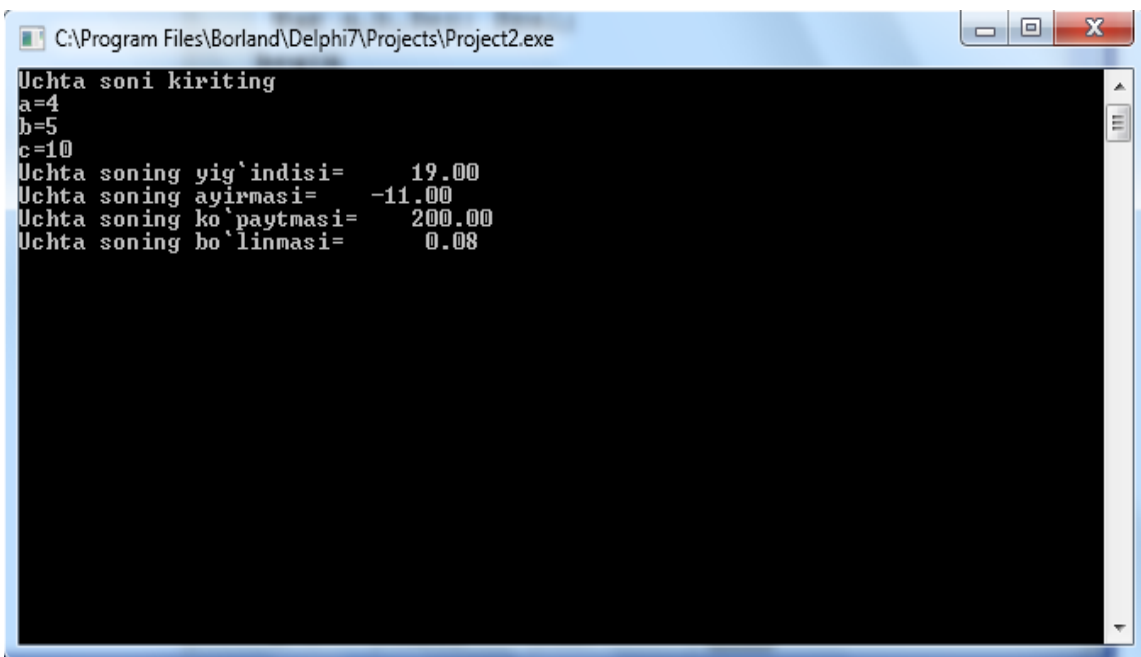
{$APPTYPE CONSOLE}

uses
  SysUtils;
var
  a,b,c,S:real;
begin
  write('Ucha soni kiriting ');
  writeln('a='); read(a);
  writeln('b='); read(b);
  writeln('c='); read(c);
  S:=a+b+c;
  writeln('Ucha soning yig'indisi=',S:10:2);
  S:=a-b-c;
  writeln('Ucha soning ayirmasi=',S:10:2);
  S:=a*b*c;
  writeln('Ucha soning ko'paytmasi=',S:10:2);
  s:=(a/b)/c;
  writeln('Ucha soning bo'linmasi=',S:10:2);
end.
```

Dastur kodi kiritilgandan so‘ng uni saqlab keyin ishga tushiramiz. Natijada ekranda Dos oynasi ochilib unda “Ikkita son kiriting:” so‘zi chiqadi. Keyin ikkita son kiritib Enter tugmasini bosish kerak bo‘ladi.



Natijada quyidagi javoblar chiqadi.



5.Keyin Obekt inspektori (Object Inspector) va Ob'ekt daraxtlar (Object TreeView) oynalari yopiladi.

6.Bosh menyudan Project=>View Source buyrug'i beriladi.

Endi Delphida konsul ilovasini yaratishning ikkinchi usulini ko'rib chiqamiz:

1.Delphi muhiti ishga tushiriladi.

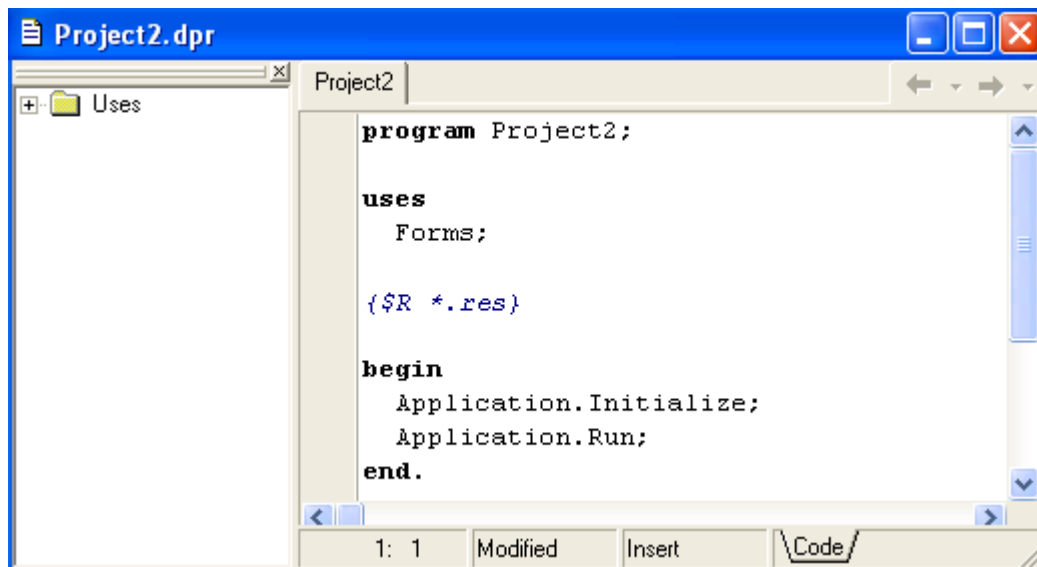
Pusk=>Программы=>Borland Delphi

2.Bosh menyudan File punktini ochib u erdan New, keyin esa Application buyruqlari beriladi. **File=> New=> Application**

3.Forma oynasi yopiladi.

4.Dastur kodini yozish (modul) oynasi yopiladi. Yopish vaqtida “Save changes to Unit1.pas?” (“Unit1.pasdagi o‘zgarishlar saqlansinmi?”) so‘rov oynasi chiqadi. U erdan “**no**” (**yo‘q**) buyrug‘i beriladi.

Natijada quyidagi Project1.drp loyiha fayli oynasi ekranga chiqadi.



7.Loyiha fayli agar lozim bo‘lsa boshqa nom bilan saqlangadi.

Bu oynadan Program, Uses, Begin va End kalit so‘zlari qoldirilib boshqalari o‘chiriladi va keyin dastur matn kodlari kiritiladi.

Agar loyiha fayli yozilgan papka ichi qaralsa unda quyidagi fayllar ro‘yxatini ko‘ramiz.

- MyProgram.dpr -loyiha fayli (bosh loyiha moduli);
- MyProgram.exe -ilova fayli yoki bajariluvchi fayl. Bu fayl kompilyator yordamida, ya’ni kompilyasiya jarayonida, agar dasturda sintaktik xatoliklar bo‘lmasa tuziladi. Boshqacha so‘z bilan aytganda, agar sizga o‘z dasturingizni ishga tushurish mumkin bo‘lsa, masalan F9 tugmasini bosish bilan bajariluvchi fayl avtomatik ravishda tuziladi. Bajariluvchi fayl avtonom fayl bo‘lib uning uchun boshqa fayl yoki biror dasturiy sistema mavjud bo‘lishi shart emas. Uni siz ishga tushirishingiz mumkin boshqa dasturlar kabi, masalan Paint, Bloknot yoki o‘yin dasturlarini ishga tushurganday;
- MyProgram.cfg -loyiha konfiguratsiyasi fayli;

- MyProgram.dof -loyiha opsiya fayli. Unda dasturning to'g'ri ishlaganligi haqida axborotlar saqlanadi,

Loyiha opsiya va konfiguratsiyasi fayllari loyiha faylining tuzilishi bilan bir vaqtda Delphi tomonidan avtomatik ravishda tuziladi. Ko'p hollarda yuqorida keltirilgan fayllardan tashqari yana .dpr kengaytmali fayl ham tuziladi. Bu fayl loyiha faylining (rezerv fayli) nusxasi bo'lib hisoblanadi. Masalan, MyProgram.-dpr. Bu fayl loyiha fayli tuzilishi davrida bir vaqtning o'zida tuzib boriladi. Agar asosiy loyiha faylida buzilish yoki uchirilish sodir bo'ladigan bo'lsa, u holda uni MyProgram.-dpr faylidan tiklash mumkin. Buning uchun kengaytma oldidagi "-" belgini olib tashlash kifoya.

2.3.Shartli o'tish operatori

Pascal tilida shart - bu mantiqiy turdagi ifoda bo'lib, u faqat «chin»(True) yoki «yolg'on»(False) qiymatni qabul qiladi.

Quyidagi mantiqiy belgilar ishlatiladi: >, <,<=,>=,<>,. Bularga munosabat amallari deyiladi.

Quyidagi mantiqiy amallar ishlatiladi:

- **NOT**-«inkor»;
- **AND**-«mantiqiy va»;
- **OR**-«mantiqiy yoki».

Bu mantiqiy amallarning bajarilish natijalari quyidagicha:

Op1	Op2	Op1 AND Op2	Op1 OR Op2	NOT Op1
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

Masalan: $(3 < 4)$ **AND** $(15 < 50)$ -mantiqiy ifoda rost (**True**),
 $(40 > 0)$ **OR** $(40 < 0.5)$ -mantiqiy ifoda rost (**True**),
 $(10 < 8)$ **AND** $(10 < 15)$ -mantiqiy ifoda yolg'on (**False**),
NOT $(100 > 3)$ -mantiqiy ifoda yolg'on (**False**).

Mantiqiy ifodalarni biror bir mantiqiy o'zgaruvchiga yuborish ham mumkin.

Masalan:

F:=(A<B) AND (A<C);

Bu erda, agar ikkala shart bajarilgandagina F mantiqiy o'zgaruvchi "chin" (True) qiymatni qabul qiladi. Aks holda "yolg'on" (False) qiymatni qabul qiladi.

Pascal tilida Shartli o'tish operatorining ikki xil ko'rinishi mavjud: to'liq va qisqa.

To'liq ko'rinish:

If <shart> then Begin

<shart rost bo'lganda bajariladigan operatorlar>

end

else

Begin

<shart yolg'on bo'lganda bajariladigan operatorlar>

end;

Qisqa ko'rinish:

If <shart> then Begin

<shart rost bo'lganda bajariladigan operatorlar>

end;

Bu erda **if** -agar; **then** -u holda; **else** -aks holda ma'nosini bildiruvchi xizmatchi (kalit) so'zlar.

Birinchi ko'rinishdagi Shartli operatorda, agar shart bajarilsa birinchi Begin va end ichidagi operatorlar ketma-ket bajariladi, aks holda ikkinchi Begin va end ichidagi operatorlar ketma-ket bajariladi.

Ikkinchi ko'rinishdagi Shartli operator quyidagicha ishlaydi. Agar berilgan shart bajarilsa Begin va end ichidagi operatorlar ketma-ket bajariladi, aks holda ular bajarilmaydi.

Agar bajariluvchi operatorlar soni bitta bo'lsa **Begin** va **end** so'zlarini yozish shart emas.

Misollar:

1) **If A>0 Then Begin C:=1; B:=C+1; end**
else Begin C:=0; B:=4; end;

2) **If D=A Then D:=A else A:=D;**

O‘ar bir Shartli o‘tish operatori ichida boshqa ichki Shartli operatorlar joylashishi ham mumkin, masalan.

If b1 then a1 else If b2 then a2 else a3;

Misollar.

A:=0.5; B:=-1.7; if A<B then A:=B else B:=A;

Javob: $0.5 < -1.7$ yolg‘on bo‘lganligi sababli $B:=A$ operator bajariladi, va bunda $A=0,5$ va $B=0,5$ ekenligi kelib chiqadi.

A:=0.1; B:=0.1; C:=0.5; D:=0;

if (A<B) OR (A>C) Then D:=B+C else

if B=A then Begin D:=C; C:=A; end;

Javob: $(0.1 < 0.1)$ yoki $(0.1 > 0.5)$ bu mantiqiy ifoda yolg‘on bo‘lganligi sababli $B=A$ shart tekshiriladi. Bu shart chin bo‘lganligi sabab $D=0,5$ ga, $S=0,1$ qiymatlarga teng ekenligi kelib chiqadi.

2.4.Shartsiz o‘tish va tanlash operatorlari

Dasturda shunday holatlar bo‘ladiki operatorlarning bajarilish shartiga qarab dasturning u yoki bu qismiga to‘g‘ridan-to‘g‘ri o‘tishga to‘g‘ri keladi. Bunday holatlarda Shartsiz o‘tish operatoridan foydalanish mumkin.

Shartsiz o‘tish operatorining ko‘rinishi quyidagicha:

Goto n;

Bu erda n -belgi(metka) bo‘lib identifikator yoki butun son bo‘lishi mumkin. Goto - o‘tish ma’nosini bildiradi.

n - belgi dasturning bosh qismida Label so‘zi yordamida e’lon qilingan bo‘lishi shart. n boshqarilish uzatiladigan joyga n : shaklida qo‘yiladi.

Misol:

.....

Goto L2;

.....

L2: $C:=x*y$;

.....

Ko'p hollarda baror bir parametrlarning qiymatiga qarab kerakli operatorlarni bajarishga to'g'ri keladi. Bunday hollarda tanlash operatorini ishlatgan qulay. Tanlash operatori ko'rinishi quyidagicha bo'ladi:

Case s of

1: A1;

2: A2;

.....

n: An;

else Begin

<B1,B2,..Bn>

end;

end;

Bu erda **Case** -xizmatchi so'z bo'lib tanlash ma'nosini beradi; **of** -«dan» ma'nosini beradi; s-operator selektori; 1,2,..n-operator belgilari; A1,A2,...An va B1,B2,...Bn-operatorlar.

Case operatori tarmoqlanish jarayonida berilgan bir necha operatoridan birini tanlash yo'li bilan amalga oshiradi. Operatorlar ketma-ketligini tanlash operator selektorining qiymatiga qarab aniqlanadi. Operator selektori haqiqiy bo'lmagan o'zgaruvchi yoki ifoda bo'lishi mumkin. Agar operator selektori qiymati operator belgilari o'zgarmas qiymatiga teng bo'lmasa B1,B2,...Bn-operatorlari ketma-ket bajariladi.

Shartli o'tish operatorining quyidagi ko'rinishi

if B Then A1 else A2;

tanlash operatorining quyidagi operatoriga ekvivalentdir.

Case B of

true: A1;

false: A2;

end;

Misol

$ax^2+bx+c=0$ kvadrat tenglamaning ildizlarini topish dasturi tuzilsin.

program Misol_AT;

```

Label 20;
Var a, b, c, d, e, f, x, x1, x2, z: real;
begin
    Writeln ('a, b, c koeffitsientlar qiymatini kiriting:');
    read (a, b, c);
    if A=0 then
        begin
            x:=-b/c; writeln (x);
            goto 20
        end
    else
        begin
            d:=b*b-4.0*a*c; z:=2.0*a; e:=-b/z;
f:=sqrt(abs(d)/Z;
        end;
        if d>=0 then
            begin
                x1:=e+f; x2:=e-f; writeln (x1,x2);
            end
        else if d=0 then
            begin x:=e; writeln(x); end else writeln ('yechim yo‘q’);
20: end.

```

2.5.Sikl operatorlari.

Ayrim masalalarda bir yoki bir necha parametrlarning o‘zgarishiga qarab ma’lum hisoblashlar bir necha marta takrorlanib bajarilishi mumkin. Masalan, $y=ax+b$ funksiyani x ning bir necha qiymatida uning mos qiymatlarini hisoblash kerak deylik. Bunday hisoblashlarni kompyuterda dastur tuzib bajarish uchun siklik strukturali dasturlar tuzish kerak bo‘ladi. Bu kabi dasturlarni Shartli o‘tish operatori yordamida ham tuzish mumkin. Lekin Paskal tilida siklik strukturali dastur tuzish uchun bir necha maxsus operatorlar mavjud.

For operatori takrorlanishlar soni aniq bo'lgan sikllik jarayonlar tashkil etishda ishlatiladi. Uning umumiy ko'rinishi quyidagicha: **For i:=m1 to m2 Do S;**

Bu erda i-sikl parametri; m1,m2 -i parametrining boshlanQich va oxirgi qiymati bo'lib, ular o'zgarmas son yoki ifoda bo'lishi mumkin; S-sikl tanasi bo'lib, bir necha operatorlardan tashkil topishi mumkin.

Agar sikl tanasi bir necha operatoridan iborat bo'lsa ular **Begin** va **End** ichiga olinadi.

Misol. 1,2,...100 conlar yig'indisini hisoblash dastursini tuzing.

Program Misol_AT;

```
const kn=10;
var i: integer; s: real;
begin
  s:=0;
  for i:=1 to kn do s:=s+i;
  write ('s=',s); readln;
end.
```

Agar **to** so'zni **Downto** so'ziga almashtirilsa sikl parametri teskari bo'yicha o'zgaradi, ya'ni -1 qadam bilan. U holda sikl ko'rinishi quyidagicha bo'ladi.

For i:=m1 Downto m2 do ;

Misol. 100 dan 1 gacha conlarni ekranga chiqarish dastursini tuzing.

Program Misol_AT;

```
var i: integer;
begin
  for i:=10 downto 1 do write (i); readln;
end.
```

While sikl operatori takrorlanishlar soni oldindan aniq bo'lmagan hollarda takrorlanishni biror bir shart asosida bajaradi. Berilgan shart oldin tekshiriladi va keyin shartning bajarilishiga qarab kerakli operatorlar ketma-ketligi bajariladi. Bu operatorning umumiy ko'rinishi quyidagicha: **While B do S;**

Bu erda B -mantiqiy ifoda; S -sikl tanasi bo‘lib, bir yoki bir necha operatorlar ketma-ketligidan iborat bo‘lishi mumkin. Mantiqiy ifoda ‘**True**’ yoki ‘**False**’ qiymat qabul qiladi.

Agar mantiqiy ifoda ‘True’ qiymat qabul qilsa S operatorlari bajariladi, aks holda bajarilmaydi, ya’ni sikl ishlashdan to‘xtaydi.

Misol. 1,2,...,10 conlar yig‘indisini hisoblash dastursini tuzing.

Program Misol_AT

```
Const k=10;  
Var i: Integer; S: real;  
Begin  
  S:=0; i:=0;  
  While i<=k do Begin i:=i+1; S:=S+i; end;  
  Write ('S=',S);  
  Readln;  
end.
```

Repeat sikl operatori ham takrorlanishlar soni oldindan aniq bo‘lmagan hollarda takrorlanishni biror bir shart asosida bajaradi. Oldin sikl tanasidagi operatorlar ketma-ketligi bajariladi. Berilgan shart keyin tekshiriladi. Agar berilgan shart rost (True) bo‘lsa, boshqaruv sikldan keyingi operatorni bajarishga o‘tadi, aks holda sikl takrorlanadi. Bu operatorning umumiy ko‘rinishi quyidagicha:

Repeat

S

Until B

Bu erda B -mantiqiy ifoda, ‘True’ yoki ‘False’ qiymat qabul qiladi; S -sikl tanasi bo‘lib, bir yoki bir necha operatorlar ketma-ketligidan iborat bo‘lishi mumkin. Agar mantiqiy ifoda ‘False’ qiymat qabul qilsa siklda takrorlanish davom etadi, aks holda to‘xtaydi.

Misol. 1,2,...,10 conlar yig‘indisini hisoblash dastursini tuzing.

program Misol_AT;

```

const k=10;
var i: integer; s: real;
begin
  s:=0; i:=0;
  repeat
    i:=i+1; s:=s+i;;
  until i>k;
  write ('s=',s);
  readln;
end.

```

Odatda **WHILE** operatori **REPEAT** operatoriga nisbatan ko‘p ishlatiladi. Bunga sabab ko‘pchilik masalalarda sikl tugallanish sharti sikl boshlanmasdan tekshirish maqsadga muvofiqdar. Zarur bo‘lsa siklni umuman bajarmasdan o‘tish mumkin.

Ko‘pchilik masalalarni yechishda tuzilgan dasturda ichma-ich joylashgan sikllar tashkil etishga to‘g‘ri keladi. Bunday sikllarga murakkab sikllar deyiladi. Murakkab sikllar tashkil etilganda quyidagi talablar bajarilishi zarur.

- ichki sikl tashqi sikl ichida to‘liq yotishi kerak;
- sikllar bir-biri bilan kesishmasligi kerak;
- sikl ichiga tashqaridan to‘g‘ridan-to‘g‘ri kirish mumkin emas;
- sikl parametrlari boshqa-boshqa identifikatorlar bilan belgila-nishi kerak;

Misol. $S = \sum_{i=1}^{10} \prod_{j=1}^5 \frac{i+j}{\sqrt{i \cdot j}}$ ifodani hisoblash dastursini tuzing.

Bu formulada agar yig‘indini ochsak u quyidagi ko‘rinishga keladi.

$$S = \sum_{i=1}^{10} \prod_{j=1}^5 \frac{i+j}{\sqrt{i \cdot j}} = \prod_{j=1}^5 \frac{1+j}{\sqrt{1 \cdot j}} + \prod_{j=1}^5 \frac{2+j}{\sqrt{2 \cdot j}} + \dots + \prod_{j=1}^5 \frac{10+j}{\sqrt{10 \cdot j}}$$

Program Misol_AT;

```

var
  i,j: integer; s: real;
begin
  s:=0;

```

```

for i:=1 to 10 do
  begin
    p:=1;
    for j:=1 to 5 do p:=p*(i+j)/sqrt(i*j);
    s:=s+p;
  end; write ('s=',s);
end.

```

2.6. Massivlar

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

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

Bir o'lchovli massivda uning har bir elementi o'zining joylashgan o'rin nomeri bilan aniqlanadi va nomeri qavs ichida indeks bilan yoziladi. Ikki o'lchovli massiv elementi o'zi joylashgan satr va ustun nomerlari yordamida aniqlanadi. SHu sabab ikki o'lchamli massiv elementi ikkita indeks orqali yoziladi. Masalan: $A[i,j]$ bu erda i-satr nomeri j-ustun nomerini bildiradi.

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

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

Masalan:

a,b:Array[1..100] of real;

c,k,d:Array[1..10,1..15] of real;

Bu erda a va b massivlari 100tadan elementga ega. c,k,d massivlari esa $10 \times 15 = 150$ tadan elementga ega.

Massivlarni e'lon qilishdan maqsad massiv elementlari uchun kompyuter xotirasidan joy ajratishdir.

Massiv elementlari qiymatlarini kiritish uchun sikl operatorlaridan foydalaniladi.

Misol: **For** i:=1 **to** 10 **do** Read(A[i]);

Bu misolda A massivning 10 ta elementi qiymatini ekrandan ketma-ket kiritish kerak bo'ladi. Xuddi shunday massiv qiymatlarini ekranga chiqarish ham mumkin.

Misol: **For** i:=1 **to** 10 **do** Write(A[i]);

Dasturda massiv elementlarini ishlatganda ularning indeksi e'lon qilingan chegaradan chiqib ketmasligi kerak.

Massiv elementlarini tartiblash usullari.

Massivni tartiblashtirishning bir necha usullari (algoritmлари) mavjud. Ulardan quyidagi usullarni qarab chiqamiz:

- tanlash usuli;
- almashtirish usuli.

Tanlash usuli yordamida massivni o'sish bo'yicha tartiblashtirish algoritmi quyidagicha:

1. Massivning birinchi elementidan boshlab qarab chiqilib eng kichik element topiladi.
2. Birinchi element bilan eng kichik element joylari almashtiriladi.
3. Ikkinchi elementidan boshlab qarab chiqilib eng kichik element topiladi.
4. Ikkinchi element bilan eng kichik element joylari almashtiriladi.
5. Bu protsess bitta oxirgi elementgacha takrorlanadi.

Bu algoritm dastursi quyidagicha bo'ladi:

Program Msol_AT;

const size=5;

var i,j,min,k,buf: integer;

a: array[1..size] of integer;

begin

```

writeln ('massivni tartiblashtirish');
write (size:3,'ta massiv elementini kiriting');
for k:=1 to size do read(a[k]);
writeln ('tartiblashtirish');
for i:=1 to size-1 do
    begin
        { kichik elementni topish }
        min:=i;
        for j:=i+1 to size do
            begin
                if a[j]<a[min] then min:=j;
                buf:=a[i];  a[i]:=a[min]; a[min]:=buf;
                for k:=1 to size do write (a[k],' ');
                writeln;
            end;
        end;
    writeln('massiv tartiblashtirildi.');
```

end.

Dastur natijasi:

Massivni tartiblashtirish

5 ta massiv elementini kiriting

12 -3 56 47 10

Tartiblatirish

-3 12 56 47 10

-3 10 56 47 12

-3 10 12 47 56

-3 10 12 47 56

Massiv tartiblashtirildi.

Almashtirish usuli yordamida massiv elementlarini o'sib borishda tartiblashtirish algoritmi quyidagicha:

1. Massivning birinchi elementidan boshlab ketma-ket hamma qo'shni elementlar bir-biri bilan solishtirilib, agar birinchisi ikkinchisidan kichik bo'lsa ular joyi almashtirilib boriladi.

2. Bu protsess davomida kichik qiymatli elementlar massiv boshiga katta elementlar esa oxiriga siljutilib boriladi. SHu sabab bu usul «puzirka» usuli ham deyiladi.

3. Bu protsess massiv elementlar sonidan bitta kam marta takrorlanadi.

Masalan:

3 2 4 5 1 bunda 3 bilan 2 va 5 bilan 1 almashtiriladi.

2 3 4 1 5 bunda 4 bilan 1 almashtiriladi.

2 3 1 4 5 bunda 3 bilan 1 almashtiriladi.

2 1 3 4 5 bunda 2 bilan 1 almashtiriladi.

1 2 3 4 5

Bu algoritmi dastursi quyidagicha bo'ladi:

Program Misol_AT;

const size=5;

var i,j,min,k,buf: integer;

a: array[1..size] of integer;

begin

writeln ('massivni puzirek(kupikcha) usulida tartiblashtirish');

write (size:3,'ta massiv elementini kiriting');

for k:=1 to size do read(a[k]);

writeln ('tartiblatirish');

for i:=1 to size-1 do

begin

for k:=1 to size-1 do

begin

if a[k]>a[k+1] then

```

begin
    buf:=a[k]; a[k]:=a[k+1]; a[k+1]:=buf;
end;
end;
for k:=1 to size do write (a[k], ' ');
writeln;
end;
writeln('massiv tartiblashtirildi.');
```

end.

Dastur natijasi:

Massivni puzirek usulida tartiblashtirish

5 ta massiv elementini kiriting

3 2 4 1 5

Tartiblashtirish

2 3 4 1 5

2 3 1 4 5

2 1 3 4 5

1 2 3 4 5

Massiv tartiblashtirildi.

Massivda eng kichik yoki eng katta elementni izlash algoritmi ma'lumki birinchi element eng kichik (katta) deb olinib keyin boshka elementlar bilan ketma-ket solishtirilib chiqiladi. Solishtirish oxirgi elementgacha bajariladi.

Quyida bu algoritm dastursi keltirilgan:

Program Misol_AT;

var i,min: integer;

a: array[1..10] of integer;

begin

writeln ('massivdan eng kichik elementni izlash');


```
write (' 10-ta massiv elementini kiriting');  
for i:=1 to 10 do read(a[i]);  
    min:=1;  
for i:=2 to 10 do  
    if a[i]<a[min] then min:=i;  
    writeln('izlanayotgan eng kichik element:',a[min]);  
    writeln('element nomeri',min);  
end.
```

Kurs ishga xulosa

Men kurs ishini bajarish davomida shunga amin bo'ldimki, zaminaviy axborot texnologiyalari yildan yiga rivojlanib bormoqda. Ularni o'rganish va ishlab chiqarishga tadbiq etish uchun tinmay o'qib izlanish kerak. Dasturlash davr talabi. Har qanday zamonaviy axborot texnologiyalarinig asosi dasturlashdan iborat. Shunday ekan avvalambor zamonaviy dasturlash tillari (Borland Delphi 7,Borland C++ Bulder 6,JAVA) ni o'rganish va talabalik davridanoq ushbu dasturlash tillarida turli hil dasturlar (o'yin dasturlar, o'quv dasturlar, amaliy dasturlar) yaratish bilim va malakamizni yanada oshishga hizmat qiladi. Men yuqorida berilgan topshiriqni bajarish va uni nazariy jihatdan o'ganish jarayonida Borland Delphi 7 dasturlash muhitida ishlash ko'nikmamni yanada oshirdim. Shu sohada taxsil olar ekanman, bundan keyin ham o'z bilimlarimni mustahkamlash va oshirish men uchun foydadan holi bo'lmaydi.

Foydalanilgan adabiyotlar ro'yhati:

1. Karimov I.A. Yuksak ma'naviyat-yengilmas kuch.T: Ma'naviyat, 2008 y.
2. Ochilov M., Jo'rayev F. "Kurs ishlarini bajarish bo'yicha uslubiy ko'rsatma".2015y
3. G'ulomov S.S va boshqalar. Axborot tizimlari va texnologiyalari. T: Sharq, 2000y
4. Mamatov A.Z. Delphi tilida dasturlash asoslari. O'quv qo'llanma, Toshkent, 2006 y
5. Aripov M.M. va boshqalar. Axborot texnologiyalari. Toshkent: 2009y

6. Abduqodirov A.A. Turbo Paskal tilida programmalash. O'quv qullanma. Toshkent, 1999y
7. Faysman A. Profetsialnoe programmirovaniye na yazyke Paskal. T: 1992

Elektron resurslar:

1. <http://www.dials.ru>
2. <http://www.cotfrum.ru>
3. <http://www.ziyonet.uz>
4. <http://www.dastur.uz>