

Borland®

O'ZBEKISTON RESPUBLIKASI XALQ TA'LIMI
VAZIRLIGI
NAVOIY DAVLAT PEDAGOGIKA INSTITUTI

Delphi™

***Zamonaviy dasturlash tillari
fanidan amaliy mashg'ulot
ishlanmasi***

Navoiy-2010

Ushbu qo`llanma akademik litsey, kasb-hunar kollejlari, oliy o`quv yurtlari talabalari hamda mustaqil foydalanuvchilar uchun mo`ljallangan.

Tuzuvchilar:

dots. Ro`ziyev R.A., o`q. Mirsanov O`.M.

Taqrizchilar:

t.f.n. Nosirova Sh., f.-m.f.n. Xudoyorov Sh.

Kirish

O`quvchi yoshlarni informatika fanidan bilimli qilib tayyorlashni informatika fanining asosiy qismi hisoblangan zamonaviy dasturlash tillari yordamida amalga oshirish maqsadga muvofiq bo`ladi.

Zamonaviy dasturlash tillarining afzallik tomonlari shundan iboratki, o`quvchida bu tillarni o`rganishda bir qancha kichkina dastur va amaliy paketlarni o`rganish zaruriyati tug`iladi, shu izlanishlar natijasi o`quvchining bilim saviyasini oshishiga yordam beradi.

Hozirgi paytda zamonaviy dasturlash tillaridan biri hisoblangan Delphi dasturlash tilidan keng foydalaniladi. Delphi dasturlash tili - bu Windows muhitida ishlashga mo`ljallangan bo`lib, u 1995-yilda Borland kompaniyasi dasturhilari guruhi Chak (Chuck) va Denni (Danny) tomonidan yaratilgan. Mazkur dastur o`zining ba`zi bir xususiyatlari bilan boshqa dasturlash tillaridan ajralib turadi. Shu sababdan bugungi kunda Delphi dasturlash tili akademik litsey, kasb- hunar kollejlari hamda oliy o`quv yurti fan dasturlariga kiritlgan. Ammo, Delphi dasturlash tilini o`qitishda o`zbek tilining lotin alifbosidagi o`quv adabiyotlarning yetishmasligi seziladi. Shuning uchun ushbu uslubiy qo`llanma o`quvchi- yoshlarning foydalanishi uchun tavsiy etilmoqda. Qo`llanma Delphi dasturlash tilini o`rganishni boshlayotganlarga mo`ljallangan bo`lib, u uchta bobdan iborat: birinchi bobda Delphi dasturlash tili va uning imkoniyatlari, ishlash texnologiyalari haqida qisqacha ma`lumot berilgan; ikkinchi bobda Delphi dasturlash tilida matematik masalalarni yechish (chiziqli, tarmoqlanuvchi, takrorlanuvchi) bo`yicha misollar keltirilgan; uchinchi bobda Delphi dasturlash tilida grafiklar va massivlarga oid bir nechta misollar berilgan.

I BOB. DELPHI DASTURLASH TILI VA UNING IMKONIYATLARI

1.1. Delphi dasturlash tili haqida ma'lumot

Delphi dasturlash tili - bu Windows muhitida ishlab chiqilgan bo'lib, 1995 yilda Borland kompaniyasi guruhi dastur tuzuvchilari Chak (Chuck) va Denni (Danny) tomonidan yaratilgan.

Bu til o'zining keng qamrovli imkoniyatlariga egaligi bilan birga, boshqa dasturlash tillaridan o'zining ba'zi bir xususiyatlari bilan ajralib turadi. Borland Delphi ning paydo bo'lishi dasturlashni rivojlantirish tarixida yorqin ko'rinish bo'ldi. Delphi ning dunyoga kelishiga quyidagi tendensiyalar sabab bo'ldi:

- Windows uchun dasturlash va komponentlar texnologiyasi;
- masalalarni yechish uchun ob'ektga yo'naltirilgan usul;
- komponentlar texnologiyasiga asoslangan ilovalarni tez yaratishning vizual muhitlari;
- interpretatsiyadan emas, kompilyatsiyadan foydalanish. Bu shundan iboratki, interpretator bilan ishlashga qaraganda kompilyator bilan ishlash tezligi o'n martalab ustunlikka ega bo'ladi;
- universal usullar yordamida ma'lumotlar bazasi bilan ishlash imkoniyatlarining mavjudligi, masalan, lokal va shu bilan bir qatorda server ma'lumotlari faylidan mijoz - server arxitekturasiga yoki ko'p bosqichli sxemasiga o'tishni ta'minlash.

Borland Delphi yuqorida bayon etilgan tendensiyalarni joriy etish maqsadida yaratilgan. Ammo, uning eng asosiy elementi Object Paskal tili bo'lib hisoblanadi.

Delphi – bu bir qancha muhim texnologiyalar yig'indisidir, ya'ni:

- mashina kodiga o'tkazuvchi yuqori unumdorlikka ega bo'lgan kompilyator mavjudligi;
- komponentlar ob'ekt-orientirlashtirilgan modelga egaligi;
- dasturlarni vizual tarzda yaratish (tuzish) imkoniyatining mavjudligi;
- ma'lumotlar bazasini yaratish (tuzish) imkoniyatlarining yetarliligi.

Bu dasturlash tili ob'ekt - orientirlashtirilgan tillar majmuiga kirib, 1995 yildan hozirgacha bir qancha avlodlari yaratilgan va ularga quyidagilarni aytib o'tish mumkin:

Delphi 1.0, Delphi 2.0, Delphi 3.0, Delphi 4.0, Delphi 5.0, Delphi 6.0, Delphi 7.0, Delphi 8.0

Dastur yuqori unumdorlikka va tezlikka ega bo'lgan kompilyatorga ega. Har bir yangi versiyasi yaratilganda dasturlash tiliga qandaydir qulaylik yoki imkoniyatlilik kiritildi. Ma'lumotlarga ko'ra hozirgi vaqtda bu eng yuqori tezlikka ega bo'lgan kompilyatordir. Masalan: u 486 DX 33 da 1 daqiqada 120 ming qatordan ortiq dastur matnini kompilyatsiya (mashina kodiga o'tkazish) qila oladi. Bundan tashqari Delphi dasturlash tili vizual instrumentlar yig'indisiga ega bo'lib, u esa dasturlash jarayonini yanada osonlashtiradi. Dasturchi dastur tuzish jarayonida xuddi rassom kabi komponentlar palitrasidagi tayyor tugmachalarni (ob'ektlarni) dasturga joylashtirishi mumkin. Mahsulotning standart turi ikkita bazaviy sinflardan tashkil

topgan. Dasturlash tili qobig`i korporativ ma'lumotlar bazasiga ulanishni qo'llab quvvatlovchi (podderjka), dastur tuzishni tezlashtiruvchi RAD (Rapid Application Development) vizual asboblari (instrumentlar) interfeysiga ega. VCL – visual komponentlar kutubxonasi foydalanuvchining standart ob'ektlari, ma'lumotlarni boshqarish ob'ektlari, grafik ob'ektlar, multimedia ob'ektlari, muloqot ob'ektlari, fayllarni boshqarish ob'ektlari, DDE va OLE larni boshqarish ob'ektlaridan tarkib topgandir. Ma'lumotlar bazasi ob'ektlari SQL ma'lumotlar bazasini boshqarish tizimi asosida Borland Database Engineni to'liq o'z ichiga oladi. Shuningdek, dasturlash tili tarkibiga Borland SQLLink ham kiradi. Bu esa Oracle, Sybase, Informix va Interbase ma'lumotlar bazasini boshqarish tizimlariga ruxsatni (dostup) amalga oshiradi. Mahsulotni ikki xil varianti mavjuddir:

1. «Mijoz - Server» (Delphi Client-Server).

2. Delphi for Windows.

«Mijoz - Server» varianti quyidagi imkoniyatlarga ega:

- SQLLinks: Oracle, Sybase, Informix va InterBase larga ruxsat beruvchi (dostup) maxsus drayverlarning mavjudligi;
- InterBase lokal serveri: lokal tarmoqqa ulanmagan kompyuterlar uchun ma'lumotlar bazasini boshqarish tizimlari dasturlarini tuzish;
- Report Smith Client/Server Edition: SQL – server uchun hisobotlar generatorining mavjudligi;
- Visual Query Builder – SQL so'rovlarini tuzish modullarini mavjudligi.

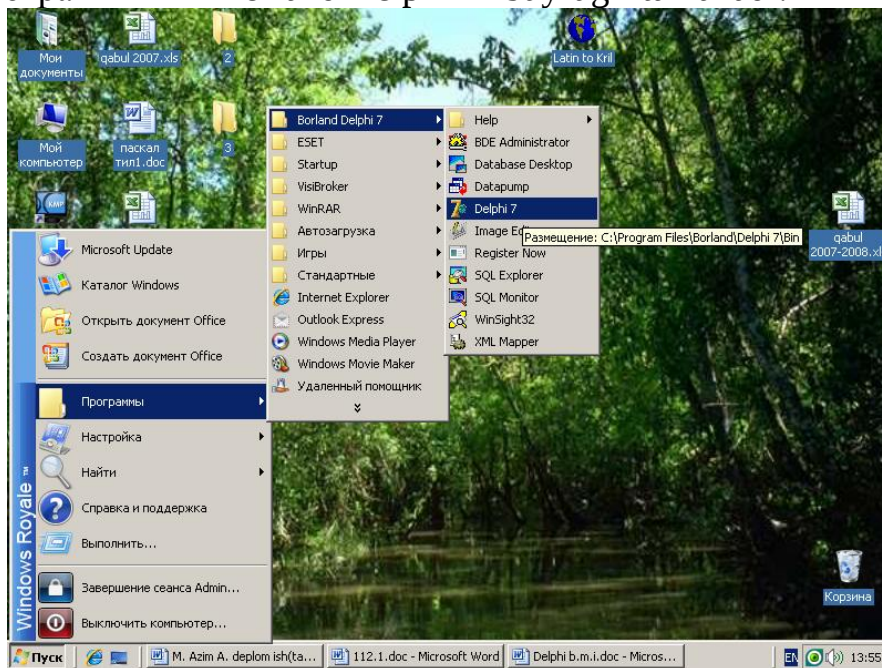
Delphi for Windows varianti personal kompyuterlar uchun dasturlar tuzish hamda dBase va Paradox ma'lumotlar bazasini boshqarish tizimlarida dasturlar tuzish imkoniyatlariga ega. Shuningdek, shaxsiy DLL komponentlarini tuzish imkoniyati ham mavjuddir.

1.2 Delphi dasturni ishga tushirish, dasturni piktogrammalari va menyulari haqida tushincha

Delphi dasturlash tilini ishga tushirishning bir nechta usullari mavjud:

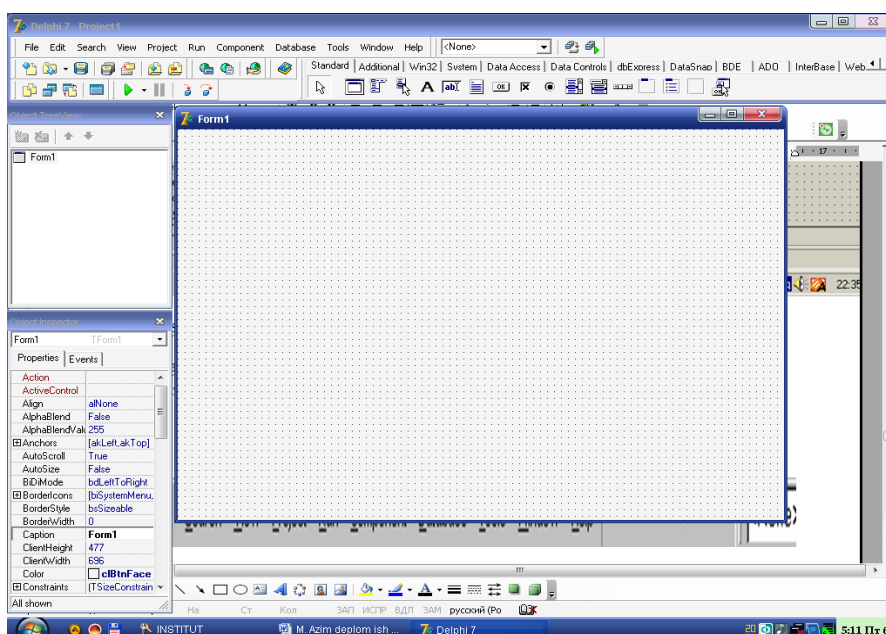
- foydalanuvchining ish stolidagi yorliq yordamida;
- quyi masalalar paneli yordamida;
- bosh menyu yordamida.

Delphi dasturini bosh menyu yordamida ishga tushurishni ko`rib chiqaylik. Agar siz ishlaydigan kompyuterda dastur o`rnatilgan bo`lsa, quyidagicha bo`ladi: “Пуск”+“Программы” + “Borland Delphi 7” buyrug`i tanlanadi.



1-rasm.Delphi dasturlash tilini yuklash

Delphi dasturlash tili ishga tushgandan so`ng quyidagi oyna hosil bo`ladi.



2-rasm.Delphi 7 oynasining to`liq ko`rinish

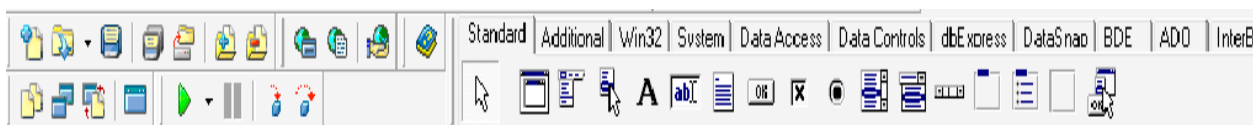
Dasturning umumiy ko`rinishi yuqori sarlavha satri, menyular satri, uskunalar va komponentalar majmui, ob'ektlar inspektori, obektlar daraxti va ishchi sohadan tashkil topgan. Menyular satrining ko`rinishi odatdagi Office dasturlarini singari sarlavha satridan keyin joylashgan. Delphi dasturining o`n bitta menyusi bo`lib ular quyidagilardir:



3-rasm.Menyular satri

Delphi asosiy menyu bandlarining vazifalari Microsoft office standart dasturlarining menyu bandlari vazifalaridan deyarli farqlanmaydi va ulardan foydalanish tartibi aynan saqlanadi.

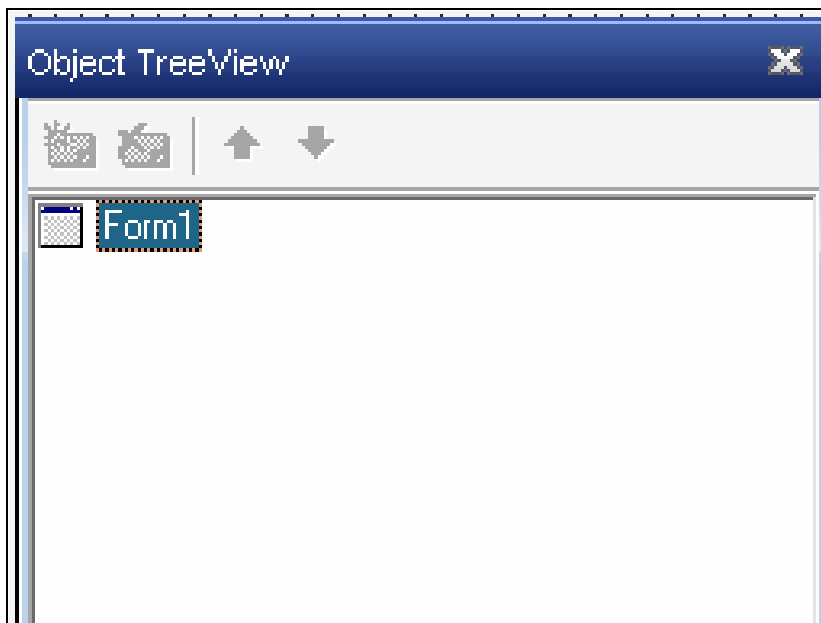
Asboblari paneli va komponentalar ro`yxati Delphi dasturlash tilida komponentalar bajaradigan vazifalariga qarab guruhlariga ajratilgan. Har bir guruh komponentalari o`zining nomiga ega bo`lgan alohida bo`limda joylashgan.



4- rasm.Chap tomonda uskunalar paneli va o`ng tomonda komponentalar palitrasi

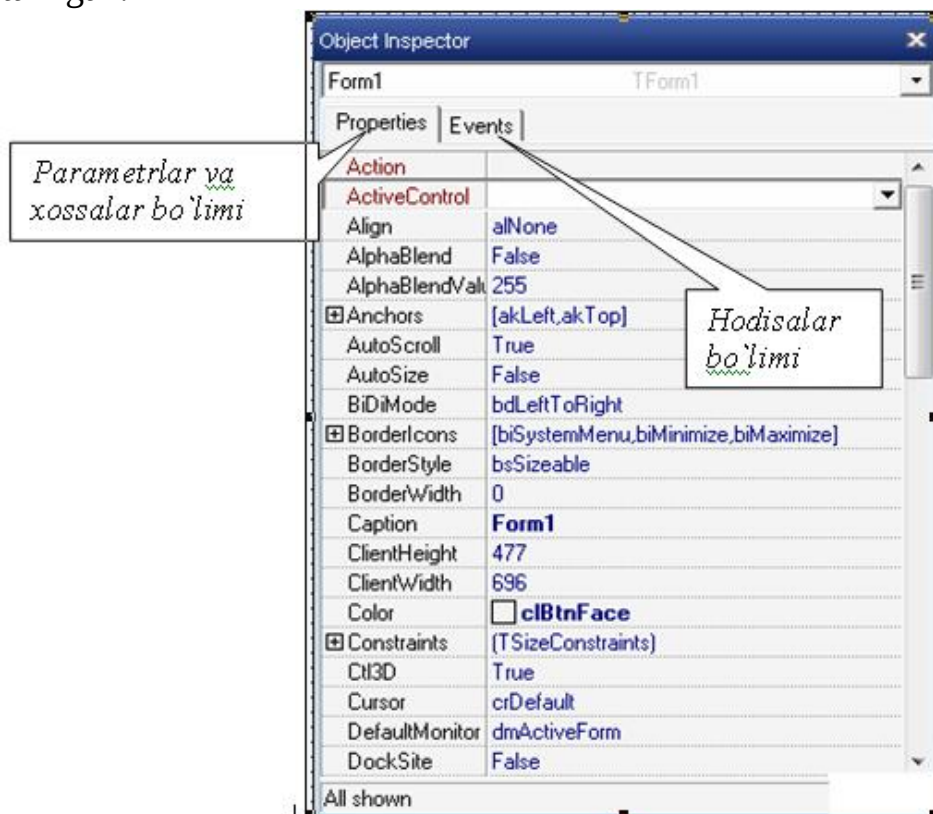
Ko`rinmay turgan bo`lim (komponenta) larni ochish uchun  tasmadan foydalaniladi.

Ob'ektlar daraxti. Ob'ektlar daraxtining ko`rinishi quyidagi tasvirda aks ettirilgan.



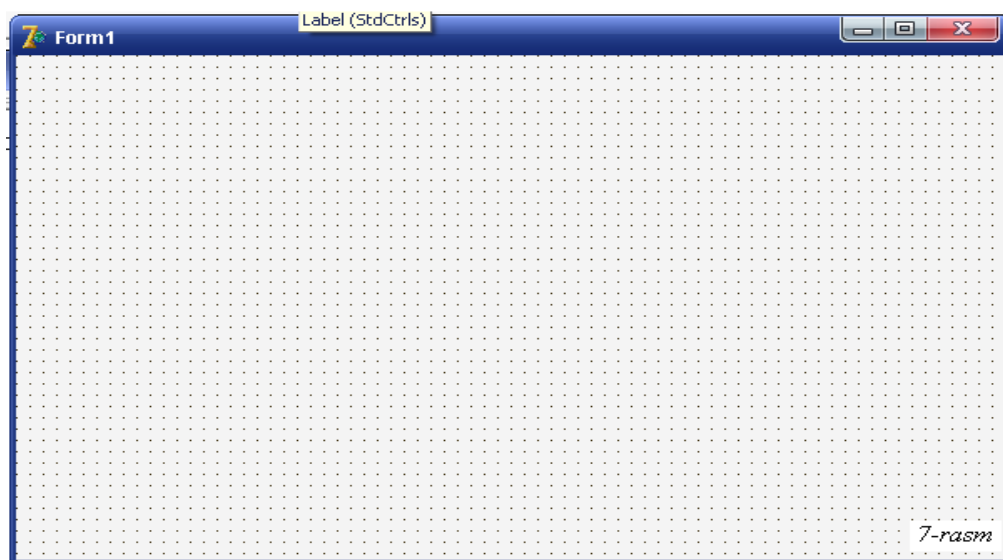
5-rasm.Ob'ektlar daraxti

Ob'ektlar inspektori. Ob'ektlar inspektorining ko`rinishi qo`yidagi tasvirda aks ettirilgan.



6-rasm.Ob`ekt xossalarini tahrirlash oynasi

Formani loyihalashtiruvchi oyna quyidagi rasmda tasvirlangan.



7-rasm.Forma oynasi

Standard jixozlar palitrasi quyidagi komponentalardan iborat:



8-rasm.Komponentalar

-  **Popup Menu** – formada kontekstli menyu hosil qiladi;
-  **Memo** – formada ko`p satrli matn maydonini hosil qiladi;
-  **Edit** – formada bir satrli matn kiritish maydonini hosil qiladi;
-  **Label** – formada yozuv hosil qiladi;
-  **Button** – formada birorta amalni bajaradigan tugmachani hosil qiladi;
-  **Check Box** – formada o`chirish yoki belgilash vazifasini hosil qilish;
-  **Radio Button** – bir nechta variantdan birortasini tanlash imkoniyatini yaratadi
-  **List Box** – formada bir nechta satrlar ro`yxatini tuzish imkonini yaratadi;
-  **Combo Box** – bir nechta satrdan birortasini tanlash imkoni hosil qilish;
-  **Scroll Bar** – formada siljitish vositasini yaratishda ishlatiladi;
-  **Radio Group** – formada bir nechta variantdan birortasi tanlanadigan ro`yxatni tuzish;
-  **Panel** – formada panelni yaratish uchun foydalaniladi.

Additional, Dialogs komponentlar palitrasi. Additional komponentlar palitrasi quyidagi ob'ektlardan tuzilgan



9-rasm.Komponentalar palitrasi

 - **TBitBtn** - TButton ob'ektiga o`xshash ob'ektdir. Bundan tashqari ob'ekt aniqlangan ko`rinishga ega. Masalan: *bkOK*, *bkClose* va boshqalar. Bunda xususiyatdagi **glyph** qatori orqali tugmachaga kerakli rasmni berish mumkin;

 - **TSpeedButton** – SpeedBar panelidagi tugmachalarga o`xshash tugmachalar hosil qilish uchun ishlatilib, tugmachaga xususiyatdagi **glyph** qatori orqali odatda rasmlar joylashtirish mumkin;

 - **TTabSet** - ko`p varaqli oynalar tashkil etish uchun ishlatiladi. Oynani nomini xususiyatdagi (svoystva) *Tabs* qatori orqali yozish mumkin. Masalan: `TabSet1.Tabs:=Notebook1.Pages`

Bir varaqdan ikkinchi varaqqa o`tish uchun hodisadagi (sobitiya) *OnClick* qatoriga `TTabSet1.TabIndex` so`zlari yoziladi;

 - **TNoteBook** – ko`p varaqli muloqotli oynalarini hosil qilish uchun ishlatiladi. Har bir varaqqa o`ziga xos ob`ektni joylashtirish mumkin. Bu *TTabSet* bilan birgalikda ishlatiladi;

 - **TTabbedNoteBook** – ko`p varaqli muloqotli yig`ma oynalar hosil qilish uchun ishlatiladi. Bunda varaqlar nomi yuqorida yozilgan bo`ladi;

 - **TMaskEdit** - *TEdit* ob`ektining analogidir. Bunda kiritiladigan matnning (belgini) formatini ham ko`rsatish mumkin. Ma`lumotlarning formati xususiyatdagi *Edit Mask* qatori orqali aniqlanadi. Bunda sana, valyuta va boshqa formatlarni berish mumkin;

 - **TOutline** – bog`langan ma`lumotlarni ierarxik ko`rinishda tasvirlash uchun ishlatiladi. Masalan: kataloglar daraxti ko`rinishida;

 - **TStringGrid** - Matnli ma`lumotlarni jadval ko`rinishida tasvirlashni hosil qilish uchun foydalaniladi. Jadvaldagi har bir elementga o`tish, xususiyatdagi *Cell* qatori orqali amalga oshiriladi;

 - **TDrawGrid** - ixtiyoriy tipdagi ma`lumotlarni jadval ko`rinishida tasvirlashni hosil qilish uchun ishlatiladi. Jadvaldagi har bir elementga o`tish, xususiyatdagi *CellRect* qatori orqali amalga oshiriladi;

 - **TLevel** – relefli tasvir hosil qilish uchun ishlatiladi;

 - **THeader** – jadvallarning sarlavhasining o`lchami o`zgaruvchan tarzda tasvirlash uchun ishlatiladi;

 - **TScrollBar** – davomini ko`rish tasvirini hosil qilish, ya`ni yugurtirish yo`lakchalarini hosil qilish uchun ishlatiladi.

Dialogs komponentlar palitrasi

Bu palitrada Windowsning standart dialogli oynalarini tashkil qilish ob`ektlari joylashgandir. Bu ob`ektlar dasturda ko`rinmas xususiyatiga ega bo`lib, bu dialogli oynalarni chiqarish quyidagicha bo`lishi mumkin.



10-rasm.Dialogs komponentlar palitrasi

Bu ob`ektlar quyidagilardir:

 - **TOpenDialog** - faylni ochish dialogli oynasini hosil qilish uchun ishlatiladi;

 - **TSaveDialog** – faylni saqlash dialogli oynasini hosil qilish uchun ishlatiladi;

 - **TFontDialog** - shriftni tanlash va sozlash dialogli oynasini hosil qilish uchun ishlatiladi;

 - **TColorDialog** – rangni tanlash dialogli oynasini hosil qilish uchun ishlatiladi;



- **TPrintDialog** – bosmaga (pechat) chiqarish dialogli oynasini hosil qilish uchun ishlatiladi;



- **TPrinterSetupDialog** – printerni sozlash dialogli oynasini hosil qilish uchun ishlatiladi;



- **TFindDialog** – so`zni qidirish dialogli oynasini hosil qilish uchun ishlatiladi;



- **TReplaceDialog** – so`zni boshqa so`zga almashtirish dialogli oynasini hosil qilish uchun ishlatiladi.

Delphi 7.0 dasturlash tizimining grafik komponentlari

Canvas ob'ekti xususiyati

Delphi dasturlash tizimining vizual komponentlar kutubxonasi bir qancha grafiklar hosil qilish imkoniyatini beruvchi ob'ektlarga egadir. Bulardan ayrimlari: *TImage*, *TShape* va *TBevel* ob'ektlari.



- **TImage** – bu ob'ekt formaning ixtiyoriy joyiga grafik tasvirni joylashtirish imkonini beradi. Bunda .bmp, .ico, .wmf formatidagi rasmlar tasvirlanadi. Ob'ekt Additional komponentlar palitrasida joylashgan. Rasmlarni ob'ektlar inspektori xususiyatidagi Picture qatori orqali qo'yish mumkin. Agar bu rasmlar formaga joylashtirilgan bo'lsa, u holda .exe kengaytmali fayli katta hajmga ega bo'lishi mumkin. Rasmlarni dastur bajarilishi davomida ham chiqarish mumkin. Buning uchun quyidagi qatorlar yozilishi kerak, ya'ni: `If OpenFileDialog1. Execute then Image1. Picture.LoadFromFile (OpenDialog1. File Name);`

Bu ob'ektning ob'ektlar inspektori xususiyatidagi Center va Stretch qatorlari eng muhim ahamiyatga ega bo'lib, ikkala qator ham Boolean tipiga mansubdir. Agar Center qatorida true bo'lsa, u holda rasm formadagi TImage ob'ektining o'rtasida joylashadi. Agar Stretch qatorida true bo'lsa, u holda rasm formadagi TImage ob'ekti o'lchamini butunligicha egallab turadi.



- **TShape** - oddiy grafik tasvirlarni formada hosil qilish uchun ishlatiladi, ya'ni doira, kvadrat, to'rtburchak va shunga o'xshash geometrik shakllar. Ob'ekt xususiyatidagi Pen qatori orqali chizma chiziqlarining rangi va ko'rinishini ko'rsatish mumkin. Brush qatori orqali chizma ichining rangi ko'rsatiladi. Xususiyatdagi bu amallarni dasturni bajarish davomida ham amalga oshirish mumkin.

Canvas xususiyati. Ko'pgina grafik ob'ektlar o'ziga xos Canvas xususiyatiga ega. Canvas xususiyatida Pen (qalam), Brush (buyovchi qalam), Font (shrift) lar mavjud. Shuningdek, Draw, TextOut, Arc, Rectangle va boshqa operatorlarga ega. Endi Canvas xususiyatining ayrimlarini ko'rib chiqaylik:

Brush – bo'yovchi qalam. Bu ham o'zining quyidagi xususiyatlariga ega:

Bitmap - ekranni 8*8 o'lchamidagi qismi rangini o'zgartiruvchi;

Color – ichki soha;

Style – ichki sohani bo'yash uslubi;

Pen – qalam, chiziqning tashqi ko'rinishini tanlash uchun;

Color – chiziq rangi;

Style - chiziqning ko`rinishi, ya'ni chiziq, punktir, shtrix va boshqalar;

Width - chiziqning qalinligi;

PenPos – qalamning o`rni (pozitsiyasi). Qalamning o`rnini *MoveTo* buyrug`i orqali ko`chirish mumkin;

Pixels - tasvirlash elementlaridan tashkil topgan ikki o`lchamli massivdir.

Endi Canvas usullarini ko`rib chiqamiz:

Oddiy chizmalarni chizish uchun: Arc, Chord, LineTo, Pie, Polygon, Polyline, Rectangle, RoundRect buyruqlari ishlatiladi.

Canvas xususiyatida rasmlarni aks ettirish uchun: Draw va StretchDraw buyruqlaridan foydalaniladi. Canvas xususiyatida matnlarni aks ettirish uchun: TextOut va TextRect buyruqlaridan foydalaniladi. TextRect buyrug`i orqali ko`rsatilgan to`rtburchak ichida matnni chiqarish mumkin. Matnning uzunligi va balandligi TextWidth va TextHeight funktsiyalari orqali ko`rsatiladi.

1.3.Dasturlash tili elementlari

Hozirgi kunda juda ko'p algoritmik tillar mavjud. Bu tillar ichida Pascal tili universal tillardan biri bo'lib, boshqa tillarga qaraganda imkoniyatlari kengroq tildir. So'ngi yillarda Pascal tili juda takomillashib, tobora ommaviylashib bormoqda.

Delphi dasturlash vositasi Turbo Pascal tilining rivoji bo'lgan Object Pascal tilini ishlatadi. Hozirgi kunda bu tilga juda ko'plab yangiliklar kiritilgan, uning imkoniyatlari yanada kengaytirilgan, shu sabab bu tilni Delphi dasturlash tili deb ham atash mumkin.

Delphi dasturlash tili ham boshqa dasturlash tillari kabi o'z alfavitiga va belgilariga ega. U 26 bosh lotin harflarini, 0 dan 9 gacha bo'lgan arab raqamlarini va quyidagi belgilarni ishlatadi: bo'shliq belgisi; 4 ta arifmetik amallar + , - , * , / ; mantiqiy amallarni bajarish uchun <, >, <=, >= , <>, = belgilarini ishlatadi. Bulardan tashqari vergul, nuqta, ikki nuqta, kichik qavs, katta va o'rta qavslar. Dasturda izohlar istalgan joyda berilishi mumkin. Ular figurali qavs ichida yoziladi.

Masalan. Program ad; { Bu dastur nomi }

O'zgaruvchilarni tavsiflash

Ma'lumki, har qanday qiymat yoki belgi bilan ish ko'rish uchun eng avvalo ularga xotirada joy ajratish zarur bo'ladi. Buning uchun ishlatilishi zarur bo'lgan o'zgaruvchi yoki o'zgarmaslar Pascal tilida e'lon qilinishi kerak. Ko'pchilik hollarda dasturlarda o'zgarmas qiymatlar bilan ish ko'rishga to'g'ri keladi. Masalan, $n=20$, $e=2.71$ kabi sonlar Pascal va Delphi tilida quyidagicha e'lon qilinadi.

Const pi=3.14; n=20; e=2.71;

Umumiy holda o'zgaruvchilar var (variable) xizmatchi so'zi orqali qabul qilishi mumkin bo'lgan qiymatiga qarab turlarga bo'linadi. Butun sonlar ishlatilishi chegarasiga qarab turlicha e'lon qilinishi mumkin:

- byte 0 dan 255 gacha bo'lgan diapozondagi butun sonlar ;
- shortint -128 dan 127 gacha bo'lgan diapozondagi butun sonlar;
- word 0 dan 65535 gacha bo'lgan diapozondagi butun sonlar;
- integer -32768 dan 32767 gacha bo'lgan diapozondagi butun sonlar;
- longint -2147483648 dan 2147483648 gacha bo'lgan diapozondagi butun sonlarni qabul qiladi.

Var I,j : integer;

Haqiqiy sonlar uchun qo'yilgan masala yechimning aniqlik darajasiga qarab quyidagi operatorlar yordamida identifikatorlar e'lon qilinadi:

1-jadval

Identifikator turi	Qiymatlar oralig'i	Aniqlik darajasi	Egallagan hajmi
Real	$2.9e^{-39} .. 1.7e^{38}$	11 – 12	6 bayt
Single	$1.5e^{-45} .. 3.4e^{38}$	7 – 8	4 bayt
Double	$5.0e^{-324} .. 1.7e^{308}$	15 – 16	8 bayt
Extended	$3.4e^{-4932} .. 1.1e^{4932}$	19 – 20	10 bayt

var x,y:real;

Mantiqiy o'zgaruvchilar esa Boolean xizmatchi so'z yordamida e'lon qilinadi. Bu tipdagi o'zgaruvchi faqat rost – true yoki yolg'on – false qiymatlarini qabul qiladi.

Belgilar Char xizmatchi so'z yordamida uzunlikka ega bo'lgan satrlar esa string operatori yordamidja aniqlanadi.

Var s: char;

d: string;

begin

end.

Pascal va Delphi algoritmik tilida tuzilayotgan har qanday dastur ma'lum bir strukturaga ega bo'ladi va bu tildagi eng kichik dastur quyidagi ko'rinishda bo'ladi.

Begin

(Dasturning asosiy tanasi)

end.

Kompyuterda biror masalani yechish uchun boshqa dasturlarga, tashqi qurilmalarga murojaat qilish mumkin, o'zgarmas yoki yangi o'zgaruvchilarning ko'rinishini e'lon qilish mumkin va hakoza.

Pascal va Delphi tilidagi dasturning umumiy strukturasi quyidagi ko'rinishda bo'ladi:

Program- Programma nomini berish;

Uses - Ishlatilayotgan kutubxona bo'limlari (modullari);

Label - Dasturning asosiy qismida ishlatilayotgan belgi (metka) larni e'lon qilish;

Const - O'zgarmaslarni e'lon qilish;

Type - Yangi o'zgaruvchilarning turini muomalaga kiritish;

Var - Asosiy dasturda muomalada bo'ladigan o'zgaruvchilarni e'lon qilish;

Protseduralarni e'lon qilish;

Begin

Dasturning asosiy qismi;

End.

Demak, har qanday dastur yuqorida berilgan asosiy tuzilmaning xususiy holi bo'lishi mumkin va ular o'z navbatida Delphi tiliga xos bo'lgan asosiy tushunchalar asosida hosil qilinadi.

Standart funksiyalar

2-jadval

t/r	Matematik ko'rinishi	Pascal tilida yozilishi
1	$ x $	abs(x)
2	$\sqrt{x}$	sqrt(x)
3	x^2	sqr(x)
4	$\sin x$	sin(x)
5	$\cos x$	cos(x)
6	$\ln x$	ln(x)

7	e^x	exp(x)
8	a^x	exp(x*ln(a))
9	x ni yaxlitlash	Round(x)
10	x ning butun qismi	Trunk(x)

Simvollar uchun ishlatiladigan funksiyalar:

Ord(x) – x belgining tartib nomerini aniqlaydi;

Pred(x) – x dan oldingi belgining tartib nomerini aniqlaydi;

Succ(x) – x dan keyingi belgining tartib nomerini aniqlaydi;

Keltirilgan funksiyalar Pascal tilida maxsus funksiyalar deb ataladi.

Tilining bir necha kalit soʻzlari:

And-va;

Array –massiv;

Begin-boshlash;

Case-variant;

Const-oʻzgarmas;

Div –qoldiqsiz;

Do –bajarish;

Downto –gacha kamaytirish;

Else –aks holda;

End –tamom;

File –fayl;

For –uchun;

Fuktion –funksiya;

Goto –ga oʻtish;

If –agar;

In –ga tegishli;

Label-belgi;

Mod-butun qoldikli boʻlishdagi qoldiq.

Oʻzgartirish funksiyalari:

Chr(n) n kodli simvol;

IntToStr(k) Butun k ni tasvirlovchi satr;

FloatToStr (n) Haqiqiy n tasvirlovi satr;

FloatToStrF(n, f , k,m) Haqiqiy n tasvirlovi satr. Bunda: f – format; k – aniqlik; m – kasr qismidagi raqamlar soni;

StrToInt (s) Satrni butun songa oʻtkazish ;

StrToFloat (s) Satrni haqiqiy songa oʻtkazish;

Round (n) Haqiqiy sonni yaxlitlash;

Trunc (n) Haqiqiy son kasr qismini olib tashlash;

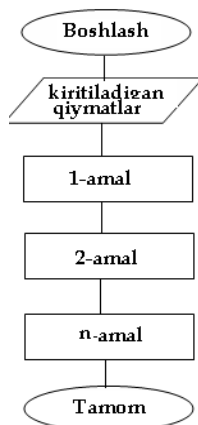
Frac(n) Kasrli sonning kasr qismi;

Int (n) Sonning butun qismi.

II-BOB. DELPHI DASTURLASH TILIDA MATEMATIK MASALALARNI YECHISH

2.1. Chiziqli masalalarga oid dasturlar tuzish

Chiziqli tuzilishga ega bo'lgan algoritmlarda ko'rsatmalar yozilish tartibida bajariladi. Ularning blok – sxemasi ishga tushirish, to'xtatish, kiritish-chiqarish jarayoni bloki hamda avvaldan ma'lum jarayon bloklari yordamida tuzilib, bir chiziq bo'ylab ketma-ket joylashgan bo'ladi. Ya'ni hech qanday shart talab qilmaydigan algoritmgacha chiziqli algoritm deb qarladi. Uni hisoblash dasturining blok sxemasi quydagicha bo'ladi.



1-chizma. Chiziqli algoritmlashning blok sxemasi

Chiziqli tuzilishdagi algoritmni tuzish masalani yechish uchun kerak bo'ladigan boshlang'ich ma'lumotlarni tashkil qiluvchi o'zgaruvchilar nomi, ularning turi va o'zgarish ko'lamini aniqlashdan boshlanadi. Boshlang'ich ma'lumotlar aniqlangandan so'ng, oraliq va yakuniy natijalar, o'zgaruvchilarining nomlari, turlari va mumkin bo'lsa, o'zgarish ko'lamini aniqlash kerak. Yuqoridagi aytilganlarni tushinish uchun quyidagi misollarni keltiramiz:

1 – m i s o l. Ikki sonning yig'indisini hisoblash dasturini tuzamiz.

$$S=a+b; (1)$$

Ikki sonni yig'indisini hisoblash dasturini avval Turbo Pascal tilida, so'ngra esa Delphi dasturlash tilida tuzamiz. Bu bilan Delphi va Pascal dasturlash tillari orasidagi farqlarni ham ko'rib chiqamiz. Berilgan masalani dastlab Pascal dasturlash tilida tuzamiz.

Program ul;

var

a,b,s:real;

begin

write('a='); read(a);

write('b='); read(b);

s:=a+b;

write('s=',s);

end.

Dastur kodini kiritib [Ctrl+F9] klavishini bosish orqali dastur ishga tushadi. Natijada “a= ”, “b=” kabi yozuv paydo bo`ladi, siz ixtiyoriy 2 ta sonni kiritasiz va [Alt+F5] tugmalar kombinatsiyasidan foydalanib natija olasiz. Bu ishlarni Pascal dasturlash muhitida bajarish mumkin.

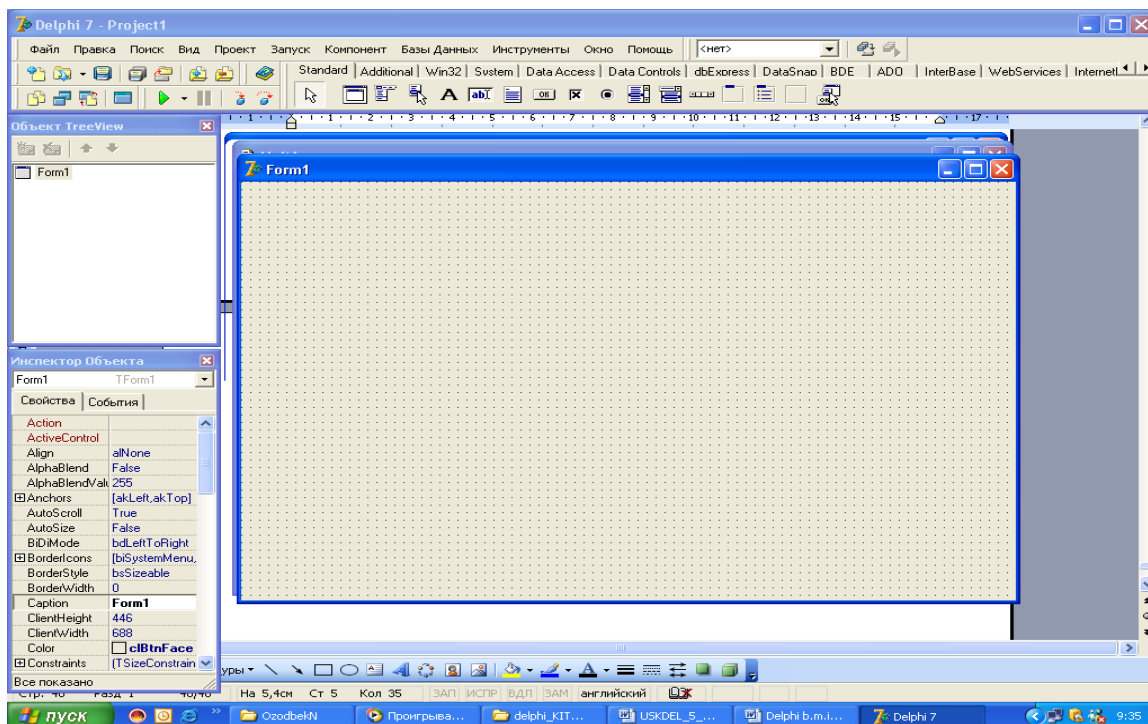
Yuqorida bayon etilgan misolni Delphi 7.0 dasturlash tili muhitida dasturini tuzamiz. Dastur tuzish Pascal tiliga asoslanib Delphi dasturlash tilida dastur kodini kiritish kerak. Pascal tili bilan Delphi dasturlash tillari orasida ishlatiladigan umumiy operatorlar va belgilar to`plamining ba`zilarini ko`rib chiqamiz: o`zgarmaslar **Const**, o`zgaruvchilar **Var**, qiymatlar tipi **Real**, **Integer**, **Char**, **String**, **Boolean**, matnda ishlatiladigan turli xil belgilar, arifmetik amallar, standart funksiyalar, shartli operatorlar va boshqalar.

Pascal tilida “**Read**” kiritish operatoridan foydalansak, Delphi dasturida esa “**strtofloat**” kiritish operatoridan foydalanamiz. Yoki chiqarish operatori “**Write**” o`rniga “**showmessage, labeln.caption**”. kabi bog`lanishlar mavjud.

Yuqoridagi misolni Delphi dasturlash tilida tuzadigan bo`lsak. Delphi dasturini yuklab yaratiladigan dasturni loyahasini tuzib olish kerak, aniqrog`i formaga kerakli tugmalar, menyular, yozuvlar va shunga o`xshash komponentlarni joylashtirib olgach, shu yaratgan komponentlarni har biriga tegishli buyruqlarni kiritish zarur.

Keltirilgan masalani Delphi dasturlash tilida dasturini tuzishimiz uchun, Delphi dasturlash tilida quyidagi bosqichlarni amalga oshiramiz:

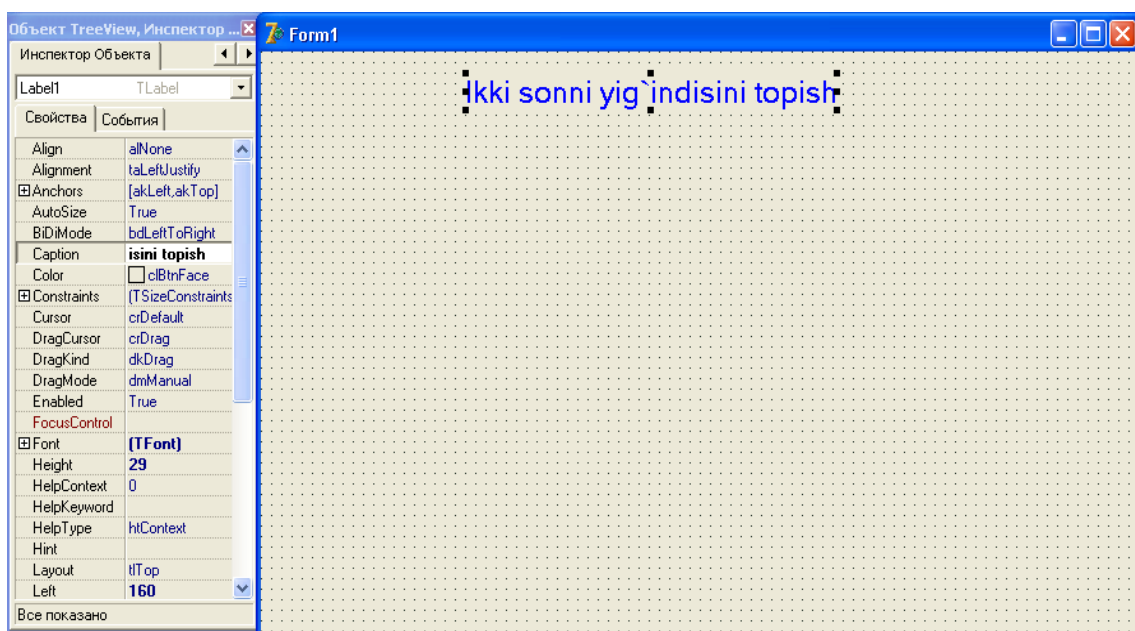
1. Dasturni ishga tushiramiz



11-rasm. Delphi dasturlash tilini dasturlash oynasi

2. Hosil bo`lgan oynada matnni ekranga joylash uchun Delphi dasturlash tilining Standard komponentalar palitrasidan (uskunalar panelidan) “A” piktogrammasi belgilanib forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda matn

joylashtirilishi lozim boʻlgan joy ajratiladi. Natijada **Label1** matn maydoni hosil qilinadi hamda parametrlar va xossalar boʻlimidan foydalanib kerakli matn teriladi.



12-rasm. Dasturlash oynasida matn kiritish qatori

3. Matnga ishlov berish uchun (masalan, kattalashtirish yoki kichiklashtirish; kursiv yoki qalin qilish va boshqa) ya'ni unga o'zgartirish kiritish uchun kerakli xossa tanlanib ular o'zgartiriladi. Masalan, kiritilgan matnni kattalashtirish yoki kichiklashtirish uchun oldin matn maydoni ajratilib, keyin Font xossasiga kiriladi va muloqot darchasidan shrift, uning o'lchami va rangi belgilanib "Ok" tugmasi tanlanadi.

4. **Label komponentasi.** Label komponentasi nafaqat ma'lumotlarni ekranga joylashtirish uchun xizmat qiladi, balki dastur natijalarini chiqarishda ham ishlatish mumkin. Buning uchun dasturda `Label.n.caption='Dastur natijasi';` buyrug'i berilishi kerak. Misol, `label5.Caption:='s='+floattostr(s);` bu yerda `s:String` – matn o'zgaruvchisi.

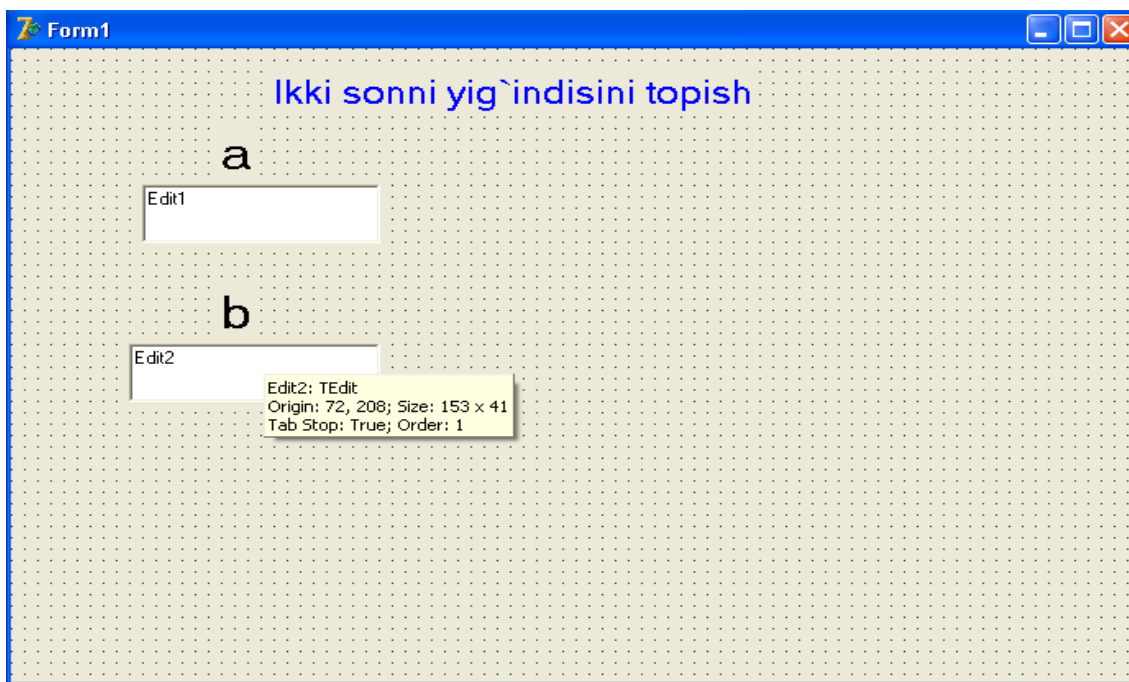
5. **Edit kiritish qatori.** Edit kiritish qatori matnni bir qatordan kiritish va uni tahrirlash uchun ishlatiladi. Matn kiritish qatorini ekranga joylash uchun Delphi ning Standard palitrasidan “**ab**” piktogrammasi belgilanib forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda matn kiritilishi lozim boʻlgan joy ajratiladi. Natijada **Edit1** matn kiritish maydoni hosil qilinadi. Matnni kiritish dastur ishchi holatiga oʻtilganda bajariladi.



13-rasm. Dasturlash oynasida “Edit” kiritish qatori

Matn qatoriga kiritilgan ma’lumot faqat matn, ya’ni String (qator) bo’lib hisoblanadi. **Edit** kiritish qatorida kiritilgan ma’lumotni dasturda o’qib va uni raqamga o’tkazish uchun ko’p hollarda Val funksiyasidan foydalaniladi. Bu funksiya Turbo Pascal da quyidagicha yoziladi. **Val(Edit1.Text,a,cod)** – bu yerda **a: Real;** - o’zgaruchisi bo’lib, **Edit1.Text** maydonidagi ma’lumotni raqam qilib o’zlashtiradi. Cod: Integer; deb elon qilinadi.

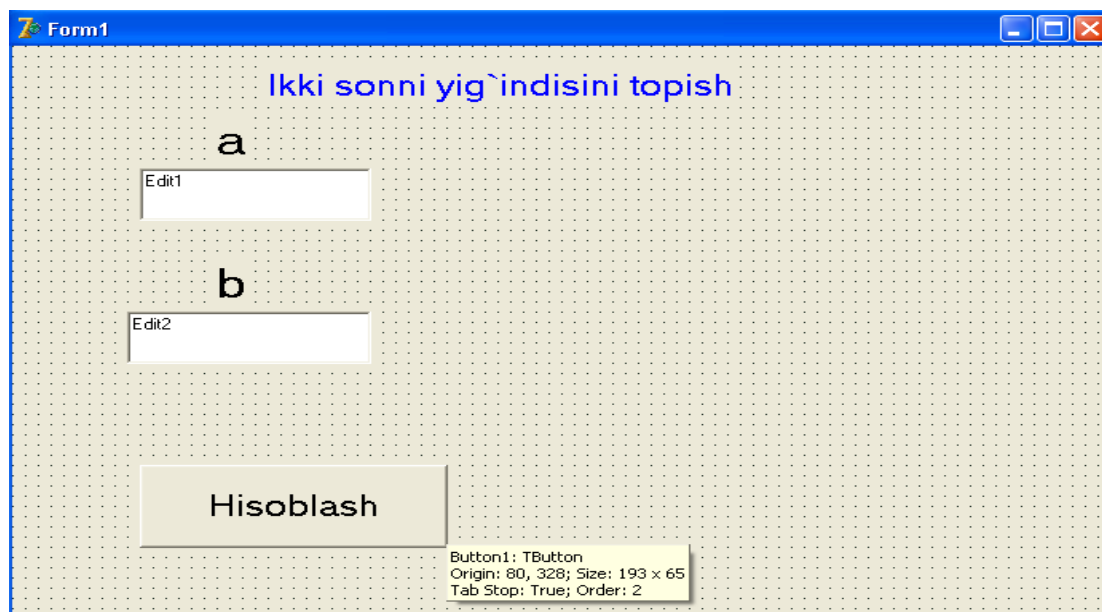
Edit2 matn kiritish maydoni ham xuddi shu tartibda bajariladi.



14-rasm. Dasturlash oynasida “Edit” kiritish qatori

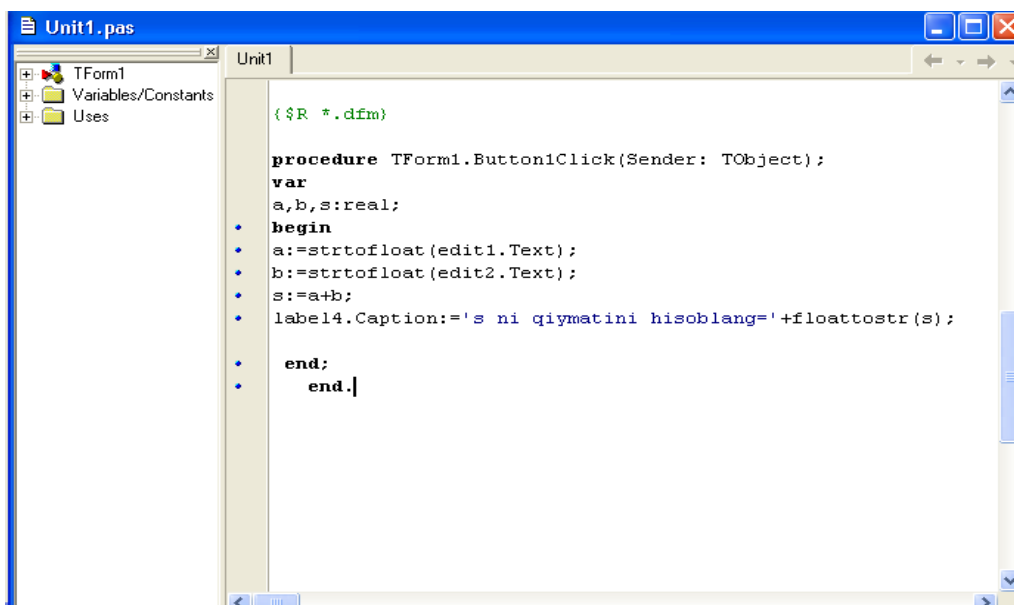
5. **Button tugmachasi.** Bu tugmacha bosilishi natijasida kutilishi lozim bo'lgan jarayonlar (masalan, hisoblashlar yoki bajarilishi lozim bo'lgan operatsiyalar) bajarilishga xizmat qiladi.

Button tugmachasini ekranga joylash uchun Delphi ning Standard palitrasidan “Ok” piktogrammasi belgilanib forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda tugmacha qo'yilishi lozim bo'lgan joy ajratiladi. Natijada “**Button1**” tugmachasi hosil qilinadi. Tugmacha nomi **Caption** xossasidan foydalanib o'zgartiriladi.



15-rasm. Dasturlash oynasida “Button” tugmasinidan foydalanish

6. Dasturdagi hisoblash jarayonlarini tashkil qilish uchun “**Button1**” (**hisoblash**) hosil qilingan tugmachani ikki marta tez-tez bosish bilan Unit1.Pas dastur kodlarini tahrirlash darchasiga o'tilib, u yerdan modul ichiga kerakli operatorlarni yozish bilan amalga oshiriladi.



16-rasm. Dasturkitirish maydoni

Biz yuqorida ikki sonni yig`indisini Turbo Pascal tilida quyidagicha tuzgan edik :

Program ul;

var

a,b,s:real;

begin

write('a='); read(a);

write('b='); read(b);

s:=a+b;

write('s=',s);

end.

Delphi dasturlash tilida esa quyidagicha bo`ladi:

var

a,b,s:real;

begin

a:=strtofloat(edit1.Text);

b:=strtofloat(edit2.Text);

s:=a+b;

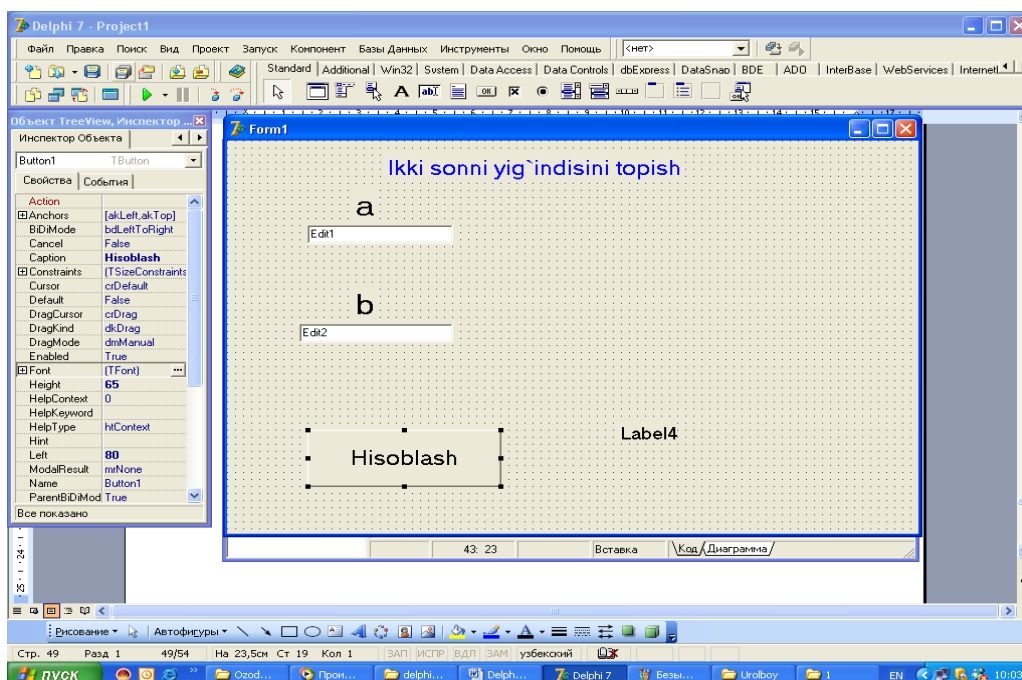
label4.Caption:='s='+floattostr(s);

end;

end.

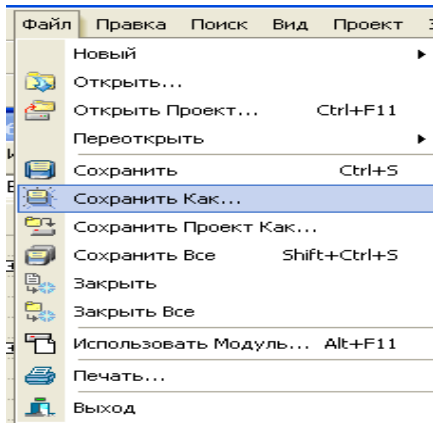
Turbo Pascal tilida o`zgaruvchilarni kiritish Write('a='); read(a); ko`rinishda bo`lsa, Delphi dasturida esa a:=strtofloat(edit1.Text); ko`rinishda bo`ladi. Yoki ma'lumotni chop etish Turbo Pascal tilida Write('s=',s); buyru'i berilsa, Delphi dasturida esa quyidagicha bo`ladi label4.Caption:='s='+floattostr(s);

7. Dasturlash maydoniga, dastur kodini kiritib bo`lgandan so`ng, F12 tugmasini tanlaymiz natijada quyidagi ko`rinishga keladi.

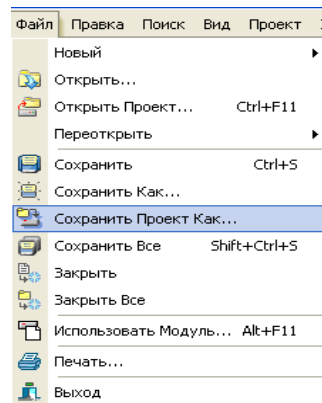


17-rasm. Dasturlash oynasining ko`rinishi

8. Hosil bo'lgan dasturni ishga tushirish uchun:
birinchi marta



ikkinchi marta



18-rasm. Tuzilgan dasturni kompyuter xotirasiga saqlash na'munasi

9. Buyruqlari orqali kompyuter xotirasiga saqlaymiz. Yuqoridagi ketma-ketlikdan so'ng, dasturni kompilatsiya qilishimiz kerak. Kompilatsiya ishini asboblar panelidagi quyidagi



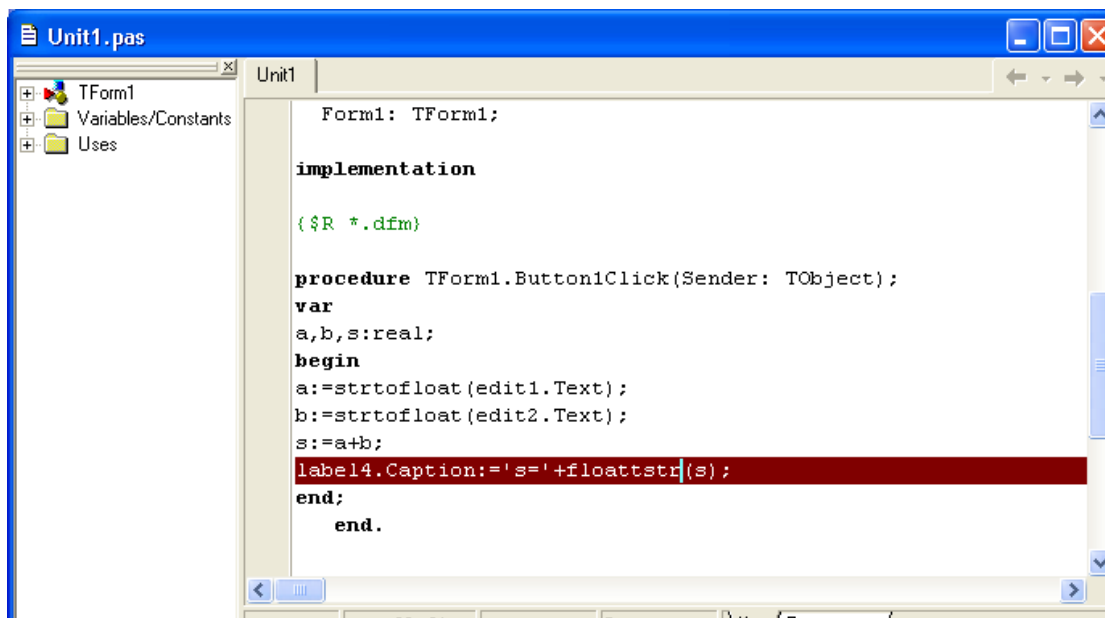
tugmacah yordamida amalga oshiramuz.

Kompilatsiyalanganidan so'ng, oynaning ko'rinish holati quyidagicha bo'ladi.

19-rasm. Ikki sonni hisoblash dastur oynasi

Izoh : Har qanday ikki sonni yig'indisini hisoblash uchun, hosil qilingan dastur maydoniga yangi qiymat kiritish orqali natijani olishimiz mumkin.

Agar dasturning biror joyida sintaksis xatolik bo'lsa, quyidagi ko'rinishga o'xshash xatolik ko'rsatiladi.



20-rasm. Dasturda xatoliklar na'munasi

Tashkil qilingan modulning to'liq ko'rinishini keltiramiz.

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Edit1: TEdit;
    Edit2: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Button1: TButton;
  procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
```

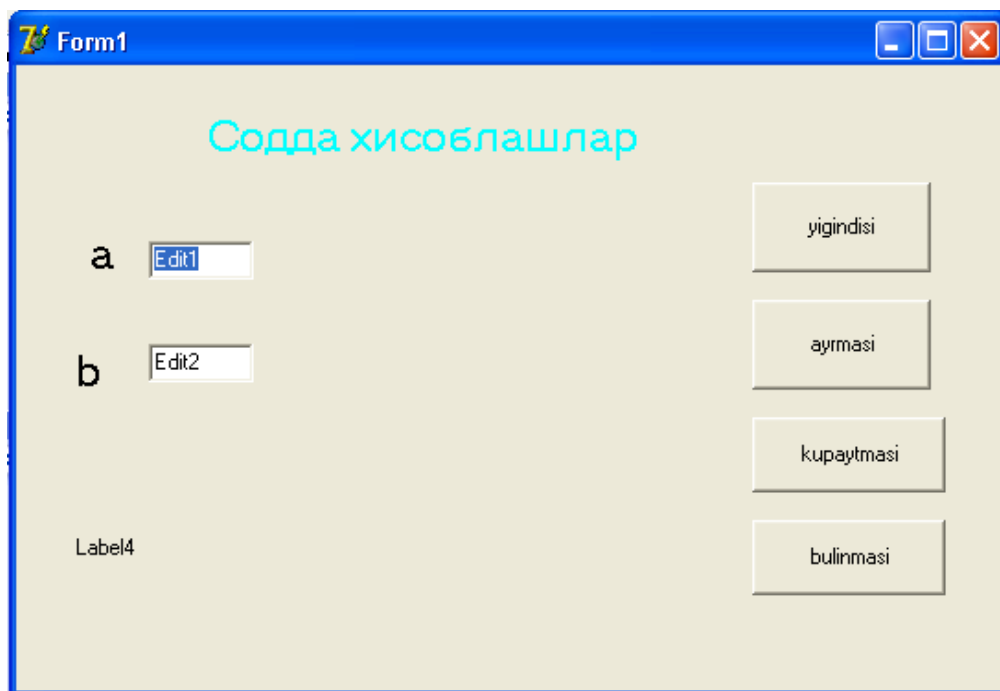
```

end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var
  a,b,s:real;
begin
  a:=strtofloat(edit1.Text);
  b:=strtofloat(edit2.Text);
  s:=a+b;
  label4.Caption:='s='+floattostr(s);
end;
end.

```

Yuqorida keltirilgan dasturdan foydalanib istalgan ikki sonni yig`indisini hisoblashimiz mumkin.

2. - m i s o l. Bitta oynada ikki sonning yig`indisini, ayirmasini, ko`paytmasini va bo`linmasini hisoblash dasturini tuzing.



21-rasm. Arifmetik amallarni bajarish oynasi

Izoh : Har qanday ikki sonni yig`indisini, ayirmasini, ko`paytmasini va bo`linmasini hisoblash uchun, hosil qilingan dastur maydoniga yangi qiymat kiritish orqali natijani olishimiz mumkin.

Formani loyihalashtirish oynasini yuqoridagi masaladan foydalangan holda, **Label1, Label2, Label3, Label4, Edit1, Edit2 , Button1, Button2, Button13, Button4** tugmalarni hosil qilamiz, parametrlar va hodisalar bo`limidan tugmalarga kerakli ma'lumotlarni kiritamiz (masalan. Button1 tugmasiga Yig`indi. v..x.k.)..**“Yig`indisi ”** belgisi yozilgan tugmaning dasturlash maydoniga quyidagi dastur kodini kiritamiz:

```
var  
a,b,s:real;  
begin  
a:=strtofloat(edit1.Text);  
b:=strtofloat(edit2.Text);  
s:=a+b;  
label4.Caption:='s'+floattostr(s);  
end;
```

“Ayirmasi” belgisi yozilgan tugmaning dasturlash maydoniga quyidagi dastur kodini kiritamiz:

```
var  
a,b,s:real;  
begin  
a:=strtofloat(edit1.Text);  
b:=strtofloat(edit2.Text);  
s:=a-b;  
label4.Caption:='s'+floattostr(s);  
end;
```

“Ko`paytmasi” belgisi yozilgan tugmaning dasturlash maydoniga quyidagi dastur kodini:

```
var  
a,b,s:real;  
begin  
a:=strtofloat(edit1.Text);  
b:=strtofloat(edit2.Text);  
s:=a*b;  
label4.Caption:='s'+floattostr(s);
```

“Bo`linmasi” belgisi yozilgan tugmaning dasturlash maydoniga quyidagi dastur kodini:

```
var  
a,b,s:real;  
begin  
a:=strtofloat(edit1.Text);  
b:=strtofloat(edit2.Text);  
s:=a/b;  
label4.Caption:='s ni qiymatini hisoblang'+floattostr(s);  
end;  
end.  
kiritamiz.
```

Yuqorida tuzilgan dasturlar modulining to`liq kodini keltiramiz.

```
unit Unit1;
interface
uses
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls;
type
    Tform1= class(TForm)
        Label1: TLabel;
        Label2: TLabel;
        Label3: TLabel;
        Edit1: Tedit;
        Edit2: Tedit;
        Button1: Tbutton;
        Button2: Tbutton;
        Button3: Tbutton;
        Label4: TLabel;
        procedure Button1Click(Sender: TObject);
        procedure Button2Click(Sender: TObject);
        procedure Button3Click(Sender: TObject);
    private
    {Private declarations}
    public
    { Public declarations }
    end;
var
    Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var
a,b,s:real;
begin
a:=strtofloat(edit1.Text);
b:=strtofloat(edit2.Text);
s:=a+b;
label4.Caption:='s='+floattostr(s);
end;
procedure TForm1.Button2Click(Sender: TObject);
var
a,b,s:real;
begin
a:=strtofloat(edit1.Text);
b:=strtofloat(edit2.Text);
s:=a-b;
```

```

label4.Caption:='s'+floattostr(s);
end;
procedure TForm1.Button3Click(Sender: TObject);
var
a,b,s:real;
begin
a:=strtofloat(edit1.Text);
b:=strtofloat(edit2.Text);
s:=a*b;
label4.Caption:='s'+floattostr(s);
end;
procedure TForm1.Button4Click(Sender: TObject);
var
a,b,s:real;
begin
a:=strtofloat(edit1.Text);
b:=strtofloat(edit2.Text);
s:=a/b;
label4.Caption:='s'+floattostr(s);
end;
end.

```

3 - m i s o l. Bitta oynada konusning yon sirti, to`la sirti, va hajmini hisoblash dasturini hosil qiling:

1. $S_{yon} = \pi Rl$;
2. $S_{to`la} = S_{asos} + S_{yon} = \pi R(R + l)$;
3. $V = \frac{1}{3} \pi R^2 h$.

Dastur kodi va oynani umumiy ko`rinishini keltiramiz:

```

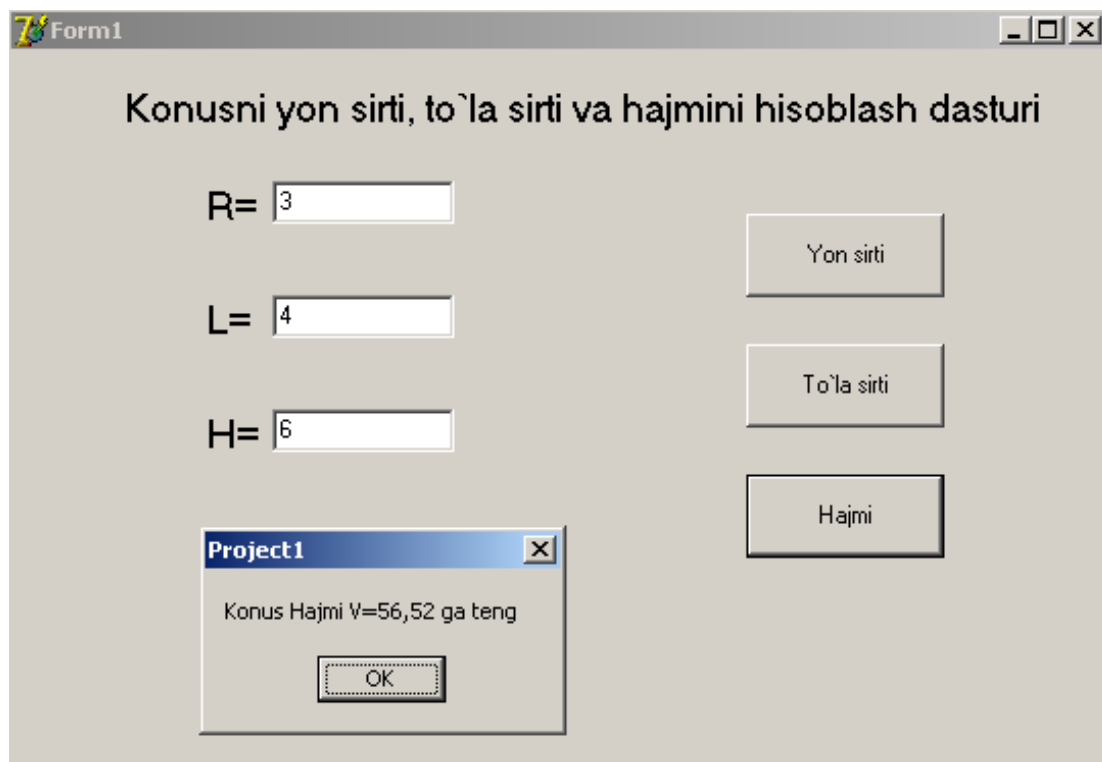
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Label3: TLabel;

```

```

Label4: TLabel;
Button1: TButton;
Button2: TButton;
Button3: TButton;
procedure Button1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
const pi=3.14;
var S,R,L:real;
begin
  R:=strtofloat(edit1.text);
  L:=strtofloat(edit2.text);
  S:=R*pi*L;
  showmessage('Konus yon sirti S='+floattostr(S)+' ga teng`');
end;
procedure TForm1.Button2Click(Sender: TObject);
const pi=3.14;
var S,R,L:real;
begin
  R:=strtofloat(edit1.text);
  L:=strtofloat(edit2.text);
  S:=R*pi*(R+L);
  showmessage('Konus to`la sirti S='+floattostr(S)+' ga teng`');
end;
procedure TForm1.Button3Click(Sender: TObject);
const pi=3.14;
var H,V,R,L:real;
begin
  R:=strtofloat(edit1.text);
  L:=strtofloat(edit2.text);
  H:=strtofloat(edit3.text);
  V:=(1/3)*R*R*pi*H;
  showmessage('Konus Hajmi V='+floattostr(V)+' ga teng`');
end;
end.

```



22-rasm. Bitta oynada konusning yon sirti, to`la sirti, va hajmini hisoblash dasturi

Izoh : Konusning yon sirti, to`la sirti, va hajmini hisoblash uchun, hosil qilingan dastur maydoniga yangi qiymat kiritish orqali natijani olishimiz mumkin.

Mustaqil bajarish uchun mashqlar

1. Koordinatalar tekisligida ikkita nuqta orasidagi masofani aniqlovchi dastur tuzing
2. To`rtta son berilgan. Shu sonlarning o`rta arifmetigi va o`rta geometr hisoblovchi dastur tuzing.
3. Quyidagi formula bilan berilgan uchburchakka tashqi chizilgan aylananing radiusini hisoblash dastur tuzing:

$$R = \frac{a * b * c}{4 * \sqrt{p * (p - a) * (p - b) * (p - c)}}$$

bunda p – uchburchakning yarim perimetri.

4. Uchburchakning a,b,c tomonlari berilgan. Shu uchburchakka ichki chizilgan aylanani radiusini dastur tuzing:

$$r = \frac{\sqrt{(p - a) * (p - b) * (p - c)}}{p}$$

bunda p – uchburchakning yarim perimetri.

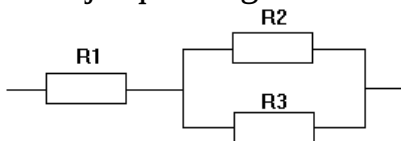
5. Uchburchakning uchta burchagi va balandligi orqali uning yuzini hisoblash dastur tuzing:

$$S = \frac{h^2 * \sin(A)}{2 * \sin(B) * \sin(C)}$$

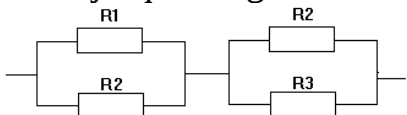
6. Uchburchakning berilgan uchta burchaklari va tomoni orqali uning yuzini hisoblash dastur tuzing:

$$S = \frac{a^2 * \sin(B) * \sin(C)}{2 * \sin(A)}$$

7. Zanjir qarshiligini hisoblash dastur tuzing:



8. Zanjir qarshiligini hisoblash dastur tuzing:



9. ABC uchburchakning ikkita tomoni va ular orasidagi burchagi berilgan. Shu uchburchakning qolgan uchinchi tomoni va ikkita burchagini aniqlash dastur tuzing

$$C^2 = A^2 + B^2 - 2 * A * B * \cos \alpha$$

10. Savdo do`konida kostyum uchun gazlama sotilmoqda. Uning kvadrat metri so`mda baholangan. X metr uzunlikka va Y metr enga ega bo`lgan gazlama narxini hisoblang.

11. Rombning tomoni va o`tkir burchagi berilgan uning yuzasini hisoblang:

$$S = a^2 \sin \alpha, \text{ bunda } a - \text{rombning tomoni, } \alpha - \text{esa o`tkir burchagi}$$

12. Arifmetik progressiyaning birinchi hadi va ayirmasi berilgan. Uning 30-chi hadi va dastlabki 40 ta hadining yig`indisini hisoblash dasturini tuzing.

13. Kesmaning koordinatalari berilgan. Shu kesmani o`rtasining koordinatalarini hisoblash dasturini tuzing.

14. 1 dan **m** gacha natural sonlar berilgan. Shu sonlarning arifmetik progressiya yordamida yig`indisini hisoblash dasturini tuzing.

Quyidagimisollarni hisoblash dasturini tuzing:

$$15. \quad a = \ln(y^{-\sqrt{|x|}}) * (\sin(x) + e^{(x+y)})$$

$$16. \quad b = \sqrt{c (\sqrt{y} + x^2)} * (\cos(x) - |c - y|)$$

$$17. \quad c = \arctg(x) - \frac{3}{5} e^{xy} + 0.5 \frac{|x+y|}{(x+y)^b}$$

$$18. \quad d = \frac{e^{|x-y|} * tg(z)}{\arctg(y) + \sqrt{x}} + \ln(x)$$

$$19. \quad e = \frac{(\cos(x) - \sin(y))^3}{\sqrt{tg(z)}} + \ln^2(x * y * z)$$

$$20. \quad f = y^x + \sqrt{|x| + e^y} - \frac{z^3 * \sin^2(y)}{y + z^2 / (y - x)}$$

21. $g = \frac{1 + \cos(x + y)}{\left| e^x - 2y / (1 + x^2 * y^2) \right|} * x^3 + \arcsin(y)$
22. $h = 2 + \frac{x^2}{\sqrt{2}} + \frac{|y^3|}{\sqrt{2}} + \frac{z^4 (\ln(x) + 1) * \sqrt{2}}{\sqrt{3}}$
23. $j = ((1 + y) * \sqrt{\sin^2(z)} - \frac{|y - x|}{5})^3$
24. $k = \ln \left| (y - \sqrt{|x|}) * (x - \frac{y}{z + x^2 / 4}) \right|$
25. $i = 0.5 * x^5 + 3 * \cos(x + y) + e^{-0.1yz} - \sqrt{|x * y|}$
26. $m = \sqrt{\left| -3 * \lg(x) * \lg(x^4 + y) / e^{-x} + 1 \right|}$
27. $n = \sqrt{e^x + \lg(x) + 1} * (\lg(y) + \cos(x * y) + \sqrt[3]{x})$
28. $p = \frac{\lg(x) - e^{x+y}}{\sqrt{2 + y^2} + |x^3 - \ln(y)|}$

2.2. Tarmoqlanuvchi jarayonlarga oid dasturlar tuzish

Tarmoqlanuvchi jarayonlarni dasturini dasturlash Delphi va Pascal dasturlash tillarida bir xil bo'ladi. Dastlab masalani Pascal tilida ko'rib chiqishga to'g'ri keladi. Pascal tilida tarmoqlanish If operatori bilan amalga oshiriladi. If operatori- bu shartli operator hisoblanadi. Pascal tilida shart - bu mantiqiy turdagi ifoda bo'lib, u faqat «chin»(True) yoki «yolg'on »(False) qiymatni qabul qiladi, quyidagi mantiqiy belgilar ishlatiladi: katta ">",kichik "<", kichik yoki teng "<=", katta yoki teng ">=", teng emas "<>", teng "=". Quyidagilar munosabat amallari yoki taqqoslash amallari ham deyiladi. Quyidagi mantiqiy amallar ishlatiladi:

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

Bu mantiqiy amallarning bajarilish natijalari quyidagicha:

3-jadval

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: $(5 < 6) \text{ AND } (6 < 50)$ -mantiqiy ifoda rost (True),
 $(20 > 0) \text{ OR } (20 < 0.5)$ -mantiqiy ifoda rost (True),
 $(10 < 8) \text{ AND } (10 < 15)$ -mantiqiy ifoda yolg'on (False),
 $\text{NOT}(100 > 3)$ -mantiqiy ifoda yolg'on (False).

O'zlashtirish mantiqiy ifodalarni biror bir mantiqiy o'zgaruvchiga yuborish ham mumkin. Masalan: $F := (A < B) \text{ AND } (A < C)$;

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:

1. To'liq ko'rinish operatori:

```
If <shart> then Begin
    <shart rost bo'lganda bajariladigan operatorlar>
End
Else
    Begin
    <shart yolg'on bo'lganda bajariladigan operatorlar>
    End;
```

2. Qisqa ko`rinish operatori:

```
If <shart> then Begin
    <shart rost blganda 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;

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.

Yuqorida bayon qilingan fikirlarni tekshirib ko`rish uchun Delphi dasturlash tilida shartli operatorlaridan foydalangan holda bir nechta dasturlar tuzamiz.

4- m i s o l. Berilgan Ikki sonlarni bo`linmasini hisoblash dasturini tuzing

$$C = \left(\frac{a}{b}\right); (1)$$

Dastur kodi va oynani umumiy ko`rinishini keltiramiz:

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs;

type

TForm1 = class(TForm)

Label1: TLabel;

Label2: TLabel;

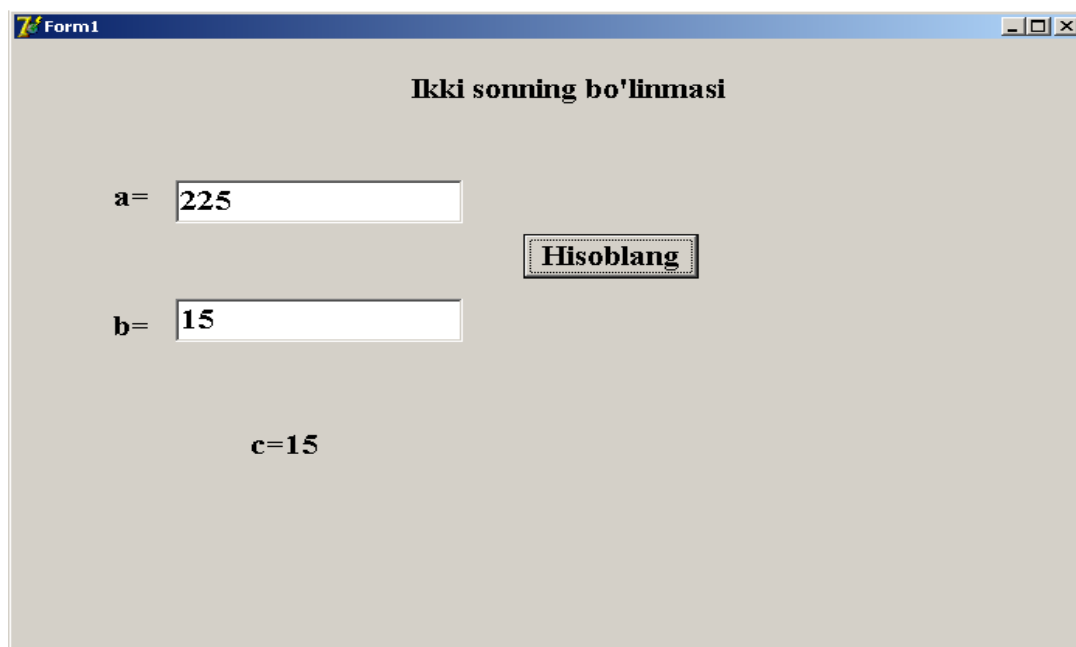
Label3: TLabel;

Label4: TLabel;

```

Edit1: TEdit;
Edit2: TEdit;
Button1: TButton;
procedure Button1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var a,b,c:real;
begin
a:=strtofloat(edit1.text);
b:=strtofloat(edit2.text);
if b=0 then label5.Caption:='Boluvchi nolga teng`
Else begin c:=a/b;
label5.Caption:='' + floattostr(c);
end;
end;

```



23-rasm. Ikki sonni bo'linmasini hisoblash dasturi

Izoh : Har qanday ikki sonni hisoblash uchun, hosil qilingan dastur maydoniga yangi qiymat kiritish orqali natijani olishimiz mumkin.

5- m i s o l. Geron formulasidan foydalanib uchburchakning yuzasini hisoblash dasturini tuzing.

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

1.-bosqich.

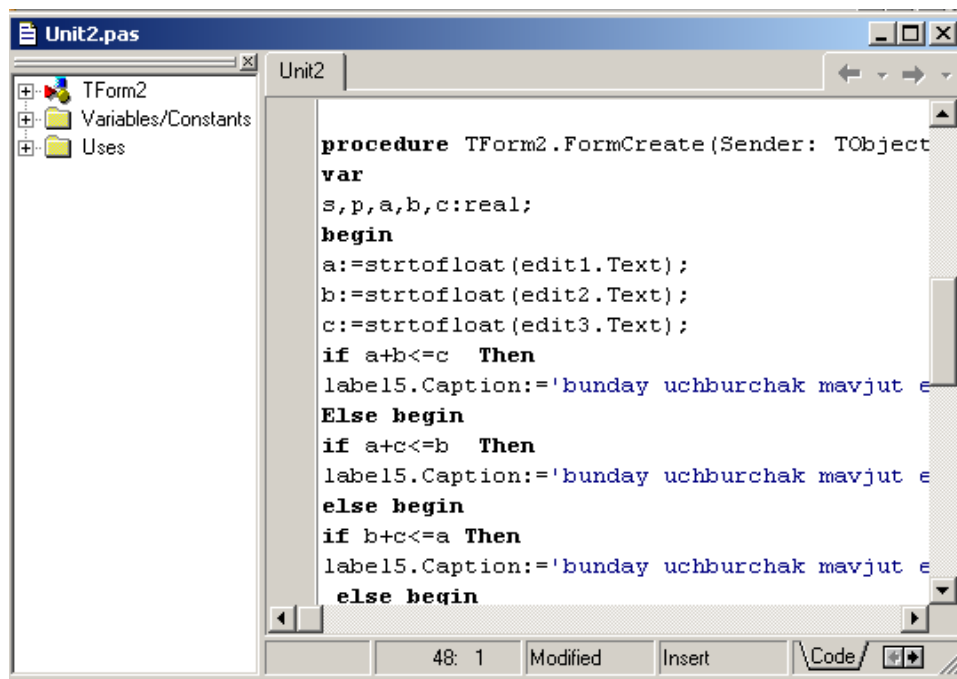
1. Dasturni ishga tushiramiz.
2. Standard jihozlar panelidan Label1, Label2, Label3, Label4, Label5 ni tanlaymiz.
2. Parametrlar va xosalar bo`limdan foydalanib Label1 ga “Uchburchakning yuzasini hisoblash”, Label2 ga “a=”, Label3 ga “b=”, Label4 ga “c=”, ni kiritamiz.
3. Standard jihozlar panelidan “Edit1”, “Edit2”, “Edit3” matn kiritish maydonlarini hosil qilamiz.
4. Standard jihozlar panelidan “Button1” tugmasi hosil qilamiz va parametrlar va xossalar bo`limdan foydalanib “Hisoblash” yozuvini hosil qilamiz.

2-bosqich.

1. “Button1” (Hisoblash) tugmasi yordamida dasturlash maydoniga o`tamiz.
2. Dasturlash maydoniga quyidagi dasturni kiriting:

```
var
s,p,a,b,c:real;
begin
a:=strtofloat(edit1.Text);
b:=strtofloat(edit2.Text);
c:=strtofloat(edit3.Text);
if a+b<=c Then
label5.Caption:='bunday uchburchak mavjut emas'+floattostr(s)
Else begin
if a+c<=b Then
label5.Caption:='bunday uchburchak mavjut emas'+floattostr(s)
else begin
if b+c<=a Then
label5.Caption:='bunday uchburchak mavjut emas'+floattostr(s)
else begin
p:=(a+b+c)/2;
s:=sqrt(p*(p-a)*(p-b)*(p-c));
label5.Caption:='S='+floattostr(s);
end;
end;
end;
end;
```

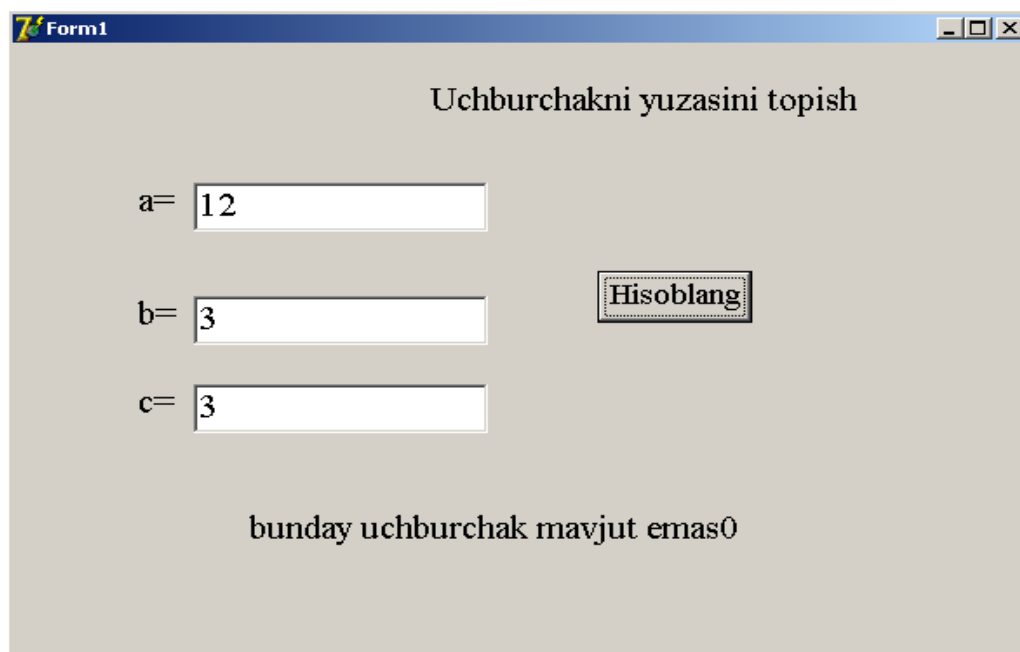
Dasturlash maydonining ko`rinishi:



24-rasm. Dasturlash maydonida dastur kiritish

3. Dasturga birorta nom berib saqlang.

Natijada oynaning ko`rinishi quyidagicha bo`ladi:



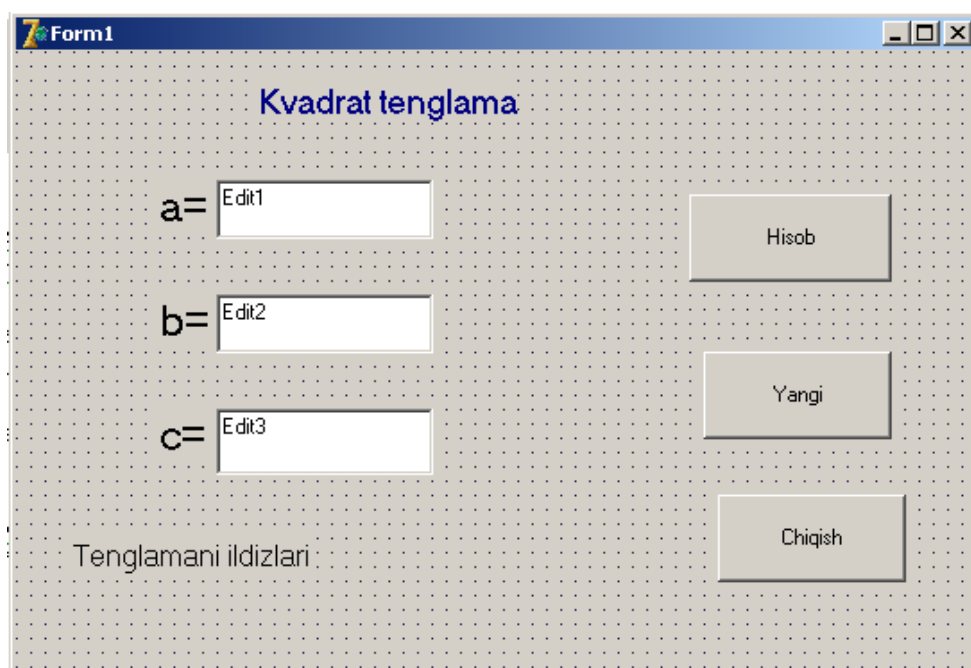
25-rasm. Uchburchakni yuzasini hisoblash dasturi

Izoh : Geron formulasi yordamida uchburchakni yuzasini hisoblash uchun, hosil qilingan dastur maydoniga yangi qiymat kiritish orqali natijani olishimiz mumkin.

6-misol. Kvadrat tenglamani yechish dasturi

1.-bosqich.

1. Dasturni ishga tushiramiz
2. Standard jihozlar panelidan Label1, Label2, Label3, Label4, Label5 ni tanlaymiz.
3. Parametrlar va xossalar bo`limdan foydalanib Label1 ga “Kvadrat tenglama”, Label2 ga “a=”, Label3 ga “b=”, Label4 ga “c=”, Label5 ga “Tenglamani ildizlari=” yozuvini kiritamiz.
4. Standard jihozlar panelidan “Edit1”, “Edit2”, “Edit3” matn kiritish maydonlarini hosil qilamiz
5. Standard jihozlar panelidan “Button1”, “Button2”, “Button3” tugmasi hosil qilamiz hamda parametrlar va xossalar bo`limdan foydalanib “Button1”ga “Hisoblash”, “Button2”ga “Yangilash”, “Button3” ga “Chiqish” matnlarni kiritamiz.



26-rasm. Kvadrat tenglamani hisoblash oynasi

2-bosqich.

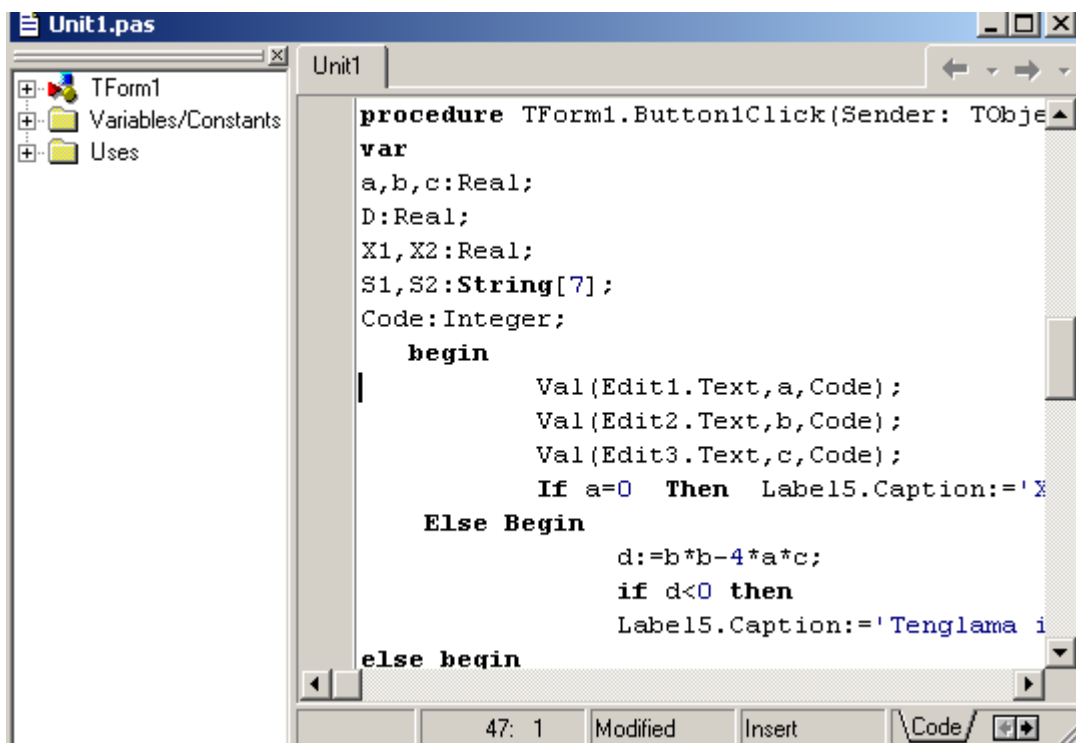
1. Button1(Hisoblash) yozuvi yozilgan tugma yordamida dasturlash maydonini yuklang (Sichqonchaning chap tugmachasini ikki marta bosgan holda).
3. Dasturlash maydoniga quyidagi dasturni kiriting:

```
var  
a,b,c:Real;  
D:Real;  
X1,X2:Real;  
S1,S2:String[7];  
Code:Integer;  
begin
```

```

Val(Edit1.Text,a,Code);
Val(Edit2.Text,b,Code);
Val(Edit3.Text,c,Code);
If a=0 Then Label6.Caption:='Xato!'
Else Begin
    d:=b*b-4*a*c;
    if d<0 then
        Label6.Caption:='Tenglama ildizlari:mavjut emas'
else begin
x1:=(-b+Sqrt(d))/(2*a);
x2:=(-b-Sqrt(d))/(2*a);
Str(x1:7:3,S1);
Str(x2:7:3,S2);
Label6.Caption:='Tenglama ildizlari:'
+Chr(13)+'x1='+S1
+Chr(13)+'x2='+S2;
end;
end;
end;

```



27-rasm. Dasturlash maydonida dastur kiritish

4. Dasturlash maydonidan “F12” yordamida chiqamiz va Button2 (Yangilash) yozuvi yozilgan tugma yordamida dasturlash maydonini yuklaymiz.
5. Dasturlash maydoniga quyidagi dasturni kiritamiz.

```

Begin
Edit1.Text:= ' ';
Edit2.Text:= ' ';

```

```

Edit3.Text:=’ ‘;
Label5.Caption:=’ ‘;
Edit1.SetFocus;
End;

```

6. Button3 (Chiqish) yozuvi yozilgan tugmaning dasturlash maydoniga quyidagi dasturni kiritamiz.

```

Begin
  Form1.Close;
End;

```

7. Dasturni fayl menyusidagi “Save As...” buyrug`i yordamida biror nom berib kompyuter xotirasiga saqlaymiz.

8. Ikkinchi marta dasturni fayl menyusidagi “Save Project As...” buyrug`i yordamida, biror nom berib kompyuter xotirasiga saqlaymiz.

Dastur kodi va oynani umumiy ko`rinishini keltiramiz:

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Edit1: TEdit;
    Edit2: TEdit;
    Edit3: TEdit;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation

```

```

{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var
a,b,c:Real;
D:Real;
X1,X2:Real;
S1,S2:String[7];
Code:Integer;
begin
    Val(Edit1.Text,a,Code);
    Val(Edit2.Text,b,Code);
    Val(Edit3.Text,c,Code);
    If a=0 Then Label5.Caption:='Xato!'
Else Begin
    d:=b*b-4*a*c;
    if d<0 then
        Label5.Caption:='Tenglama ildizlari:mavjut emas'
else begin
x1:=(-b+Sqrt(d))/(2*a);
x2:=(-b-Sqrt(d))/(2*a);
Str(x1:7:3,S1);
Str(x2:7:3,S2);
Label5.Caption:='Тенглама илдизлари:'
+Chr(13)+'x1='+S1
+Chr(13)+'x2='+S2;
end;
end;
end;
procedure TForm1.Button2Click(Sender: TObject);
begin
Edit1.Text:=' ';
    Edit2.Text:=' ';
    Edit3.Text:=' ';
    Label5.Caption:=' ';
    Edit1.SetFocus;
end;
procedure TForm1.Button3Click(Sender: TObject);
begin
    Form1.Close;
end;
end.

```

28-rasm. Kvadrat tenglamani yechish dasturi

Izoh : Deskirminanti manfiy bo`lmagan har qanday kvadrat tenglamani hisoblash uchun, hosil qilingan dastur maydoniga yangi qiymat kiritish orqali natijani olishimiz mumkin.

7- M i s o l. Klaviaturadan kiritilgan butun sonlarni ishorasi va absolyut qiymatini chiqaradigan, agar nol son kiritrsa, nol degan so`z chiqadigan dastur tuzing.

Keltirilgan misolni yechish uchun dastlab matematik tahlil qilishimiz zarur.

Ixtiyoriy butun sonni kiritish orqali son qanday ekanligini bilish mumkin. Agar ixtiyoriy a son berilgan, $a > 0$ bo`lsa bu son butun musbat sonlar to`plamiga tegishli, aks holda $a < 0$ bo`lsa bu son butun manfiy sonlar to`plamiga tegishli bo`ladi. Agar  $a = 0$  bo`lganda esa, bu son nolga teng bo`ladi.

Dastur kodi va oynani umumiy ko`rinishini keltiramiz:

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls;

type

TForm1 = class(TForm)

Label1: TLabel;

Label2: TLabel;

Edit1: TEdit;

Button1: TButton;

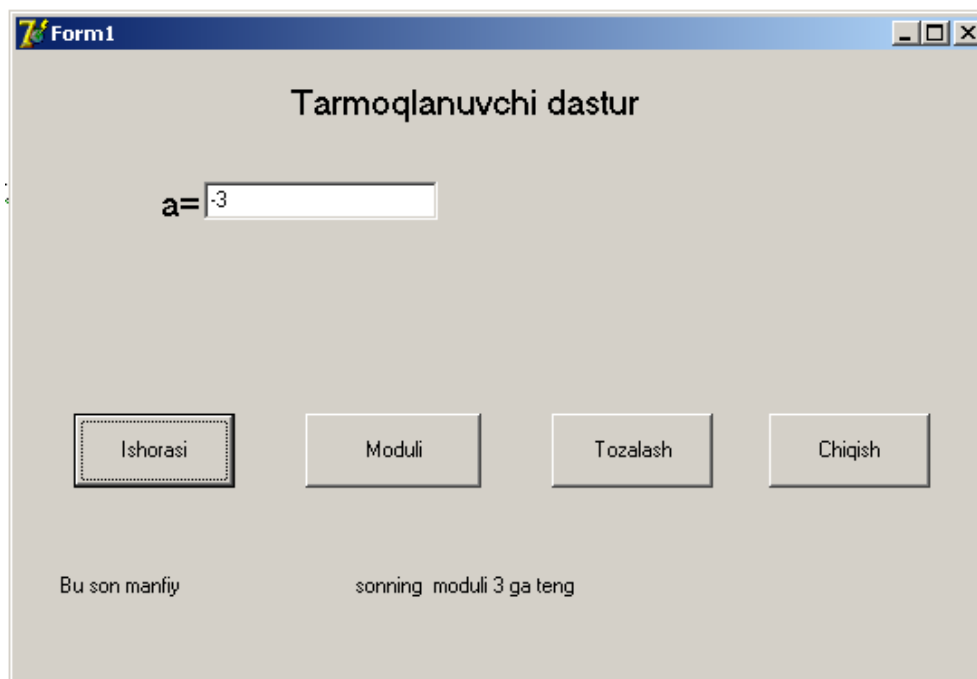
Button2: TButton;

Button3: TButton;

```

    Button4: TButton;
    Label3: TLabel;
    Label4: TLabel;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var a:real;
begin
    a:=strtofloat(edit1.text);
        if a>0 then label2.Caption:=' Bu son musbat ' else
            label2.Caption:='Bu son manfiy ';
    if a=0 then label2.Caption:=' Bu son nolga teng ';
end;
procedure TForm1.Button2Click(Sender: TObject);
var a,a1:real;
begin
    a:=strtofloat(edit1.text);
    a1:=abs(a);
        label3.Caption:=' sonning moduli '+floattostr(a1)+' ga teng`;
end;
procedure TForm1.Button3Click(Sender: TObject);
begin
    edit1.Text:='';
        label2.Caption:='';
    edit1.SetFocus;
end;
    procedure TForm1.Button4Click(Sender: TObject);
begin
    form1.Close;
end;
end.

```



39-rasm. Sonni ishorasini va modulini hisoblash dasturi

Izoh : Har qanday sonni ishorasini va modulini aniqlashimiz uchun, hosil qilingan dastur maydoniga yangi qiymat kiritish orqali natijani olishimiz mumkin.

8- M i s o l. O`. Mirsanovning 1 oylik xarajatini hisoblash dasturini tuzing

Bu tarzdagil masalani yechish uchun 1 oyda ishlatiladigan xarajatlar ro`yxatini tuzib olishimiz kerak. Misol tariqasida xarajatlar ro`yxati quyidagicha bo`lsin:

1. Ijara haqi;
2. Telefon xizmati;
3. Yeb ichish xarajati;
4. Yo`l haqi;
5. Kiyim –kechak xarajati;
6. Ba`zi bir xarajatlar.

Dastur kodi va oynani umumiy ko`rinishini keltiramiz:

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls;

type

TForm1 = class(TForm)

Label1: TLabel;

Label2: TLabel;

Edit1: TEdit;

Label3: TLabel;

```

    Edit2: TEdit;
    Label4: TLabel;
    Edit3: TEdit;
    Label5: TLabel;
    Edit4: TEdit;
    Label6: TLabel;
    Edit5: TEdit;
    Label7: TLabel;
    Edit6: TEdit;
    Label8: TLabel;
    Edit7: TEdit;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var
    s,a1,a2,a3,a4,a5,a6,a7: real;
begin
    a1:=strtofloat(edit1.Text);
    a2:=strtofloat(edit2.Text);
    a3:=strtofloat(edit3.Text);
    a4:=strtofloat(edit4.Text);
    a5:=strtofloat(edit5.Text);
    a6:=strtofloat(edit6.Text);
    a7:=strtofloat(edit7.Text);
    s:=a7-(a1+a2+a3+a4+a5+a6);
    if s<0 then showmessage('O`ralboyning qarzi '+floattostr(s)+' sum');
    showmessage('O`ralboyning qolgan puli'+floattostr(s)+' sum');
end;
end.

```

30-rasm. Xarajatlarni hisoblash dasturi

Izoh : Yuqorida keltirilgan xarajatlarga o`xshash xarajatlarni hisoblash uchun, hosil qilingan dastur maydoniga yangi qiymat kiritish orqali natijani olishimiz mumkin.

8- m i s o l. Ushbu

$$1. y = \begin{cases} \frac{\sin x}{x^2 + 1}, & \text{agar } x < -1, \\ \sin^2 + e^{\sqrt[3]{x}}, & \text{agar } -1 \leq x \leq 1, \\ \ln(x + 1,8), & \text{agar } x > 1. \end{cases}$$

funksiya qiymatini hisoblash dasturini tuzing.

Dastur kodi va oynani umumiy ko`rinishini keltiramiz:

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs;

type

TForm1 = class(TForm)

Label1: TLabel;

Label2: TLabel;

Edit1: TEdit;

Button1: TButton;

procedure Button1Click(Sender: TObject);

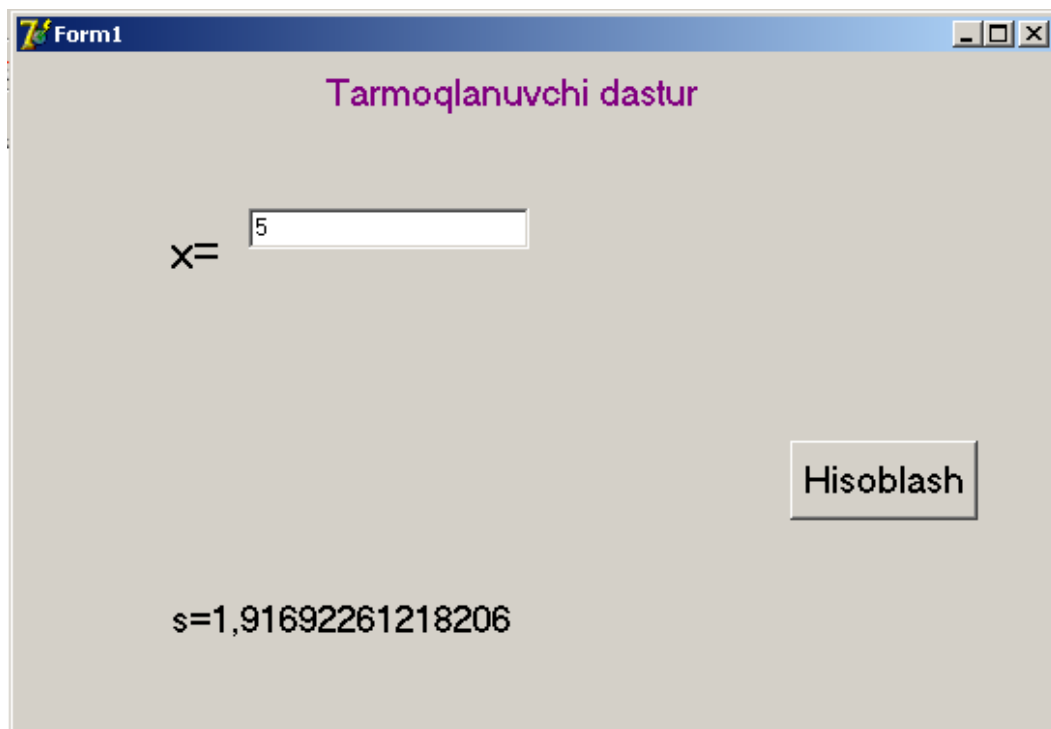
private

{ Private declarations }

```

public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
Procedure TForm1.Button1Click(Sender: TObject);
label 1,2,3;
var
  s,x:real;
begin
  x:=strtofloat(edit1.Text);
  if x<-1 then goto 1;
  if x>1 then goto 2;
  s:=sqr(sin(x))+exp(ln(x)/3);
  goto 3;
  1:s:=sin(x)/(sqr(x)+1);
  goto 3;
  2:s:=ln(x+1.8);
  3:label2.Caption:='s='+floattostr(s);
end;
end.

```



31-rasm. Funksiya qiymatini hisoblash dasturi

Izoh : Ko`rsatilgan funksiyadagi x ni boshqa qiymatlarini hisoblash uchun, hosil qilingan dastur maydoniga yangi qiymat kiritish orqali natijani olishimiz mumkin.

Mustaqil bajarish uchun mashqlar

Parametrning ma'lum qiymatlarida berilgan funksiyaning qiymatini hisoblash oynasi va dasturini tuzing.

$$1. \ y = \begin{cases} \sqrt[3]{x^2 + a^2}, & \text{agar} \quad x^2 + a^2 < 4, \\ x + a, & \text{agar} \quad x^2 + a^2 = 4, \\ \ln x, & \text{agar} \quad x^2 + a^2 > 4. \end{cases}$$

$$2. \ y = \begin{cases} 3x^2 + 4\sqrt{a-x}, & \text{agar} \quad a > x, \\ \sin \sqrt{3a+x}, & \text{agar} \quad a = x, \\ e^{\sqrt{x+a}}, & \text{agar} \quad a < x. \end{cases}$$

$$3. \ y = \begin{cases} \sin^2 x & \text{agar} \quad |x| < \frac{\pi}{4}, \\ \sin(\operatorname{tg} x), & \text{agar} \quad |x| = \frac{\pi}{4}. \end{cases}$$

$$4. \ y = \begin{cases} a \sin \omega t, & \text{agar} \quad |a - \omega t| < 1, \\ a \cos \omega t, & \text{agar} \quad |a - \omega t| = 1, \\ a \operatorname{tg} \omega t, & \text{agar} \quad |a - \omega t| > 1. \end{cases}$$

$$5. \ y = \begin{cases} \sin \sqrt{x} & \text{agar} \quad x = \frac{\pi}{2}, \\ \operatorname{tg} \sqrt{x+1}, & \text{agar} \quad x > \frac{\pi}{2}, \\ \sin(\cos x), & \text{agar} \quad x < \frac{\pi}{2}. \end{cases}$$

$$6. \ y = \begin{cases} \sqrt[5]{x-\pi} + \sin \frac{\pi}{2} & \text{agar} \quad x = \frac{\pi}{2}, \\ \operatorname{arctg} \sqrt{x+1} & \text{agar} \quad x > \frac{\pi}{2}, \\ \sin(\cos x), & \text{agar} \quad x < \frac{\pi}{2}. \end{cases}$$

$$7. \ y = \begin{cases} 3^{x-1}, & \text{agar} \quad x > 1, \\ (x-1)^3, & \text{agar} \quad x < 1, \\ 0, & \text{agar} \quad x = 1. \end{cases}$$

$$8. \quad y = \begin{cases} x^a + \ln|x+a|, & \text{agar } x < 2|a|, \\ x^2 + e^{x-a}, & \text{agar } x = 2|a|, \\ a^x + \sin \sqrt{x-a}, & \text{agar } x > 2|a|. \end{cases}$$

$$9. \quad y = \begin{cases} \frac{x^2 - x}{a^{x-1}}, & \text{agar } x > a, \\ x^2 - 2a, & \text{agar } x \leq a. \end{cases}$$

$$10. \quad y = \begin{cases} \lg x + a^x, & \text{agar } x < a, \\ 0,5\sqrt{1+ax^2}, & \text{agar } x = a, \\ \ln|\sin x|, & \text{agar } x > a. \end{cases}$$

$$11. \quad y = \begin{cases} 4,7x + 1,2^x, & \text{agar } x < 1, \\ 2,5x + \ln x + 1,2, & \text{agar } 1 \leq x \leq 2, \\ x^2 + 1,2x, & \text{agar } x > 2. \end{cases}$$

$$12. \quad y = \begin{cases} ax^3 + 3\ln|a+x|, & \text{agar } |ax| < 1, \\ 2ax + 3a^2x^2, & \text{agar } |ax| = 1, \\ e^{\sqrt{ax+\sin x}}, & \text{agar } |ax| > 1. \end{cases}$$

$$13. \quad y = \begin{cases} \sin \frac{x}{\sqrt{3x+x^2}}, & \text{agar } x < \frac{\pi}{2}, \\ 2\sqrt{x+\sin(x-2)}, & \text{agar } x \geq \frac{\pi}{2}. \end{cases}$$

14. Tekislikda uchta nuqta o'zining koordinatalari bilan berilgan. Shu nuqtalardan qaysilari bir-biridan maksimal uzoqlikda joylashgan.

15. $\max^3(a,b,c) + \min^2(a,b,c)$ hisoblang.

16. (x,y) koordinatalar bilan berilgan nuqta, koordinatalar tekisligining qaysi choragida yotadi.

2.3. Takrorlanuvchi jarayonlarga oid dasturlar tuzish

Takrorlanish jarayonlarini algoritmlarni ham Delphi dasturlash tilida dasturini tuzishdan oldin, Pascal tilida ko`rib chiqish kerak. Chunkiy yuqorida ta'kitlab o`tilgandek Delphi dasturlash tilida Pascal dasturlash tiliga asoslanib dastur tuziladi.

Ayrim masalalarda bir yoki bir necha parametrlarning o`zgarishiga qarab ma'lum hisoblashlar bir necha marta takrorlanib bajarilishi mumkin. Masalan,  $y=kx+b$  funksiyani  $x$  ning bir necha qiymati mos qiymatlarini hisoblash zarur bo`lsin. 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. Pascal tilida siklik strukturali dastur tuzishda bir necha maxsus operatorlar mavjud. For operatori takrorlanishlar soni aniq bo`lgan siklik jarayonlar tashkil etishda ishlatiladi. Uning umumiy ko`rinishi quyidagicha:

1) **For i:=m1 to m2 Do S;**

Bu erda i-sikl parametri; m1, m2 -i parametrining boshlang`ich 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 operatorlardan iborat bo`lsa, ular Begin va End ichiga olinadi.

Misol: Pascal tilida 1 dan n gacha ketma-ket butun natural sonlarni yig`indisini hisoblaydigan dastur tuzing.

Program dastur;

Var i,n,S: Integer;

Begin

S:=0;

For i:=1 to n do S:=S+i;

Write ('S=',S);

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 S;

Misol: 20 dan 1 gacha sonlarni ekranga chiqarish dastursini tuzing.

Program yigindi;

Var i: Integer;

Begin

For i:=20 DoWnto 1 do Write (i);

End.

2) 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 sonlar yig`indisini hisoblash dastursini tuzing.

Program summa1;

Const kn=10;

Var i: Integer; S: Real;

Begin

S:=0; i:=0;

While i<=kn 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`tdi, 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,...,100 sonlar yig`indisini hisoblash dastursini tuzing.

Program summa;

Const kn=100;

Var i: Integer; S: Real;

Begin

S:=0; i:=0;

Repeat

i:=i+1; S:=S+i;;

Until I>kn;

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 muvofiqdir. 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.

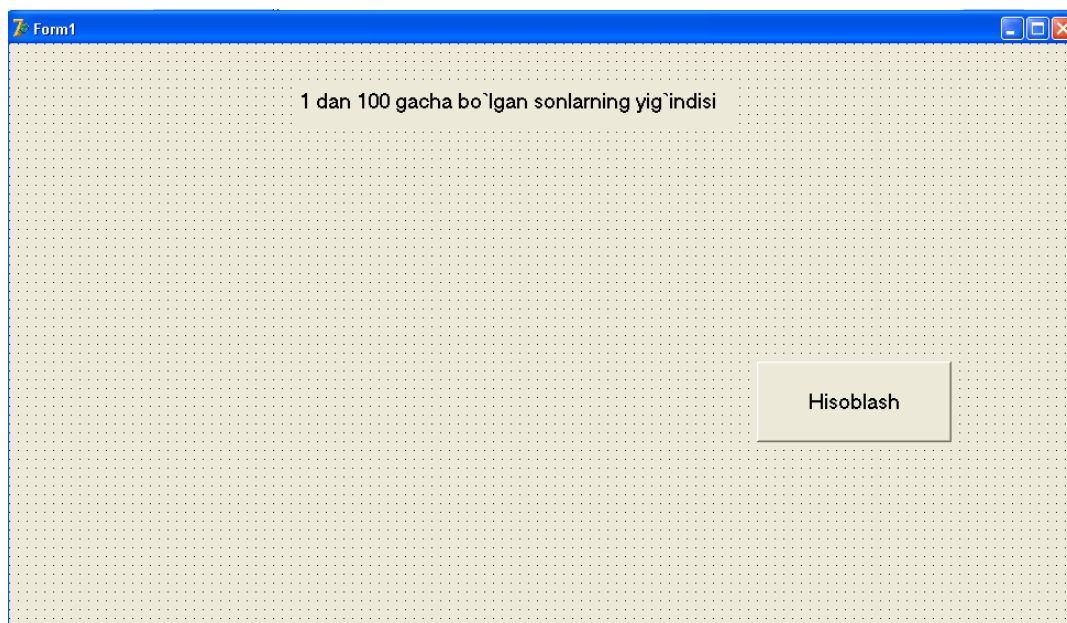
Yuqorida bayon qilingan fikrlarni tekshirib ko`rish uchun Delphi dasturlash tilida, sikl operatorlaridan foydalangan holda bir nechta dasturlar tuzamiz.

9 - m i s o l. 1 dan 100 gacha ketma- ket keluvchi natural sonlarning yig`insisini hisoblash dasturini tuzing.

Bu misolni matematik nuqtaiy –nazardan tahlil qiladigan bo`lsak quyidagicha:  $1+2+3+\dots+100=5050$  ko`rinish hosil bo`ladi.

Dasturni loyihasi juda oson. Bizga asosan bitta “Button” tugmasidan foydalanishimiz kifoya. Tugmadan natijani chiarish uchun foydalanamiz. Yana qo`shimcha tugmalarni ba`zi vazifalar uchun kiritishimiz mumkin.

Oyna holatining umumiy ko`rinishini quyidagicha bo`ladi:



32-rasm. Tarmoqlanuvchi jarayonlarni dasturlash oynasi

“Hisoblash” tugmasi yordamida dasturlash maydonini ishga tushiramiz va

var

i,s:integer;

begin

s:=0;

for i:=1 to 100 do

s:=s+i;

showmessage('Sonning yig`indisi S='+floattostr(S)+' ga teng');

end;

dasturni kiritamiz va kerakli nom bilan kompyuter xotirasiga saqlaymiz.

Dastur kodi va oynani umumiy ko`rinishini keltiramiz:

Unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls;

type

Tform1 = class(Tform)

Label1: TLabel;

Button1: Tbutton;

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: Tform1;

implementation

{\$R *.dfm}

procedure Tform1.Button1Click(Sender: TObject);

var

I,s:integer;

begin

s:=0;

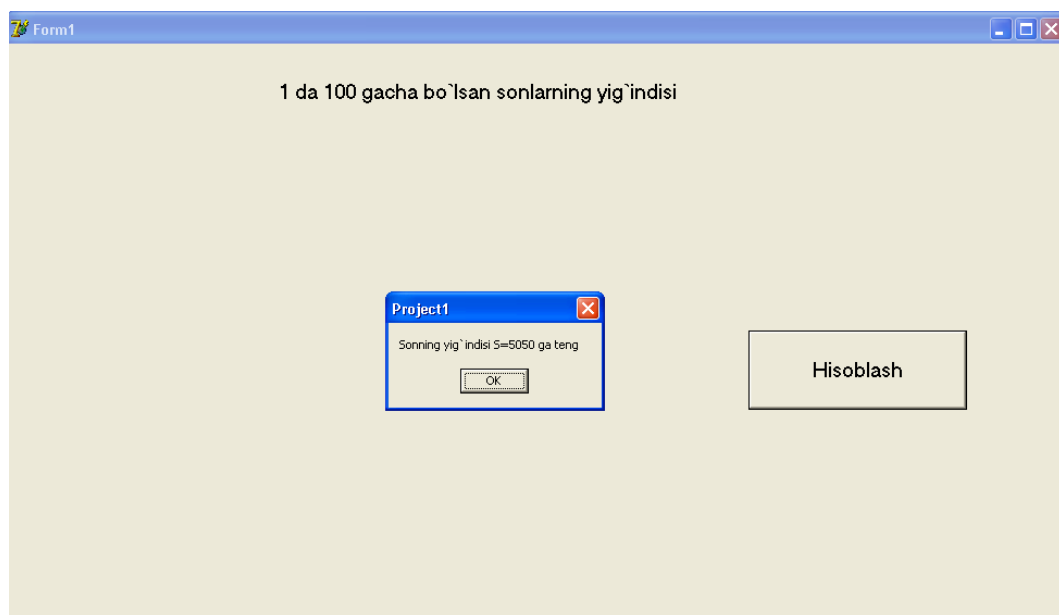
for i:=1 to 100 do

s:=s+I;

showmessage('Sonning yig`indisi S='+floattostr(S)+' ga teng');

end;

end.



33-rasm. 1 dan 100 gacha bo`lgan sonlarning yig`indisini hisoblash dasturi

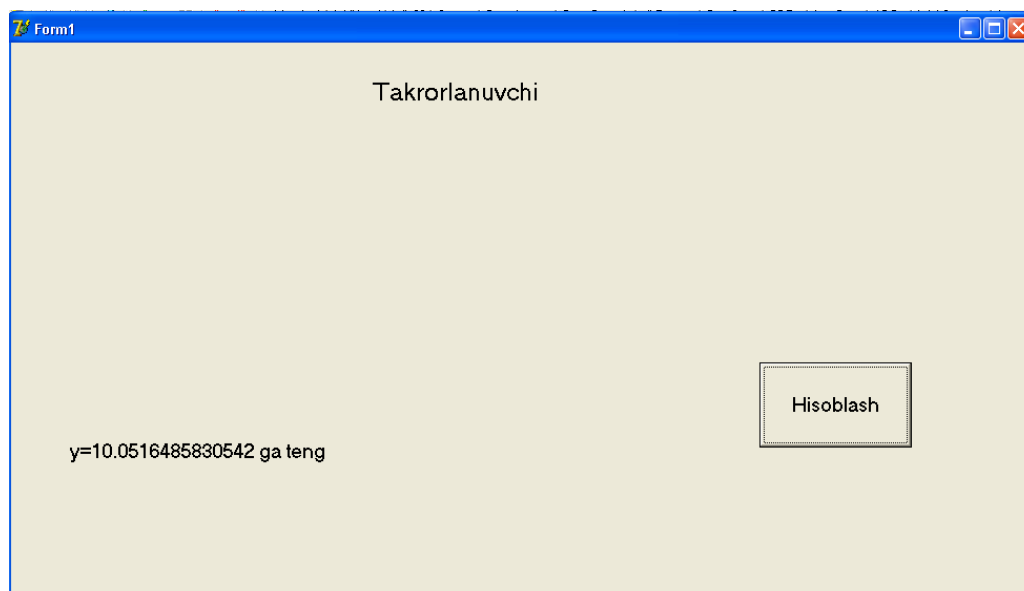
10 – m I s o l. Ushbu $y = \prod_{i=1}^6 \sum_{k=1}^5 \frac{k+i}{2k+3i}$ jarayonni mos oyna va dastur modulini tuzing.

Dastur kodi va oynani umumiy ko`rinishini keltiramiz:

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Label1: TLabel;
    Button1: TButton;
    Label2: TLabel;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var
  k,i:integer;
  y,p:real;
begin
  p:=0;
  for k:=1 to 5 do begin
  y:=1;
  for i:=1 to 5 do begin
  p:=p+(k+i)/(2*k+3*i);
  end;
  y:=y*p;
  Label2.caption:='y='+floattostr(y)+' ga teng`;
end;
  end;
end.

```



34-rasm. Takrorlanuvchi dastur

Mustaqil bajarish uchun mashqlar

1. Raqamlari yig`indisining kubiga teng bo`lgan barcha uch xonali sonlarni toping.
2. 2,3,4,5,6,7,8,9 sonlariga ko`paytirilganda raqmlarining yig`indisi o`zgamaydigan ikki xonali sonlarni toping.
3. Raqamlari yig`indisi berilgan butun songa teng uch xonali barcha sonlarni toping.
4. Barcha shunday uch xonali sonlarni topingki, ularni kvadratga ko`targanda uchata raqami bir xil bo`lsin, nol bundan mustasno.
5. Do`konda 16 kgli, 17 kgli va 21 kgli pol buyoqlari yashiklarda mavjud. Ularni ochib ko`rmasdan 185 kg pol buyoq tanlang. Barcha variantlarni ko`rib chiqing.
6. $42 \cdot 4^*$ yulduzchalar o`rniga shunday raqamlarni tanlang, hosil bo`lgan besh xonali son 72ga bo`linsin.
7. O`zi va raqamlari yig`indisi 7ga karrali uch xonali sonlarni toping.
8. Shunday to`rt xonali sonlarni topingki, 133ga bo`lganda qoldiq 125ga va 134ga bo`lganda qoldiq 111ga teng bo`lsin.
9. Raqamlari yig`indisi berilgan butun songa teng to`rt xonali barcha sonlarni toping.
10. O`zi va raqamlari yig`indisi 11ga karrali to`rt xonali sonlarni toping.
11. $42 \cdot 4^*$ yulduzchalar o`rniga shunday raqamlarni tanlang, hosil bo`lgan olti xonali son 11ga bo`linsin.
12. Berilgan natural 10 sonni bo`luvchilari yig`indisini toping.
13. Berilgan N sonidan oshmaydigan barcha pifagor natural sonnlarni toping. (Agar berilgan uch xonali sonning dastlabki ikkita raqami yig`indisining kvadrati uchinchi raqamiga teng bo`lsa, bunday natural son pifagor soni deb ataladi)
14. Birinchi ikkita raqamlari yig`indisi oxirgi ikkita raqamlari yig`indisiga teng bo`lgan barcha to`rt xonali sonlarni toping.

Quyidagilarni:

$$15. \prod_{k=1}^3 \prod_{n=3}^5 \frac{\sqrt{n^k + 1}}{\ln n}$$

$$16. \sum_{l=1}^4 \prod_{n=1}^5 \frac{\sqrt{i+2n}}{\cos(n-i)}$$

$$17. \sum_{k=2}^4 \prod_{i=3}^6 \frac{\ln i + k^i}{\sqrt[k]{i+k}}$$

$$18. \prod_{k=1}^4 \prod_{n=2}^5 \frac{\cos(n^2 - k)}{3nk}$$

$$19. \sum_{k=2}^6 \prod_{n=k}^5 \frac{\ln k}{\sqrt{k+n^2}}$$

$$20. \sum_{k=2}^5 \sum_{n=4}^6 \frac{n^3}{k^4 + n^k}$$

$$21. \prod_{n=1}^6 \sum_{m=1}^{10} \frac{\sqrt{n+m}}{n^{m+1}}$$

$$22. \sum_{n=1}^4 \prod_{m=1}^n \frac{\sqrt{n^m - m^n}}{n^m + m^n}$$

$$23. \sum_{k=1}^4 \prod_{n=3}^6 \frac{n^k}{n^2 + 3k + 1}$$

$$24. \prod_{n=2}^4 \sum_{m=n}^6 \sqrt{tg(2n+m)}$$

$$25. \sum_{i=1}^5 \sum_{k=1}^4 \frac{2k^i + 5^i}{\ln|k+1|^2}$$

$$26. \prod_{k=1}^4 \prod_{n=2}^5 \frac{(k+n)^{\frac{1}{n}}}{n^{k+1}}$$

$$27. \sum_{k=1}^5 \prod_{i=k}^4 \frac{k+l^2}{3^{k+1}}$$

$$28. \prod_{k=1}^5 \sum_{n=2}^7 \frac{k^n + 1,5}{\sqrt{n+k^2}}$$

$$29. \sum_{i=z}^4 \sum_{k=1}^7 \frac{tgk}{k^i - k}$$

dasturini tuzing.

III BOB. DELPHI DASTURLASH TILIDA MASSIVLAR

3.1. Delphi dasturlash tilida massivlar bilan ishlash

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 kvadrat 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'lchovli massiv elementi ikkita indeks orqali yoziladi. Masalan: $A[i,j]$ bu yerda 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: $A,B:\text{array}[1..10]$ of real; $C,A1,D1:\text{Array}[1..10,1..20]$ of real;

Bu yerda A va B massivlari 10 tadan elementga ega. $C,A1,D1$ massivlari esa $10 \times 20 = 200$ 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 100 do Read($A[i]$);

Bu misolda A massivning 100 ta elementi qiymatini ekrandan ketma-ket kiritish kerak bo'ladi. Xuddi shunday massiv qiymatlarini ekranga chiqarish ham mumkin.

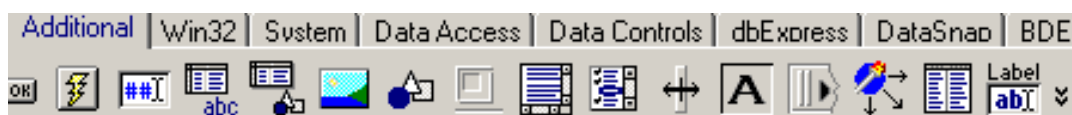
Misol: For $i:=1$ to 100 do Write($A[i]$);

Dasturda massiv elementlarini ishlatganda ularning indeksi e'lon qilingan chegaradan chiqib ketmasligi kerak. Yuqoridagi ma'lumotlarni tasdiqlash uchun, massivlarga oid bir nechta misollar ko'rib o'tamiz.

15-Misol. $A(5)$ massiv elementlarini yig'indisi va o'rta arifimetigini topish dasturi tuzilsin.

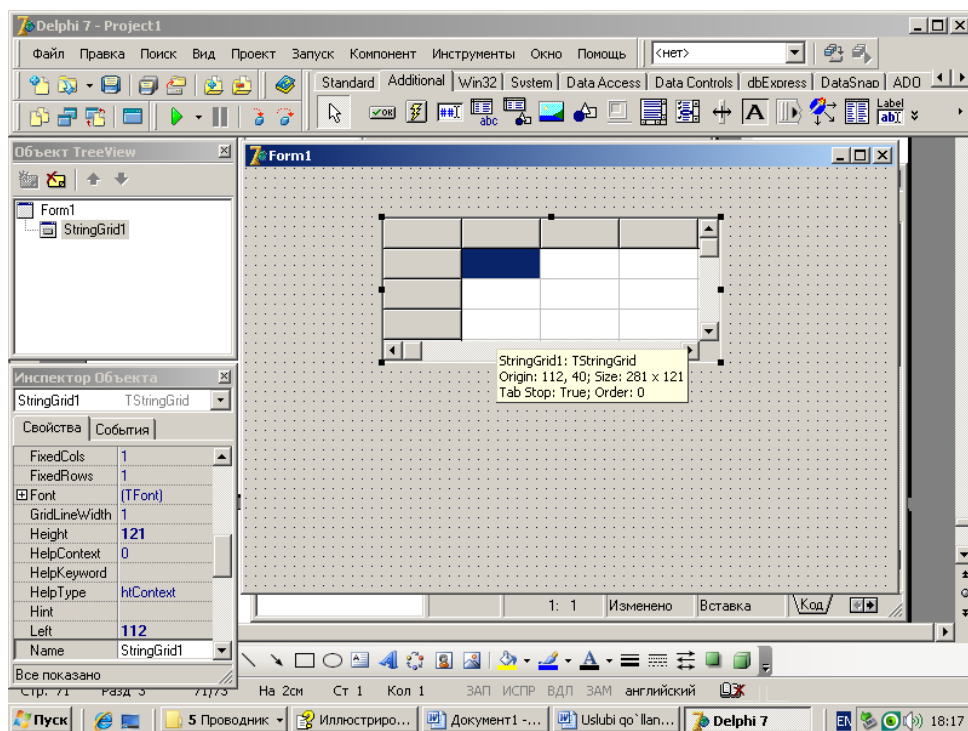
1-bosqich.

1. Komponentalar ro'yxatini Additional bandidan **StringGrid1(abc)** tugmani tanlaymiz 41-rasmda ko'rsatilgan.



41-rasm. Additional panilining ko`rinishi

Tugma tanlangandan so`ng, oynaning ko`rinish holati quyidagicha bo`ladi:



42-rasm. StringGrid1(abc) tugmasidan foydalanish

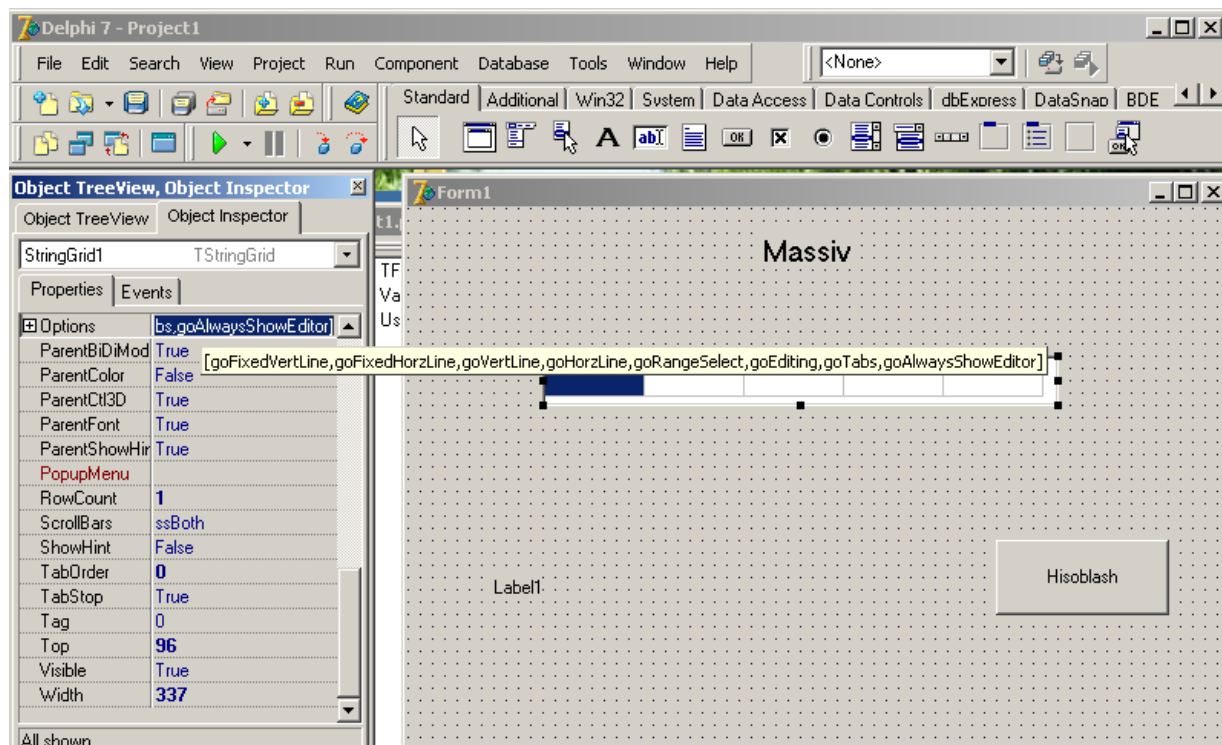
Hosil bo`lgan jadvaldan 5 ta ustun, 1 ta satr yarating. Bu jadvalni yaratish uchun, parametrlar va xossalar bo`limidan **ColCount** hodisalar bandiga 5 raqamini kiritamiz. Satr bo`yicha o`zgartirish kiritish uchun, **RowCount** bandiga 1 raqamini kiritamiz.

Umumiy holda oynani ishga tayyorlashimiz uchun parametrlar va xossalar bo`limidan jadvalda ko`rsatilgan buyruqlarning tarkibi quyidagicha:

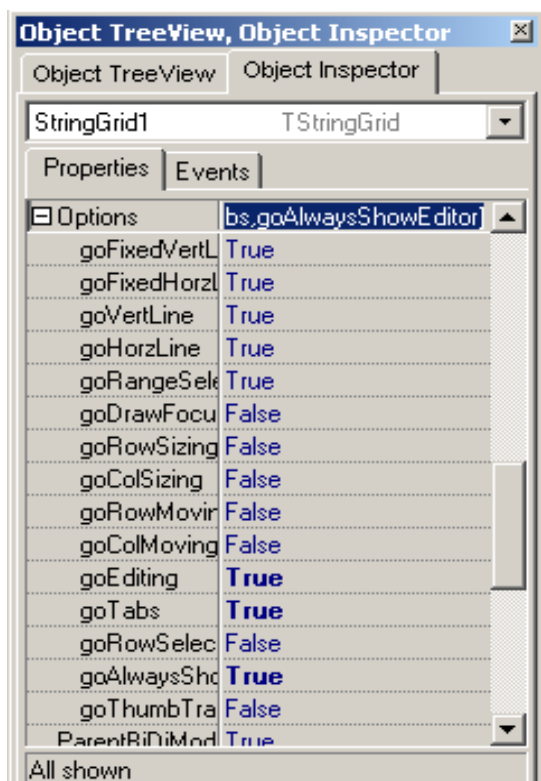
9-jadval

Parametrlar va xossalar bo`limi	Hodisalar bo`limi
ColCount	5
FixedCols	0
RowCount	1
DefaultRowHeight	24
Height	24
DefaultColWidth	64
Width	328
Options . goEditing	True
Options . AlwaysShowEditing	True
Options .goTabs	True

Jadvalda ko`rsatilgan 3 ta **Options . goEditing, Options . AlwaysShowEditing, Options . goTabs** buyruqlarni hosil qilishimiz uchun parametrlar va xossalar bo`limidan “Options” parametirining “+” ishorasini “-” ishoraga o`tkazamiz (jadval ustiga sichqoncha tugmasini bir marta bosganimizdan so`ng, aytilgan topshiriqni bajarishimiz mumkin).



43-rasm. StringGrid1(abc) tugmasidan foydalanish



44-rasm. Options xossasidan foydalanish

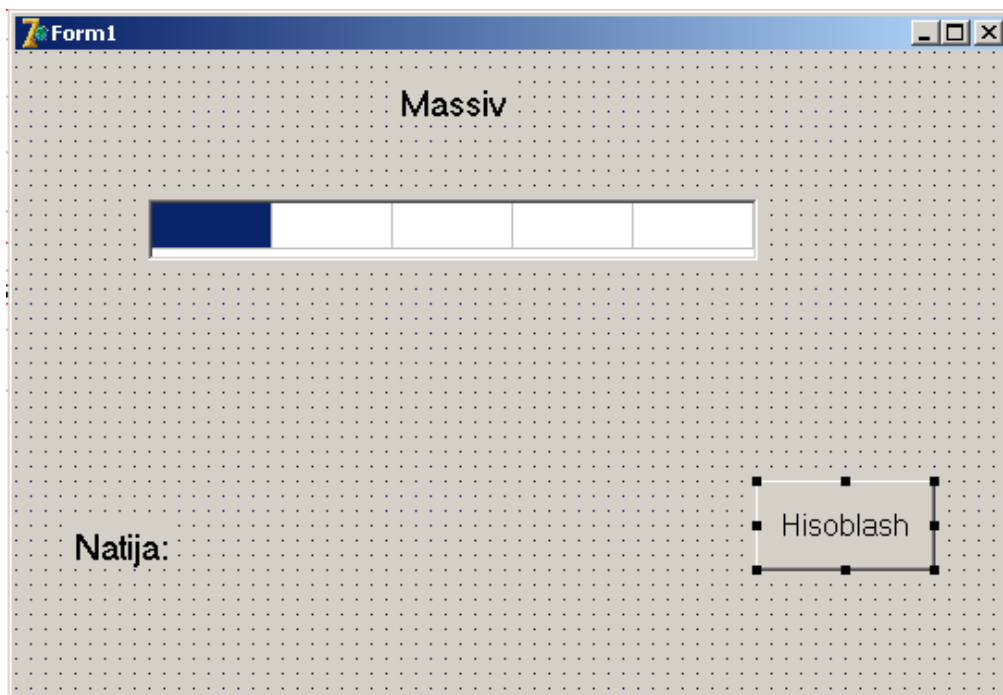
Hodisalar bo`limining **goEditing** bandida 2 ta **False** va **True** buyrug`i mavjud. **True** buyrug`ini tanlasangiz katakchaga ma`lumot kiritgandan so`ng dastur bilan bog`lanish hosil qiladi aks holda ya`ni **False** buyrug`i tanlasangiz kiritilgan ma`lumot dastur bilan bog`lashga ruxsat etmaydi.

Parametrlar va xossalar bo`limidan “Options” parametirini “+” ishorasini “-” ishoraga o`tkazganimizdan so`ng oynani ko`rinishi quyidagicha bo`ladi:

2. “Label1” tugmasidan foydalanib “Massiv” so`zini kiritamiz.

3. “Label2” tugmasiga “Natija” so`zini kiritamiz.

Natijada oynaning umumiy ko`rinishi quyidagicha bo`ladi:



45-rasm. Dasturlashda “Button” tugmasidan foydalanish

2-bosqich.

1. Dasturlash maydoniga quyidagi dasturni kiritamiz:

```
unit Unit1;
```

```
interface
```

```
uses
```

```
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, StdCtrls, Grids;
```

```
Type
```

```
TForm1 = class(TForm)
```

```
  StringGrid1: TStringGrid;
```

```
  Label1: TLabel;
```

```
  Button1: TButton;
```

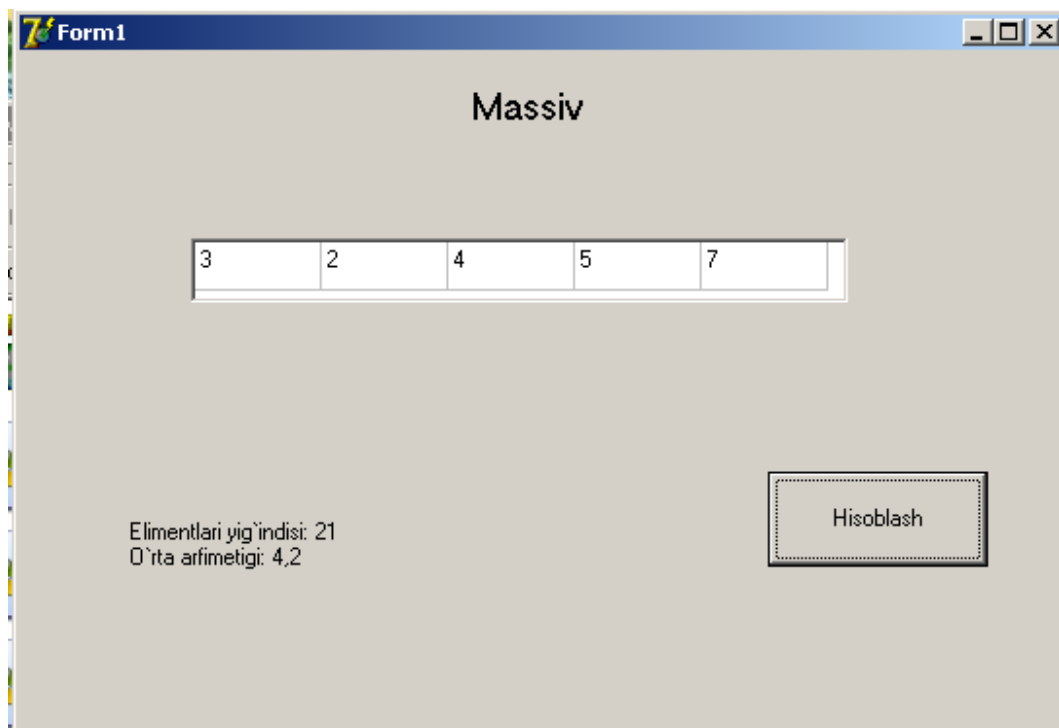
```

Label2: TLabel;
procedure Button1Click(Sender: TObject);
procedure StringGrid1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var
  a : array[1..5] of integer;
  summ: integer;
  sr: real;
  i: integer;
begin
  for i:= 1 to 5 do
    if Length(StringGrid1.Cells[i-1, 0]) <>0
    then a[i] := StrToInt(StringGrid1.Cells[i-1,0])
    else a[i] := 0;
  summ := 0;
  for i :=1 to 5 do
  summ := summ + a[i]; sr := summ / 5;
Label1.Caption :=
  'Elimentlari yig`indisi: ' + IntToStr(summ)
  + #13+ 'O`rta arifimetigi: ' + FloatToStr(sr);
end;
  procedure TForm1.StringGrid1Click(Sender: TObject);
var Key: Char;
begin
  case Key of
    #8,'0'..'9' : ;
  #13:
    if StringGrid1.Col < StringGrid1.ColCount-1
    then StringGrid1.Col := StringGrid1.Col + 1;
    else key := Chr(0);
  end;
end;
end.

```

3. Dastur kiritilganidan so`ng, biron bir nom bilan saqlaymiz.
4. Dastur ishini yakunlab ma'lum bir qiymatlarini yachyekalarga kiritamiz

Oynani umumiy ko`rinishi keltiramiz:



46-rasm. 5 o`lchovli massiv elementlari yig`indisini va o`rta arifimetigini hisoblash dasturi

Izoh : Har qanday 5 o`lchovli massiv elementlari yig`indisini va o`rta arifimetigini hisoblash uchun, hosil qilingan dastur maydoniga yangi qiymat kiritish orqali natijani olishimiz mumkin.

16-Misol. $A(5)$ massiv elementlarining eng kichchigina va eliment nomerini topish dasturini tuzing.

Dastur kodi va oynani umumiy ko`rinishini keltiramiz:

Unit1;

interface

uses

**Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls, Grids;**

type

TForm1 = class(TForm)

StringGrid1: TStringGrid;

Label1: TLabel;

Label2: TLabel;

Button1: TButton;

procedure Button1Click(Sender: TObject);

procedure StringGrid1Click(Sender: TObject);

```

private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
const
  k=5;
var
  a:array[1..k]of integer;
  min:integer;
  i:integer;
begin
  for i:=1 to k do
  a[i]:=StrToInt(StringGrid1.Cells[i-1,0]);
  min:=1;
  for i:=2 to k do
  if a[i]< a[min]then min:=i;
  label2.caption:='Eng kichchik elimenti:'
  +IntToStr(a[min])+#13+'Eliment nomeri:'+ IntToStr(min);
end;
  procedure TForm1.StringGrid1Click(Sender: TObject);
  var Key: Char;
  begin
    case Key of
      #8,'0'..'9': ;
      #13:
        if StringGrid1.Col < StringGrid1.ColCount-1
        then StringGrid1.Col:= StringGrid1.Col + 1;
        else key := Chr(0);
        end;
      end;
    end.

```

47-rasm. Massiv elimentlari ustida amallari

Izoh : Har qanday 5 o`lchovli massiv elementlarining eng kichchigi va eliment nomerini toppish uchun, hosil qilingan dastur maydoniga yangi qiymat kiritish orqali natijani olishimiz mumkin.

17-Misol. $A(5)$ massiv elementlarini o`lish tartibi bo`yicha tartiblash dasturini tuzing.

Dastur kodi va oynani umumiy ko`rinishini keltiramiz:

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls, Grids;

type

TForm1 = class(TForm)

StringGrid1: TStringGrid;

Label1: TLabel;

Label2: TLabel;

Button1: TButton;

procedure Button1Click(Sender: TObject);

procedure StringGrid1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

```

Form1: TForm1;
implementation
{ $R *.dfm }
procedure TForm1.Button1Click(Sender: TObject);
const
n=5;
var
a:array[1..n] of integer;
min:integer;
j:integer;
buf:integer;
i,k:integer;
begin
for i:=1 to n do
a[i]:=StrToInt(StringGrid1.Cells[i-1,0]) ; label2.caption:='';
for i:=1 to n-1 do begin
  min:=i;
  for j:=i+1 to n do if a[j] < a [min]
  then min:=j;
  buf:=a[i]; a[i]:=a[min]; a[min]:=buf;
  for k:=1 to n do
  Label2.caption:=label2.caption+' '+IntToStr(a[k]);
  Label2.caption:=label2.caption+#13; end;
  Label2.caption:=label2.caption+#13+'Massiv elimentlarini tartiblash.';
end;
procedure TForm1.StringGrid1Click(Sender: TObject);
var Key: Char;
begin
  case Key of
  #8,'0'..'9': ;
  #13:
  if StringGrid1.Col < StringGrid1.ColCount-1
  then StringGrid1.Col:= StringGrid1.Col + 1;
  else key := Chr(0);
  end;
end;
end.

```

48-rasm. 5 o`lchovli massiv elementlarini tartiblash dasturi

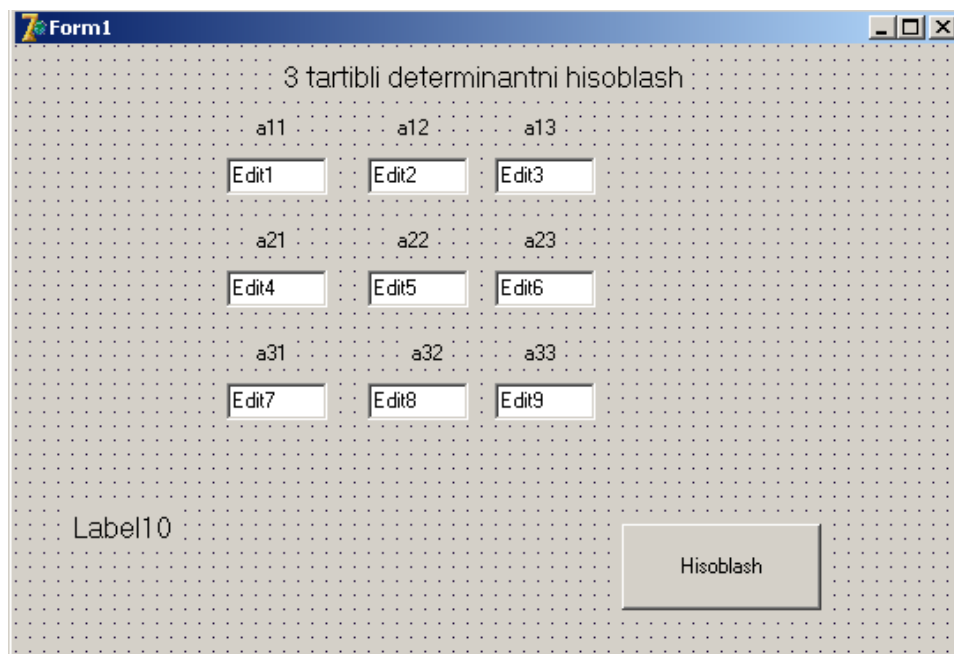
Izoh : Har qanday 5 o`lchovli massiv elementlarining tartiblash uchun, hosil qilingan dastur maydoniga yangi qiymat kiritish orqali natijani olishimiz mumkin.

18-Misol. Uchunchi tartibli determinantni hisoblash dasturini tuzing.

Bizga oily matematika kursida ma'lumki uchunchi tartibli determinantni hisoblash quyidagicha bo`ladi:

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \cdot a_{22} \cdot a_{33} + a_{12} \cdot a_{23} \cdot a_{31} + a_{13} \cdot a_{21} \cdot a_{32} - a_{13} \cdot a_{22} \cdot a_{31} - a_{32} \cdot a_{23} \cdot a_{11} - a_{21} \cdot a_{12} \cdot a_{33}$$

Determinantning hisoblash dasturini tuzishimiz uchun 9 ta haddan foydalanishimizga to`g`ri keladi. Shuning uchun biz 9 ta Edit tugmasi, 11 ta label tugmasi va bitta button tugmasidan foydalanamiz.



49-rasm. Uchunchi tartibli deteminantni hisoblash oynasining loyihasi

Mos dastur modulini hosil qilamiz:

unit Unit1;

interface

uses

**Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls;**

type

TForm1 = class(Tform)

Edit1: Tedit;

Edit2: Tedit;

Edit3: Tedit;

Edit4: Tedit;

Edit5: Tedit;

Edit6: Tedit;

Edit7: Tedit;

Edit8: Tedit;

Edit9: Tedit;

Label1: Tlabel;

Label2: Tlabel;

Label3: Tlabel;

Label4: Tlabel;

Label5: Tlabel;

Label6: Tlabel;

Label7: Tlabel;

Label8: Tlabel;

Label9: Tlabel;

Button1: Tbutton;

Label10: Tlabel;

```

Label11: TLabel;
procedure Button1Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var
  s,a11,a12,a13,a21,a22,a23,a31,a32,a33: real;
begin
  a11:=strtofloat(edit1.Text);
  a12:=strtofloat(edit2.Text);
  a13:=strtofloat(edit3.Text);
  a21:=strtofloat(edit4.Text);
  a22:=strtofloat(edit5.Text);
  a23:=strtofloat(edit6.Text);
  a31:=strtofloat(edit7.Text);
  a32:=strtofloat(edit8.Text);
  a33:=strtofloat(edit9.Text);
  a12:=strtofloat(edit2.Text);
  s:=a11*a22*a33+a12*a23*a31+a21*a32*a13-(a13*a22*a31-a12*a21*a33-
  a32*a23*a11);
  label10.Caption:='s='+floattostr(s);
end;
end.

```

Mos dastur oynasining umumiy ko`rinishi

51-rasm. Uchunchi tartibli determinantning hisoblash dasturi

Izoh : Har qanday uchunchi tartibli diterminantni hisoblash uchun, hosil qilingan dastur maydoniga yangi qiymat kiritish orqali natijani hosil qilishimiz mumkin.

Mustaqil bajarish uchun mashqlar

1. $y = \frac{\sum_{j=1}^6 \ln |x_i + 2,5|}{\prod_{i=1}^6 x_i^2}$ ni hisoblang

2. $y = \sum_{i=1}^5 \frac{ilz_i}{2^i}$ ni hisoblang

3. $y_i = \prod_{i=1}^3 a_i^2 b_i$ ni hisoblang

4. $A(10)$ vektor elementlarining yig`indisini va o`rta arifmetigini hisoblash dasturini toping.

5. $A(10)$ vektor elementlarining ko`paytmasining va o`rta geometrigini hisoblash dasturini toping.

6. $A(10)$ vektor elementlarini eng kichchigini toppish dasturini chizing

7. $A(10)$ vektor elementlarini eng kattasini toppish dasturini chizing

8. $A(5)$ vektor elementlarini o`sib borish tartibida joylashtiring

9. $A(5)$ vektor elementlarini kamayib boorish tartibida joylashtiring

10. Uchunchi tartibli kvadrat matrisani teskari matritsasini toppish dasturini tuzing

11. Ikkinchi tartibli determinantni hisoblash dasturini tuzing

12. To`rtinchi tartibli determinantni hisoblash dasturini tuzing

IV BOB. DELPHI DASTURLASH TILIDA GRAFIKLAR

4.1. Delphi dasturlash tilida ba'zi bir matematik funksiyalarni grafigini yasash

Delphi dasturlash tilida dasturchiga grafik dasturlar sxema va chizma, illyustratsiyalar yaratishga imkon beradi. Dastur grafikani ob'ekt (forma yoki Image komponentasi) yuzasiga chiqaradi. Ob'ekt yuzasiga **Canvas** xossasi mos keladi. Ob'ekt yuzasiga grafik element (to'g'ri chiziq, aylana, turtburchak va hokazo), chiqarish uchun bu ob'ektning **Canvas** xossasiga mos usul qo'llash lozim. Misol uchun Form1.canvas. Rectangle (10,10,100,100) instruktsiyasi dastur oynasida to'rtburchak chizadi.

Chizish sohasi

Yuqorida ko'rilgan **Canvas** xossasi –Tcanvas tipidagi ob'ektdir. Grafik primitivlarni chiqarish usullari Canvas xossasini abstrakt chizish sohasi deb qaraydi. Chizish sohasi alohida nuqtalar – piksellardan iborat. Pikel holati uning gorizontal (X) va vertikal (Y) koordinatalari bilan aniqlanadi. Chap yuqori pikel koordinatalari (0,0). Koordinatalar yuqoridan pastga va chapdan o'ngga qarab o'sib boradi. Soha o'lchovlarini **image** komponentasining Height I Width xossalari va formaning ClientHeight va Clientwidth xossalari orqali aniqlash mumkin.

Qalam

Qalam geometrik figuralarni chizish uchun ishlatiladi. Chiziq ko'rinishi Tren ob'ektining quyidagi jadvalda ko'rsatilgan xossalari orqali aniqlanadi.

Tren (qalam) xossalari.

4-jadval

Xossa	Ta'rifi
Color	Chiziq rangi
Width	Chiziq qalinligi
Style	Chiziq ko'rinishi
Mode	Akslantirish rejimi

Chiziq qalinligi **width** xossasi orqali piksellarda beriladi. Chiziq turini **style** xossasi belgilaydi. Quyidagi jadvalda chiziq turini belgilovchi nomlangan konstantalar sanab o'tilgan.

Style xossasi qiymatlari.

5-jadval

Константа	Chiziq ko'rinishi
psSolid	Uzluksiz chiziq
psDash	Punktir chiziq, uzun shtixlar
psDot	Punktir chiziq, qisqa shtrixlar
psDashDot	Punktir chiziq, uzun va qisqa chiziqlar ketma-ketligi

psDashDotDot	Punktir chiziq, bitta uzun va ikkita qisqa shtrixlar ketma-ketligi
psClear	Chiziq aks ettirish

Mode xossasi chiziq rangining fon rangiga munosabatini ko'rsatadi. Odatda chiziq rangi Pen.Color xossasi qiymati bilan belgilanadi.

Dasturchi chiziq uchun fon rangiga nisbatan **invers** rang berishi mumkin. Bu holda hatto chiziq va fon rangi bir xil berilgan bo'lsa ham chiziq ajralib turadi.

Quyidagi jadvalda Mode xossasi qiymati sifatida ishlatish mumkin bo'lgan konstantalar berilgan.

Mode xossasi qiymatlari

6-jadval

Konstanta	Chiziq rangi
pmBlack	Qora, Pen. Color xossasi qiymatiga bog'liq emas
pmWhite	Oq, Pen. Color xossasi qiymatiga bog'liq emas
pmCopy	Chiziq rangi Pen. Color xossasi qiymatiga bog'liq
pmNotCopy	Chiziq rangi Pen. Color xossasi qiymatiga invers
pmNot	Chiziq rang sohasining mos nuqtasi rangiga invers

Mo'yqalam

Mo'yqalam (Canvas.Brush) yopiq sohalarni chizish va soha ichini bo'yash uchun mo'ljallangan usullardan foydalaniladi. Mo'yqalam ob'ekt jadvalda ko'rsatilgan ikki xossaga ega.

Tbrush (muyqalam) xossalari

7-jadval

Xossa	Ta'rifi
Color	Yopiq sohani bo'yash rangi
Style	Sohani to'ldirish uslubi

Kontur ichidagi soha bo'yalishi yoki shtrixlanishi mumkin. Sohani to'ldirish usulini belgilovchi konstantalar quyidagi jadvalda berilgan.

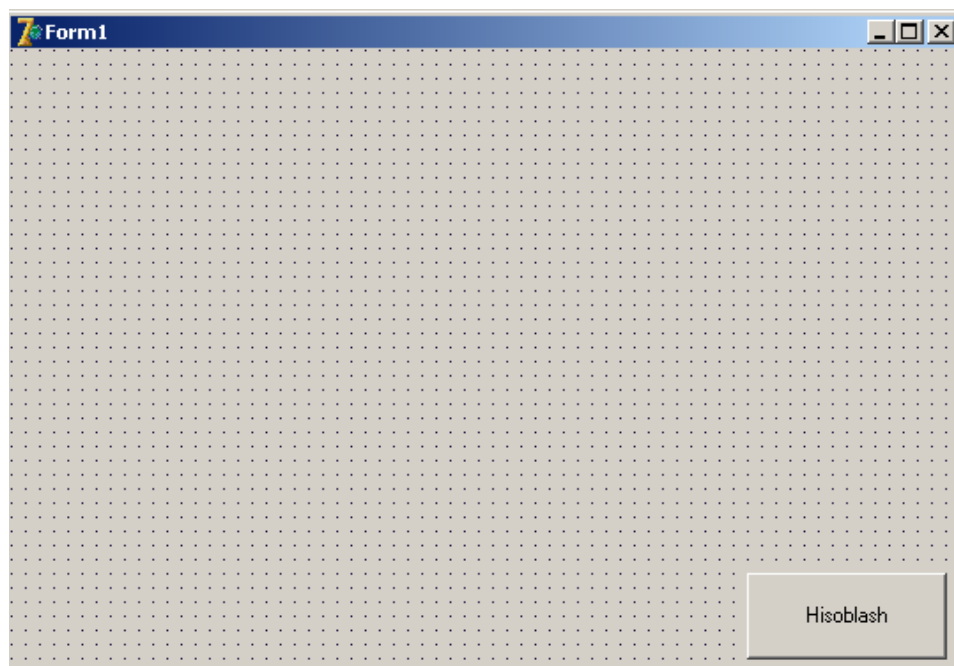
Brush.style xossasi qiymatlari

8-jadval

Konstanta	Soha bo'yash uslubi
bsSolid	Uzluksiz bo'yash
bsClear	Sohani bo'yalmaydi
bsHorizontal	Gorizantal shtrixlash
bsVertical	Vertikal shtrixlash

bsFDiagonal	Diogonal shtirixlash, oldinga og`ish
bsBDiagonal	Diogonal shtirixlash, orqaga og`ish
bsCross	Katakli gorizantal-vertikal shtrexlash
bsDiagCross	Katakli diogonal shtirxlash

11-Misol. Sohalarni bo`yash usullari dasturini tuzing. Buning uchun Formani loyihalashtirishuvchi oynasini ishga tushiramiz va “**Button1**” tugmasini hosil qilib, unga “Hisoblash” matnini kiritamiz



35-rasm. Sohani bo`yash uchun dasturlash oynasi

Hosil bo`lgan oynadan “**Button1**” tugmasining dasturlash maydoniga quyidagi dasturni kiritamiz:

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var

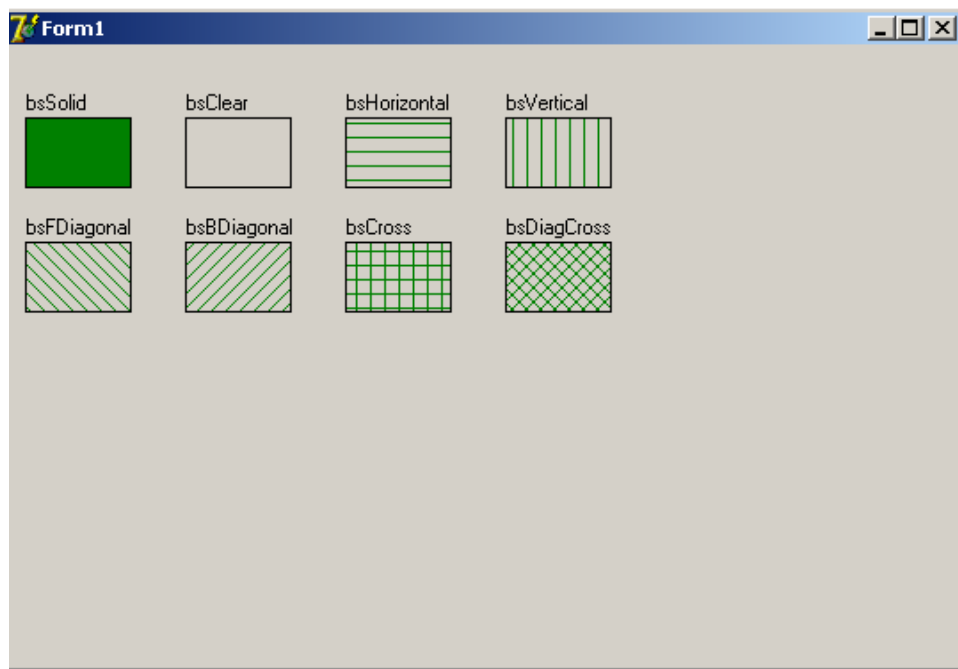
```

```

Form1: TForm1;
implementation
{ $R *.dfm }
procedure TForm1.Button1Click(Sender: TObject);
const
bsName: array[1..8] of string =
('bsSolid','bsClear','bsHorizontal',
'bsVertical','bsFDiagonal','bsBDiagonal',
'bsCross','bsDiagCross');
var
x,y: integer;
w,h: integer;
bs: TBrushStyle;
k: integer;
i,j: integer;
begin
button1.visible:=false;
w:=60; h:=40;
y:=80;
for i:=1 to 2 do
begin
x:=10;
for j:=1 to 4 do
begin
k:=j+(i-1)*4;
case k of
1: bs:= bsSolid;
2: bs:= bsClear;
3: bs:= bsHorizontal;
4: bs:= bsVertical;
5: bs:= bsFDiagonal;
6: bs:= bsBDiagonal;
7: bs:= bsCross;
8: bs:= bsDiagCross; end;
Canvas.Brush.Color := clGreen;
Canvas.Brush.Style := bs;
Canvas . Rectangle (x, y, x+w, y-h) ;
Canvas.Brush.Style := bsClear;
Canvas.TextOut(x, y-55, bsName[k]);
x := x+w+30;
end;
y:= y+h+30;
end;
end;
end.

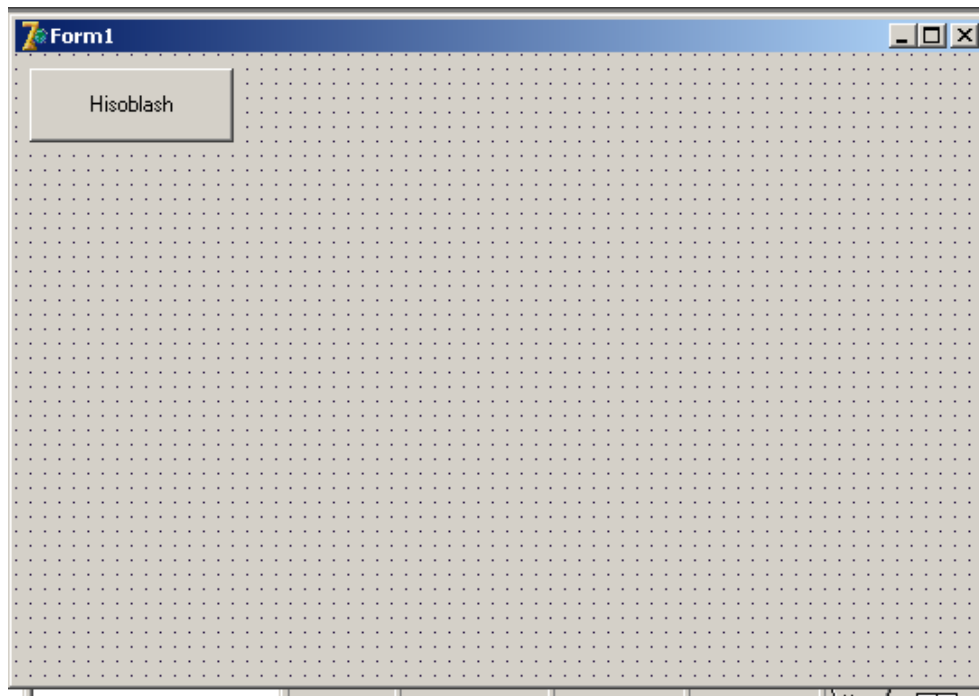
```

Sohani bo'yash usullari dasturi oynasi



36-rasm. Sohalarni bo'yash dasturi

12- m i s o l. $y=x$ funksiyaning grafigini hosil qiling. Formani loyihalashtiruvchi oynasini ishga tushiramiz va **“Button1”** tugmasini hosil qilib, unga **“Hisoblash”** matnini kiritamiz



37-rasm. Funksiya grafigini hosil qilish oynasi

“Button1” tugmasi dasturlash maydoniga quyidagi dasturni kiritamiz:

```
unit Unit1;  
interface  
uses
```

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms, Dialogs, StdCtrls;

type

TForm1 = class(TForm)

Button1: TButton;

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{\$R *.dfm}

Function f(x:real):real;

begin

f:=x;

end;

procedure GrOfFunc;

var

x1,x2:real;

y1,y2:real;

x:real;

y:real;

dx:real;

l,b:integer;

w,h:integer;

mx,my:real;

x0,y0:integer;

begin

l:=10;

b:=Form1.ClientHeight-20;

h:=Form1.ClientHeight-40;

w:=Form1.Width-40;

x1:=0;

x2:=25;

dx:=0.01;

y1:=f(x1);

y2:=f(x1);

x:=x1;

repeat

y := f (x);

if y < y1 then y1:=y;

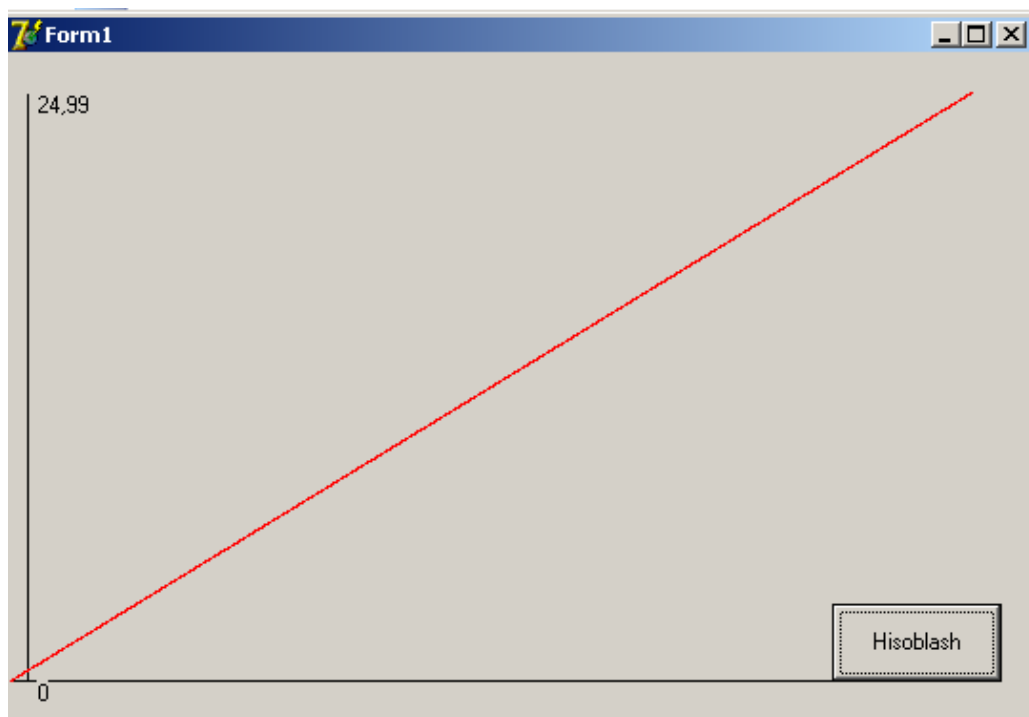
if y > y2 then y2:=y;

```

x:=x+dx; until (x >= x2);
my:=h/abs(y2-y1);
mx:=w/abs(x2-x1);
x0:=1;
y0:=b-Abs(Round(y1*my)) ;
with form1.Canvas do
begin
// ???
MoveTo(l,b);LineTo(l,b-h);
MoveTo(x0,y0);LineTo(x0+w,y0);
TextOut(l+5,b-h,FloatToStrF(y2,ffGeneral,6,3));
TextOut(l+5,b,FloatToStrF(y1,ffGeneral,6,3));
x:=x1; repeat
y:=f(x);
Pixels[x0+Round(x*mx),y0-Round(y*my)]:=clRed;
x:=x+dx;
until (x >= x2);
end;
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
Button1.Visible:=true;
GrOfFunc;
end;
end.

```

Dasturni kiritganimizdan so`ng, oynada quyidagi grafik hosil bo`ladi



38-rasm. Funksiya grafigi

13- m i s o l. $y = \sin^2 x$ funksiya girafigini chizmasini chizing.

Dastur kodi va oynani umumiy ko`rinishini keltiramiz:

unit Unit1;

interface

uses

**Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls;**

type

TForm1 = class(TForm)

Button1: TButton;

procedure Button1Click(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

Form1: TForm1;

implementation

{\$R *.dfm}

Function f(x:real):real;

begin

f:=Sin(x)* Sin(x) ;

end;

procedure GrOfFunc;

var

x1,x2:real;

y1,y2:real;

x:real;

y:real;

dx:real;

l,b:integer;

w,h:integer;

mx,my:real;

x0,y0:integer;

begin

l:=10;

b:=Form1.ClientHeight-20;

h:=Form1.ClientHeight-40;

w:=Form1.Width-40;

x1:=0;

x2:=25;

dx:=0.01;

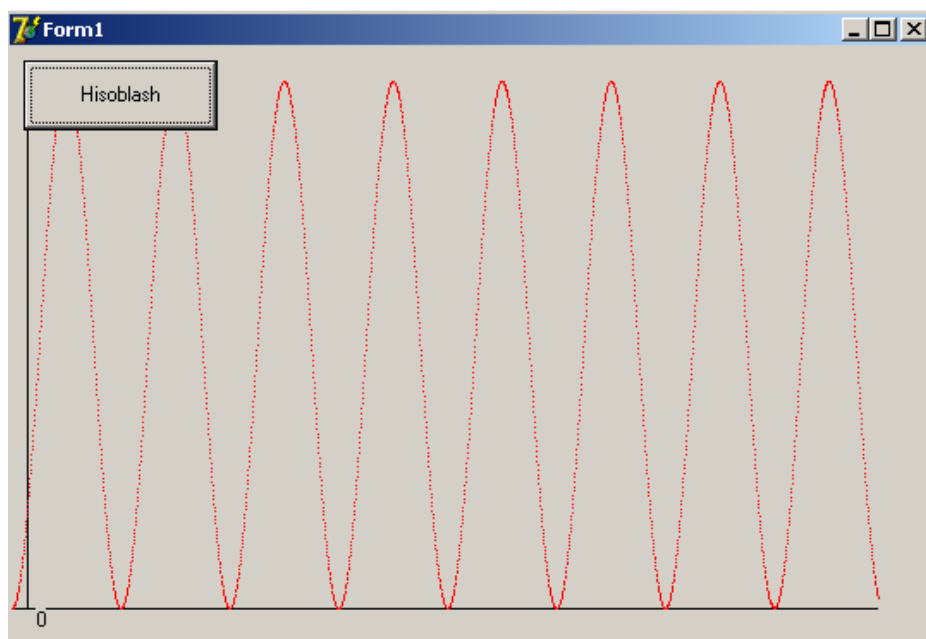
y1:=f(x1);

y2:=f(x1);

```

x:=x1;
repeat
y := f (x);
if y < y1 then y1:=y;
if y > y2 then y2:=y;
x:=x+dx; until (x >= x2);
my:=h/abs(y2-y1);
mx:=w/abs(x2-x1);
x0:=1;
y0:=b-Abs(Round(y1*my)) ;
with form1.Canvas do
begin
// ???
MoveTo(l,b);LineTo(l,b-h);
MoveTo(x0,y0);LineTo(x0+w,y0);
TextOut(l+5,b-h,FloatToStrF(y2,ffGeneral,6,3));
TextOut(l+5,b,FloatToStrF(y1,ffGeneral,6,3));
x:=x1; repeat
y:=f(x);
Pixels[x0+Round(x*mx),y0-Round(y*my)]:=clRed;
x:=x+dx;
until (x >= x2);
end;
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
Button1.Visible:=true;
GrOfFunc;
end; end.

```



49-rasm. Funksiya grafigi

14- m i s o l . . $f = 2 \sin x \cos(x-1)e^{\frac{x}{5}}$ funksiyaning grafigini hosil qiluvchi dastur tuzing.

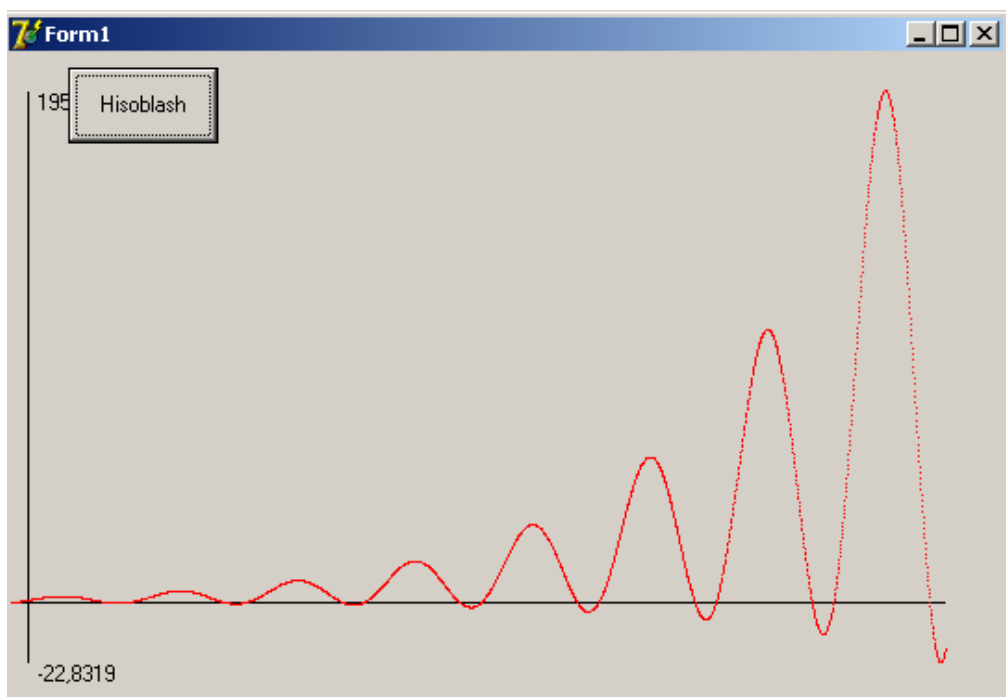
Dastur kodi va oynani umumiy ko`rinishini keltiramiz:

```
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
implementation
{$R *.dfm}
Function f(x:real):real;
begin
  f:=2*Sin(x)*cos(x-1)*exp(x/5) ;
end;
procedure GrOfFunc;
var
  x1,x2:real;
  y1,y2:real;
  x:real;
  y:real;
  dx:real;
  l,b:integer;
  w,h:integer;
  mx,my:real;
  x0,y0:integer;
begin
  l:=10;
  b:=Form1.ClientHeight-20;
  h:=Form1.ClientHeight-40;
  w:=Form1.Width-40;
  x1:=0;
  x2:=25;
```

```

dx:=0.01;
y1:=f(x1);
y2:=f(x1);
x:=x1;
repeat
y := f (x);
if y < y1 then y1:=y;
if y > y2 then y2:=y;
x:=x+dx; until (x >= x2);
my:=h/abs(y2-y1);
mx:=w/abs(x2-x1);
x0:=1;
y0:=b-Abs(Round(y1*my)) ;
with form1.Canvas do
begin
// ???
MoveTo(l,b);LineTo(l,b-h);
MoveTo(x0,y0);LineTo(x0+w,y0);
TextOut(l+5,b-h,FloatToStrF(y2,ffGeneral,6,3));
TextOut(l+5,b,FloatToStrF(y1,ffGeneral,6,3));
x:=x1; repeat
y:=f(x);
Pixels[x0+Round(x*mx),y0-Round(y*my)]:=clRed;
x:=x+dx;
until (x >= x2);
end;
end;
procedure TForm1.Button1Click(Sender: TObject);
begin
Button1.Visible:=true;
GrOfFunc;
end;
end.

```



40-rasm.funksiya grafigi

Mustaqil bajarish uchun mashqlar

Quyidagi funksiyalarning grafigini chizmasini hosil qiling:

1. $y = 2x - 5$

2. $y = \sqrt{x}$

3. $y = \sqrt{x-1}$

4. $\cos x$

5. $\sin x$

6. $\sin^2(x-1)$

7. $\sin 3x$

8. $\cos^3\left(\frac{x-1}{2}\right)$

9. $\cos x \sin x$

10. $f = 2 \sin x \cos(2x-1)e^{\frac{3x}{7}}$

Foydalanilgan adabiyotlar:

1. Sh. A.Nazirov, M.M.Musayev, A.Ne`matov, R.V.Qobulov “Delphi tilida dasturlash asoslari”. Toshkent 2007 yil.
2. Informatika. A. Ahmedov, N. Tayloqov , Toshkent, “O`zbekiston” 2008 -y.
- 3 www.ref.uz – Referatlar , kurs ishilar va diploma ishilar.
4. Поляков Д.В., Круглов И.Ю.Програмирование в среде TURBO PASCAL . – М.Изд-во МАИ, 1992
5. С.А. Немнюгиню. Turbo Pascal, Москва Харьков-Минск 2000
6. Иллюстрированный самоучитель по Delphi 7 для начинающих. Lib.xost.ru ,
Kutubxonasi
7. Н.Б.Культин Delphi 6. программирование на object Pascal

MUNDARIJA:

Kirish.....	3
--------------------	----------

I BOB. DELPHI DASTURLASH TILI VA UNING IMKONIYATLARI

1.1. Delphi dasturlash tili haqida umumiy tushuncha.....	4
1.2. Delphi dasturni ishga tushirish, dasturni piktogrammalari va menyulari haqida tushincha.....	6
1.3.Dasturlash til elementlari.....	13

II BOB. DELPHI DASTURIDA MATEMATIK MASALALARNI YECHISH

2.1. Chiziqli masalalarga oid dasturlar tuzish.....	16
2.2. Tarmoqlanuvchiga oid dasturlar tuzish.....	32
2.3. Takrorlanuvchiga oid dasturlar tuzish.....	49

III BOB. DELPHI DASTURLASH TILIDA VA MASSIVLAR

3.1. Delphi dasturida massivlar bilan ishlash.....	56
--	----

III BOB. DELPHI DASTURLASH TILIDA GRAFIKLAR

3.1. Delphi dasturlash tilining graph moduli ba`zi bir matematik funksiyalarni grafigini yasash.....	69
---	----

Foydalanilgan adabiyotlar	81
--	-----------

