

R.A. RO`ZIYEV

DASTURLASH ASOSLARI

**O‘ZBEKISTON RESPUBLIKASI
OLIV VA O‘RTA MAXSUS TA‘LIM VAZIRLIGI
NAVOIY DAVLAT PEDAGOGIKA INSTITUTI**

R.A.Ro‘ziyev

Dasturlash asoslari

O‘quv qo‘llanma

Navoiy 2020

UDK 681.3.062;004.432

BBK 32.973.26-018.1

Ro'ziev R.A. O'quv qo'llanma. Toshkent: "_____ "nashriyoti, 2020. 158- bet.

Taqrizchilar: Alov R.D.- fizika-matematika fanlari doktori, O'zbekiston Milliy universiteti "Hisoblash modellari va texnologiyalari" kafedrası professori.

Ibragimov A.A.- fizika-matematika fanlari nomzodi, dotsent, Navoiy davlat pedagogika instituti "Informatika o'qitish metodikasi" kafedrası mudiri.

Ma'sul muharrir: O.O.Suvonov

Annotatsiya

O'quv qo'llanma bo'lajak informatika o'qituvchilarini axborotlashgan jamiyatda tayyorlashda dasturlash tilini tadbiiq etishning nazariy ham amaliy asoslari, ya'ni dasturlash texnologiyasi, rivojlanish bosqichlari, algoritm tushunchasi, Delphi dasturlash tili, uning asosiy elementlari va komponentalari, Delphi dasturlash tilida multimediya ilovalari bilan ishlash, ma'lumotlar bazasini yaratish texnologiyalari to'liq yoritilgan. Barcha keltirilgan ma'lumotlar katta miqdordagi misollar orqali bayon qilingan.

Ushbu qo'llanma 5110700- informatika o'qitish metodikasi bakalavriat ta'lim yo'nalishi davlat ta'lim standarti va o'quv rejasiga mos ishlab chiqilgan. Undan Oliy o'quv yurti, akademik litsey va umumiy o'rta ta'lim maktablarida informatika va axborot texnologiyalari, algoritmish va dasturlash asoslari fanlaridan ma'ruza, amaliy va laboratoriya hamda mustaqil mashg'ulotlarni olib borishda foydalanish mumkin.

So‘z boshi

Ta’limni axborotlashtirish muammosini yechishni yangicha qarashdagi muhimligi, ta’lim mazmunini, bo‘lajak mutaxassislarni tayyorlash sifatini takomillashtirish uchun uning shakl va formalarini qanday o‘zgartirish, insonni shaxs sifatida va malakali rivojlanish usullarini izlash hisoblanadi. Buning uchun bunday yondashishlar, birinchi novbatda, ta’limda yangi standartlar va texnologik invariantlarni yaratishga emas, balki zamonaviy axborot va kommunikatsiya texnologiyalaridan foydalanish asosida ta’limni modernizatsiyalash usullarini ishlashga qaratilishi lozim.

Bu jarayonning ajralmas va o‘ta muhim qismi bo‘lib ta’lim jarayonini kompyuterlashtirilishi hisoblanadi. O‘zbekiston Respublikasining “Ta’lim to‘g‘risida”gi Qonuni va O‘zbekiston Respublikasi Prezidentining “O‘zbekiston Respublikasini yanada rivojlantirish bo‘yicha harakatlar strategiyasi to‘g‘risida”gi 2017-yil 7-fevraldagi PF-4947-sonli Farmoni ta’lim tizimini tubdan isloh qilinishiga asos bo‘ldi. Bu yangi ta’lim tizimida dunyo axborot-ta’lim olamiga kirib borishga, o‘quv-tarbiya jarayonini ham nazariy ham amaliy jihatdan o‘zgartirishga, o‘sib kelayotgan yosh avlodni axborot jamiyatida yangicha bilim olish, yaratuvchanlik ruhida tarbiyalashga keng yo‘l ochib berdi.

Kompyuter texnologiyasi ta’limda, uning bir bo‘lagi emas, balki ta’lim jarayonining samaradoligi yuqori darajada rivojlanishidagi bir butun ajralmas qismiga aylandi. Hozirgi kunda hisoblash mashinalari vositalarining tez sur‘atlar bilan rivojlanib borayotganligi, dasturlash tillaridan foydalanish hamda bu boradagi bilimlarini kengaytirishda har bir foydalanuvchida boshlang‘ich ko‘nikmalarning shakllanishiga alohida o‘rin tutmoqda.

Jumladan, Axborot texnologiyalari va tizimlari hamda dasturlash tillaridan foydalanish sohasini rivojlantirish borasida Prezidentimiz hamda Vazirlar Mahkamasi tomonidan shu kungacha bir qancha Farmon, Qaror va Qonunlar qabul qilindi.

Demak, ta’lim olayotgan o‘quvchi-talabalarni davr talabiga javob bera oladigan yetuk mutahassis, komil inson bo‘lib tarbiyalanishlarida, axborotlashtirishning milliy tizimini shakllantirishda, iqtisodiyot va jamiyat hayotining barcha sohalarida zamonaviy axborot texnologiyalarini, kompyuter texnikasi va telekommunikatsiya vositalarini ommaviy ravishda joriy etishda hamda ulardan foydalanishda, fuqarolarning axborotga ortib borayotgan talab-extiyojlarini yanada to‘liqroq qondirishda, jahon axborot hamjamiyatiga kirishda hamda yaratuvchanlik tamoyili asosida yang-yangi g‘oyalarga tayangan holdi dasturlash tillaridan kreativ xususiyatlarini shakllantirish maqsadida “Algoritmash” va “Dasturlash tillari” fanlarini o‘qitish katta ahamiyat kasb etadi.

Ushbu o‘quv qo‘llanmada bayon qilingan ma’lumotlar, 5110700- informatika o‘qitish metodikasi bakalavriat ta’lim yo‘nalishi, davlat ta’lim standarti va o‘quv rejasiga mos keltirilgan mavzular asosida taqdim etilgan. Shuningdek, akademik litsey va umumiy o‘rta ta’lim maktab o‘quvchilari hamda mustaqil dasturlash bilan

shugʻullanuvchilar Delphi dasturlash tili boʻyicha oʻz bilim va koʻnikmalarini oshirishlariga qaratilgan.

Qoʻllanma beshta bobdan iborat boʻlib, har bir bobda keltirilgan tushunchalardan foydalanish aniq misollar yordamida yoritilgan hamda bilimlarni mustahkamlash uchun topshiriqlar keltirilgan. Jumladan, birinchi bobda (1-kurs “Algoritm” fan dasturiga mos), dasturlash texnologiyalari va algoritm nazariyasi haqida qisqacha maʼlumotlar berilgan. Keyingi boblarda 2-kursda “Dasturlash asoslari” fan dasturiga mos, yaʼni ikkinchi bobda, Delphi dasturlash tilini ishga tushirish va foydalanish usullari aks ettirilgan; uchinchi bobda, Delphi dasturlash tilida matematik masalalarni yechish (chiziqli, tarmoqlanovchi, takrorlanuvchi) jarayonlarni dasturlash hamda massivlar, prosedura va funksiyalarga doir misol va masalalarning dastur matnlari keltirilgan. Shuningdek, toʻrtinchi va beshinchi boblarda grafik imkoniyatlari, multimedia ilovalari bilan ishlash, maʼlumotlar bazasini yaratish texnologiyalari haqida maʼlumotlar va mavzularga mos zaruriy tushunchalardan foydalanish imkoniyatlari misollar orqali yoritilgan.

Kitobxon Delphi dasturlash tilida barqaror dasturlash koʻnikmasiga ega boʻlishi uchun qoʻllanmada keltirilgan mashqlarni hech boʻlmaganda bir qismini mustaqil bajarishini tavsiya etamiz.

Oʻquv qoʻllanma baʼzi kamchiliklardan xoli boʻlmasligi mumkin. Uni yanada mukammal qilish boʻyicha bildirilgan fikr-mulohazalarni mamnuniyat bilan qabul qilamiz.

I -BOB. DASTURLASH TEXNOLOGIYALARI

1.1. Dasturni qayta ishlash bosqichlari

Dasturlash – bu dastur yaratish jarayoni bo‘lib, uni yaratishda quyidagi bosqichlardan o‘tiladi:

- qo‘yilgan masalani algoritmlash mumkinligini tekshirish;
- qo‘yilgan masalaning algoritmini tanlash yoki qayta ishlash;
- buyruqlarni yozish;
- dasturni kompilyatsiya qilish, ya‘ni ishlashini tekshirish va xatosini to‘g‘rilash;
- testdan o‘tkazish.

Qo‘yilgan masalani algoritmlash mumkinligini tekshirish – bu bosqich muhim bosqichlardan biri bo‘lib, masalaning qo‘yilishi tahlil qilinadi va ma‘lum bir standart shaklga keltiriladi.

Masalan, $ax^2+bx+c=0$ kvadrat tenglamani yechish uchun algoritmlash jarayoni quyidagi shaklda formallashtiriladi:

- yechimni olish uchun a, b, c kiruvchi berilganlar korteji boshlang‘ich shart hisoblanadi;
- berilganlar albatta dialogli holatda, klaviaturadan dastur ishlayotgan vaqtda kiritilishi maqsadga muvofiq;
- chiquvchi ma‘lumotlar diskriminantning qiymatiga bog‘liq, ya‘ni u manfiymas bo‘lishi lozim;
- agarda diskriminant manfiy bo‘lib qolsa, bu haqda ogohlantiruvchi chiquvchi jumla bo‘lishi kerak.

Qo‘yilgan masalaning algoritmini tanlash yoki qayta ishlash - bu bosqichda natija olish uchun kerak bo‘ladigan muhit tekshiriladi. Agarda masala turli usullar bilan yechiladigan bo‘lsa, dasturchi eng samarali, ya‘ni tez va aniq ishlaydigan usulni tanlaydi. So‘ngra, bu usul masalaga moslashtirilib qayta ishlanadi.

Buyruqlarni yozish – dasturga qo‘yilgan talablar tekshirilganidan va algoritm tuzilgandan so‘ng, u tanlangan dasturlash tillaridan birida yoziladi.

Dastur ishlashini tekshirish – bu bosqichda yaratilgan dastur ichidagi xatoliklar izlanadi. Dasturdagi xatoliklar ikki qismga bo‘linadi: **sintaktik** (matn ichidagi xatoliklar) va **algoritmik**. Sintaktik xatoliklarni (belgilarning almashganlik, tushirib qoldirilganligi va hokazolar) oson topiladi. Algoritmik xatoliklarni topish mushkulroq kechadi. Ma‘lumotlarni kiritish bir-ikki bor takrorlanganda dastur to‘g‘ri ishlasa, xatoliklarni tekshirish bo‘limi yakunlangan hisoblanadi.

Testdan o‘tkazish – bu bosqich o‘ta muhim bo‘lib, yaratilgan dasturdan boshqalar ham foydalanishi hisobga olinadi. Ushbu bosqichda eng ko‘p qancha ma‘lumotni ko‘tara olishi va unda kiritilishi mumkin bo‘lgan noto‘g‘ri ma‘lumotlar tekshiriladi.

Dasturlash tillari haqida umumiy ma‘lumot

Til	Yaratilgan yili	Mualliflar	Tashkilot, firma
Ada	1979-80	Jean Ichbian	Cii-Honeywell (Fransiya)
ARL	1961-1962	Kenneth Iverson, Adin Falkoff	IBM
DELPHI	1995	Chak (Chuck) va Denni (Danny)	Borland
BASIC	1964-1965	JohnKemeny, Thomas Kurtz	Dartmouth Colleje
C	1972-1973	Dennis Ritchie	Bell Laboratories
C++	1980	Bjarne Strostrup	Bell Laboratories
FORTRAN	1950-1958	John Backus	IBM
HTML	1989	Tim Berners-Li	CERN, Jeneva
LOGO	1968-70	Seymour Papert	MassachusetS Institute of Techn.
Pascal	1967-1971	Niklaus Wirth	Federal Institute of Technology (Shveysariya)
SIMULA	1967	Ole-Yoxan Dal, Kristen Nigaard	Norvegiya XM
Java	1995	Djeyms Gosling	Sun Microsystems

1.2. Algoritm tushunchasi

Yuqorida qayd qilganimizdek, qo'yilgan biror masalani EHMda yechish uchun, dastlab uning matematik modeli va algoritmi yaratilib so'ngra dasturini tuzish kerak bo'ladi. Bu uchlikda algoritm bloki muhim ahamiyatga ega. Endi algoritm tushunchasining ta'rifi va xossalarini bayon qilamiz.

Algoritm, bu qo'yilgan masalani yechish zarur bo'lgan amallar ketma-ketligidir. Masalan, kvadrat tenglamani yechish uchun quyidagi amallar ketma-ketligi zarur bo'ladi:

1. a, b, c - koeffitsientlar berilgan bo'lsin;
2. berilgan a, b, c - koeffitsientlar yordamida diskriminant $D=b^2-4ac$ hisoblanadi;
3. $D>0$ bo'lsa, $x_{1,2} = (-b \pm \sqrt{D})/(2 * a)$;
4. $D<0$ bo'lsa haqiqiy yechimi yo'q.

Misol sifatida yana berilgan a, b, c tomonlari bo'yicha uchburchakning yuzasini Geron formulasi bo'yicha hisoblash masalasini ko'rib o'taylik:

1. a, b, c –uchburchakning tomonlari uzunliklari;

2. $p=(a+b+c)/2$ –perimetrning yarmi hisoblansin;
3. $T=p(p-a)(p-b)(p-c)$ hisoblansin;
4. $S=\sqrt{T}$ hisoblansin.

Yuqoridagi misollardan ko‘rinib turibdiki, algoritmning har bir qadamida bajariladigan amallar tushunarli va aniq tarzda ifodalangan hamda chekli sondagi amallardan keyin aniq natijani olish mumkin.

Bayon etilgan, tushunarlilik, aniqlilik, cheklilik va natijaviylik tushunchalari algoritmning asosiy xossalarini tashkil etadi. Bu tushunchalar keyingi paragraflarda alohida ko‘rib o‘tiladi.

“Algoritm” atamasi IX-asrda yashab ijod etgan buyuk alloma Muhammad Al-Xorazmiy nomi bilan bog‘liq. Algoritm so‘zi Al-Xorazmiy nomini Yevropa olimlari tomonidan buzib talaffuz qilinishidan yuzaga kelgan. Al-Xorazmiy birinchi bo‘lib o‘nlik sanoq sistemasining tamoyillarini va undagi to‘rtta amallarni bajarish qoidalarini asoslab bergan.

1.3. Algoritmning asosiy xossalari

Algoritmning beshta asosiy xossasi bor.

1. **Diskretlilik (Cheklilik).** Bu xossaning mazmuni algoritmlarni doimo chekli qadamlardan iborat qilib bo‘laklash imkoniyati mavjudligida. Ya’ni uni chekli sondagi oddiy ko‘rsatmalar ketma-ketligi shaklida ifodalash mumkin. Agar kuzatilayotgan jarayonni chekli qadamlardan iborat qilib qo‘llay olmasak, uni algoritm deb bo‘lmaydi.

2. **Tushunarlilik.** Biz kundalik hayotimizda berilgan algoritmlar bilan ishlayotgan elektron soatlar, mashinalar, dastgohlar, kompyuterlar, turli avtomatik va mexanik qurilmalarni kuzatamiz.

Ijrochiga tavsiya etilayotgan ko‘rsatmalar, uning uchun tushunarli mazmunda bo‘lishi shart, aks holda ijrochi oddiygina amalni ham bajara olmaydi. Shuningdek, ijrochi har qanday amalni bajara olmasligi ham mumkin.

Har bir ijrochining bajarishi mumkin bo‘lgan ko‘rsatmalar yoki buyruqlar majmuasi mavjud bo‘lib, u ijrochining ko‘rsatmalar tizimi (sistemi) deyiladi. Demak, ijrochi uchun berilayotgan har bir ko‘rsatma ijrochining ko‘rsatmalar tizimiga mansub bo‘lishi lozim.

Ko‘rsatmalarni ijrochining ko‘rsatmalar tizimiga tegishli bo‘ladigan qilib ifodalay bilishimiz muhim ahamiyatga ega. Masalan, quyi sinfning a’lochi o‘quvchisi "son kvadratga oshirilsin" degan ko‘rsatmani tushinmasligi natijasida bajara olmaydi, lekin "son o‘zini-o‘ziga ko‘paytirilsin" shaklidagi ko‘rsatmani bemalol bajaradi, chunki u ko‘rsatma mazmunidan ko‘paytirish amalini bajarish kerakligini anglaydi.

3. **Aniqlik.** Ijrochiga berilayotgan ko‘rsatmalar aniq mazmunda bo‘lishi zarur. Chunki ko‘rsatmadagi noaniqliklar mo‘ljaldagi maqsadga erishishga olib kelmaydi. Odam uchun tushunarli bo‘lgan "3-4 marta silkitilsin", "5-10 daqiqa qizdirilsin", "1-2 qoshiq solinsin", "tenglamalardan biri yechilsin" kabi noaniq ko‘rsatmalar robot yoki kompyuterni qiyin ahvolga solib qo‘yadi.

Bundan tashqari, ko‘rsatmalarning qaysi ketma-ketlikda bajarilishi ham muhim ahamiyatga ega. Demak, ko‘rsatmalar aniq berilishi va faqat algoritmda ko‘rsatilgan tartibda bajarilishi shart.

4. **Ommaviylik.** Har bir algoritm mazmuniga ko‘ra bir turdagi masalalarning barchasi uchun ham o‘rinli bo‘lishi kerak. Ya‘ni masaladagi boshlang‘ich ma‘lumotlar qanday bo‘lishidan qat‘iy nazar algoritm shu xildagi har qanday masalani yechishga yaroqli bo‘lishi kerak. Masalan, ikki oddiy kasrning umumiy mahrajini topish algoritmi, kasrlarni turlicha o‘zgartirib bersangiz ham ularning umumiy maxrajlarini aniqlab beraveradi. Yoki uchburchakning yuzini topish algoritmi, uchburchakning qanday bo‘lishidan qat‘iy nazar, uning yuzini hisoblab beraveradi.

5. **Natijaviylik.** Har bir algoritm chekli sondagi qadamlardan so‘ng albatta natija berishi shart. Bajariladigan amallar soni ko‘p bo‘lsa ham baribir natijaga olib kelishi shart. Chekli qadamdan so‘ng qo‘yilgan masala yechimga ega emasligini aniqlash ham natija hisoblanadi. Agar ko‘rilayotgan jarayon cheksiz davom etib natija bermasa, uni algoritm deb bo‘lmaydi.

1.4. Algoritmni tasvirlash usullari

Yuqorida ko‘rilgan misollarda odatda biz masalani yechish algoritmini so‘zlar va matematik formulalar orqali ifodaladik. Lekin algoritm boshqa ko‘rinishlarda ham berilishi mumkin. Biz endi algoritmning eng ko‘p uchraydigan turlari bilan tanishamiz.

1. **Algoritmning so‘zlar orqali ifodalanishi.** Bu usulda ijrochi uchun beriladigan har bir ko‘rsatma jumlar, so‘zlar orqali buyruq shaklida beriladi.

2. **Algoritmning formulalar bilan berilishi.** Bu usuldan matematika, fizika, kimyo kabi aniq fanlardagi formulalarni o‘rganishda foydalaniladi. Bu usul ba‘zan analitik ifodalash deyiladi.

3. **Algoritmning grafik shaklida tasvirlanishi.** Algoritm maxsus geometrik figuralar yordamida tasvirlanadi va bu grafik ko‘rinish blok-sxema deyiladi.

4. **Algoritmning jadval ko‘rinishda berilishi.** Algoritmning bu tarzda tasvirlanishidan ham ko‘p foydalanamiz. Masalan, maktabda qo‘llanilib kelinayotgan to‘rt xonali matematik jadvallar yoki turli xil lotareya jadvallari. Funktsiyalarning grafiklarini chizishda ham algoritmning qiymatlar jadvali ko‘rinishlaridan foydalanamiz. Bu kabi jadvallardan foydalanish algoritmning sodda bo‘lganligi tufayli ularni o‘zlashtirib olish oson kechadi.

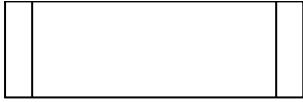
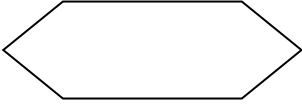
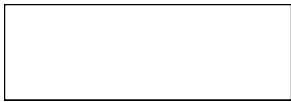
Yuqorida ko‘rilgan algoritmning tasvirlash usullarini asosiy maqsadi, qo‘yilgan masalani yechish uchun zarur bo‘lgan amallar ketma-ketligining eng qulay holatini aniqlash va shu bilan foydalanuvchi tomonidan dastur yozishni yanada osonlashtirishdan iborat. Aslida dastur ham algoritmning boshqa bir ko‘rinishi bo‘lib, u insonning kompyuter bilan muloqotini qulayroq amalga oshirish uchun mo‘ljallangan.

Blok-sxemalarni tuzishda foydalaniladigan asosiy sodda geometrik figuralar quyidagilardan iborat.



Oval (ellips shakli), u algoritmning boshlanishi yoki tugallanishini belgilaydi.

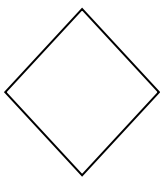
To‘g‘ri burchakli to‘rtburchak, qiymat berish yoki tegishli ko‘rsatmalarni bajarish jarayonini belgilaydi.



Parallelogramm, ma'lumotlarni kiritish yoki chiqarishni belgilaydi.

Oltiburchak, takrorlash operatorini ifodalashda ishlatiladi.

Yordamchi algoritmgga murojatni belgilaydi.



Romb, shart tekshirishni belgilaydi va shart bajarilsa "ha", tarmoq bo'yicha, aks holda "yo'q"-tarmog'i bo'yicha amallar bajarilishini ta'minlaydi.

Strelka - amallar ketma ketligining bajarilish yo'nalishini ko'rsatadi.

Blok-sxemalar bilan ishlashni yaxshilab o'zlashtirib olish zarur, chunki bu usul algoritmlarni ifodalashning qulay vositalaridan biri bo'lib dastur tuzishni osonlashtiradi, dasturlash qobiliyatini mustahkamlaydi. Algoritmik tillarda blok - sxemaning asosiy strukturalariga maxsus operatorlar mos keladi.

Bu o'rinda shuni aytish lozim, yani blok-sxemalardagi yozuvlar odatdagi yozuvlardan katta farq qilmaydi.

1.5. Chiziqli algoritmlar

Har qanday murakkab algoritmnı ham uchta asosiy struktura yordamida tasvirlash mumkin. Bular ketma-ketlik, tarmoqlash va takrorlash strukturalaridir. Bu strukturalar asosida chiziqli, tarmoqlanuvchi va takrorlanuvchi hisoblash jarayonlarining algoritmlarini tuzish mumkin. Algoritmlarni shartli ravishda quyidagi turlarga ajratish mumkin:

- chiziqli algoritmlar;
- tarmoqlanuvchi algoritmlar;
- takrorlanuvchi yoki siklik algoritmlar;
- ichma-ich joylashgan siklik algoritmlar;
- rekurrent algoritmlar;
- takrorlanishlar soni oldindan no'ma'lum bo'lgan algoritmlar;
- ketma-ket yaqinlashuvchi algoritmlar.

Faqat ketma-ket bajariladigan amallardan tashkil topgan algoritmlarga-chiziqli algoritmlar deyiladi. Bunday algoritmnı ifodalash uchun ketma-ketlik strukturasi ishlatiladi. Strukturada bajariladigan amal mos keluvchi shakl bilan ko'rsatiladi. Misol

tariqasida ikki sonning yig`indisi algoritmi va blok-sxemasi umumiy ko`rinishini ko`rsatib o`tamiz:

Algoritm

1-usul:

alg ikki sonning yig`indisi

arg haq a,b

natija haq s

boshl

kiritish a,b

o`zlashtirish $s:=a+b$

chiqarish s

tam

2-usul:

1-qadam. algoritm boshlandi

2-qadam. a, b, s haqiqiy qiymat qabul qiluvchi o`zgaruvchilar

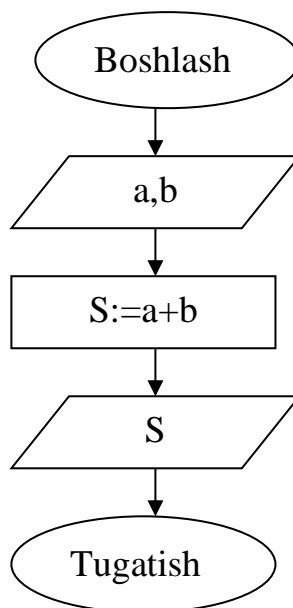
3-qadam. a va b ga qiymat berilsin

4-qadam. $s:=a+b$ hisoblansin

5-qadam. s natija chop etilsin

6-qadam. algoritm tugadi

Blok-sxema



1.1-chizma.

1.6. Tarmoqlanuvchi algoritmlar

Agar hisoblash jarayoni biror bir berilgan shartning bajarilishiga qarab turli tarmoqlar bo'yicha davom ettirilsa va hisoblash jarayonida har bir tarmoq faqat bir marta bajarilsa, bunday hisoblash jarayonlariga tarmoqlanuvchi algoritmlar deyiladi. Tarmoqlanuvchi algoritmlar uchun ayri strukturasi ishlatiladi. Tarmoqlanuvchi strukturasi berilgan shartning bajarilishiga qarab ko'rsatilgan tarmoqdan faqat bittasining bajarilishini ta'minlaydi.

Misol sifatida berilgan ikki sondan kattasini topuvchi algoritmini ko'rsatib o'tamiz:

Algoritm

1-usul:

alg ikki sonning kattasi

arg haq a,b

natija haq max

boshl

kiritish a,b

agar $a > b$

u holda $\max := a$ **aks holda** $\max := b$ **hal bo'ldi**

chiqarish max

tam

2-usul:

1-qadam. algortm boshlandi

2-qadam. a, b, max haqiqiy qiymat qabul qiluvchi o'zgaruvchilar

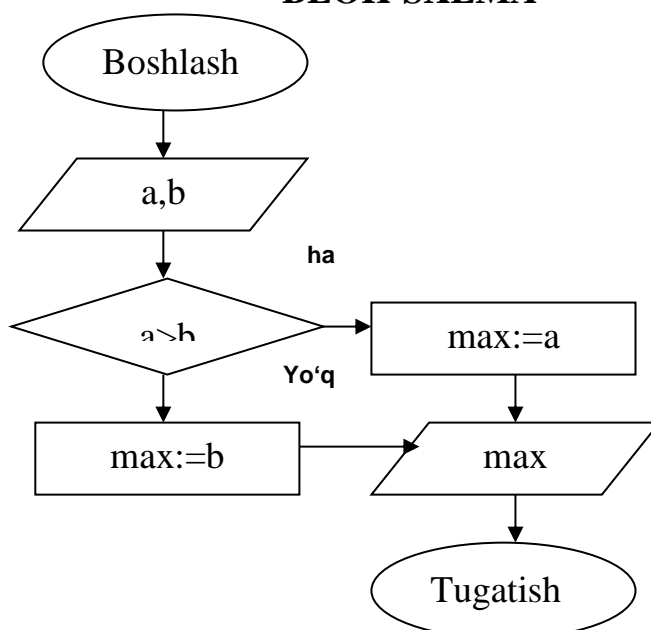
3-qadam. a va b ga qiymat berilsin

4-qadam. agar $a > b$ bo'lsa, u holda $\max := a$ aks holda $\max := b$

5-qadam. max natija chop etilsin

6-qadam. algortm tugadi

BLOK-SXEMA



1.2-chizma.

Berilgan shart romb orqali ifodalanadi, **P**-berilgan shart. Agar shart bajarilsa, "ha" tarmoq bo'yicha **A** amal, shart bajarilmasa "yo'q" tarmoq bo'yicha **B** amal bajariladi.

1.7. Takrorlanuvchi algoritmlar

Agar biror masalani yechish uchun tuzilgan, zaruriy amallar ketma-ketligining ma'lum bir qismi biror parametrغا bog'liq holda ko'p marta qayta bajarilsa, bunday algoritmlar takrorlanuvchi algoritmlar yoki siklik algoritmlar deyiladi. Takrorlanuvchi algoritmlarga tipik misol sifatida odatda qatorlarning yig'indisi yoki ko'patmasini hisoblash jarayonlarini qarash mumkin. Quyidagi ko'paytmani hisoblash algoritmi va blok-sxemasini tuzaylik.

2-misol. $s=1-2+3-4+5- \dots$ qator yig'ingisini berilgan hadlar uchun hisoblang.

Algoritm

1-usul:

alg qator yig'indisi

arg but n

natija but s

boshl but i

kiritish n

agar $n \leq 0$

u holda chiqarish "n ga qayt qiymat berilsin"

aks holda

s:=0

sb i uchun 1 dan n gacha

agar i – toq

u holda s:=s+i

aks holda s:=s-i

hal bo'ldi

so

chiqarish s

tam

2-usul:

1-qadam. algoritmlar boshlandi

2-qadam. i, s, n butun qiymat qabul qiluvchi o'zgaruvchilar

3-qadam. n qiymat berilsin

4-qadam. agar $n \leq 0$ bo'lsa, u holda chop etilsin "n ga qayt qiymat berilsin" va 3-qadamga o'tilsin aks holda 5-qadamga o'tilsin.

5-qadam. s:=0 va i:=1 deb olinsin

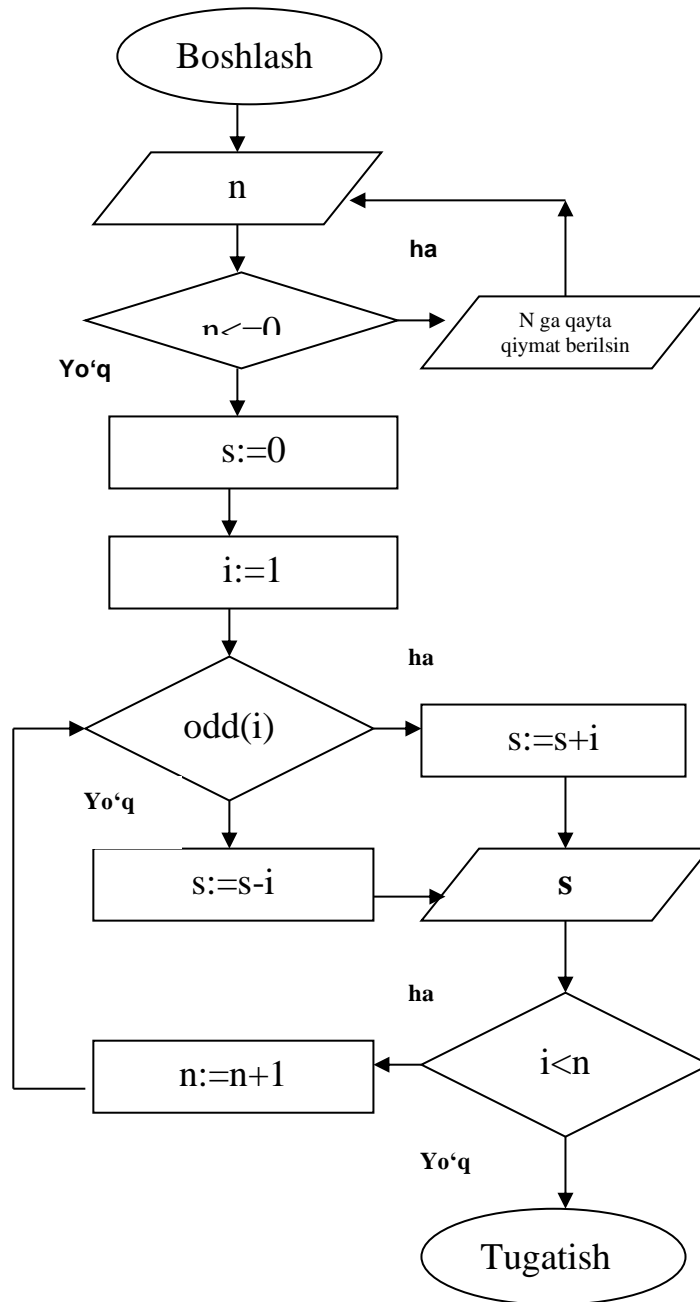
6-qadam. agar i – toq bo'lsa, u holda s:=s+i aks holda s:=s-i

7-qadam. agar $i < n$ bo'lsa, u holda i:=i+1 deb olinsin va 6-qadam o'tilsin aks holda 8-qadamga o'tilsin.

8-qadam. s natija chop etilsin

9-qadam. algoritmlar tugadi

Blok-sxema



1.3-chizma.

1.8. Massivlarni algoritmlash

Bitta indeks qatnashsa, bu massiv bir o'lchovli, ikkita indeks bo'lsa, ikki o'lchovli, n ta indeks bo'lsa, n- o'lchovli massivlar deyiladi. Bir o'lchovli massivlar vektorlar, ikki o'lchovli massivlar matritsalarini ifodalashda ishlatiladi. Vektorlarga va matritsalariga doir masalalarni algoritm va blok-sxemasini tuzaylik.

1-misol. A[N] vektorning eng kichik va eng katta elementlari hamda ular joylashgan tartib raqamlar aniqlansin.

Algoritm

1-usul:

```
alg min_max
  arg but n; haq jad A [1..n]
  natija but nmin, nmax;
  haq amin, amax
boshl but i
  kiritish n
  sb i uchun 1 dan n gacha
    kiritish A[i]
  so
  amin:=A[1]; nmin:=1;
  amax:=A[1]; nmax:=1;
  sb i uchun 2 dan n gacha
    agar A[i] < amin
      u holda amin:=A[i]; nmin:=i
    hal bo'ldi
    agar A[i] >= amax
      u holda amax:=A[i]; nmax:=i
    hal bo'ldi
  so
  chiqarish nmin, amin, nmax, amax
```

tam

2-usul:

1-qadam. algortm boshlandi

2-qadam. n, nmin, nmax, i butun qiymat qabul qiluvchi o'zgaruvchilar, amin, amax haqiqiy qiymat qabul qiluvchi o'zgaruvchilar, a[1..n] haqiqiy qiymat qabul qiluvchi jadval

3-qadam. n ga qiymat berilsin

4-qadam. a[i] massiv elementlari kiritilsin

5-qadam. amin:=a[1]; nmin:=1; amax:=a[1]; nmax:=1; deb olinsin

6-qadam. i:=2 deb olinsin

7-qadam. agar a[i] < amin bo'lsa, u holda amin:=a[i]; nmin:=i deb olinsin

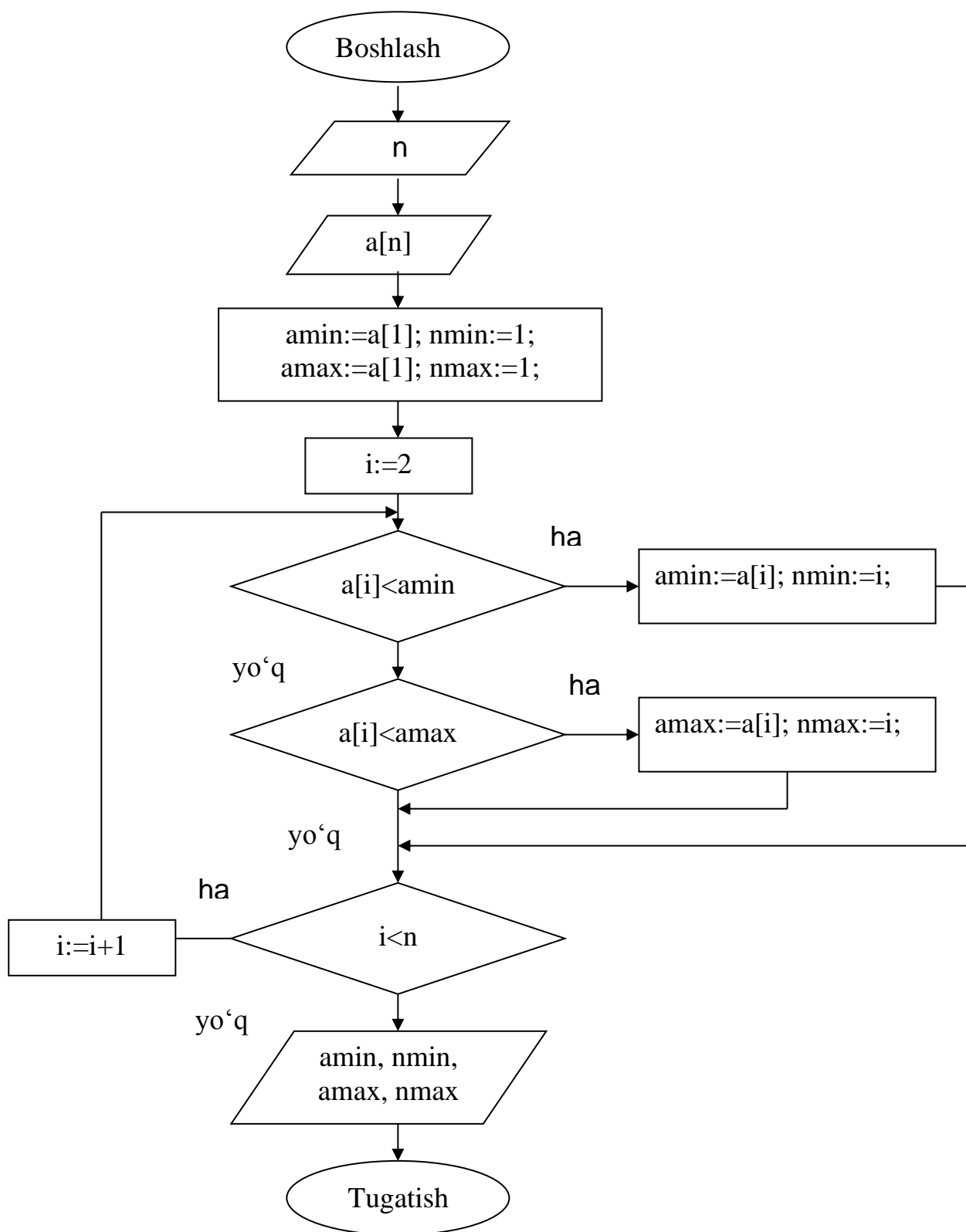
8-qadam. agar a[i]>= amax bo'lsa, u holda amax:=a[i]; nmax:=i deb olinsin

9-qadam. agar i<n bo'lsa, u holda i:=i+1 deb olinsin va 7-qadam o'tilsin aks holda 10-qadamga o'tilsin.

10-qadam. amin, nmin, amax, nmax natijalar chop etilsin

11-qadam. algoritm tugadi

Blok-sxema



1.4-chizma.

2-misol. A[3,4] matritsaning ustunlaridagi eng katta elementlardan B vektorni hosil qilish.

Algoritm

1-usul:

```
alg max_ustun
  arg haq jad a[1..3,1..4]
  natija haq jad b[1..4]
boshl but i,j
  sb i uchun 1 dan 3 gacha
    sb j uchun 1 dan 4 gacha
      kiritish a[i,j]
    so
  so
  sb j uchun 1 dan 4 gacha
    b[j]:=a[1,j]
    sb i uchun 2 dan 3 gacha
      agar b[j]<a[i,j]
        u holda b[j]:=a[i,j]
      hal bo'ldi
    so
  chiqarish b[j]
so
tam
```

2-usul:

1-qadam. algoritm boshlandi

2-qadam. i, j butun qiymat qabul qiluvchi o'zgaruvchilar, a[1..3,1..4], b[1..4] haqiqiy qiymat qabul qiluvchi jadvallar

3-qadam. a[i,j] matritsa elementlari kiritilsin

4-qadam. j:=1 deb olinsin

5-qadam. b[j] :=a[1,j] deb olinsin

6-qadam. i:=2 deb olinsin

7-qadam. agar b[j]<a[i,j] bo'lsa, b[j]:=a[i,j] deb olinsin

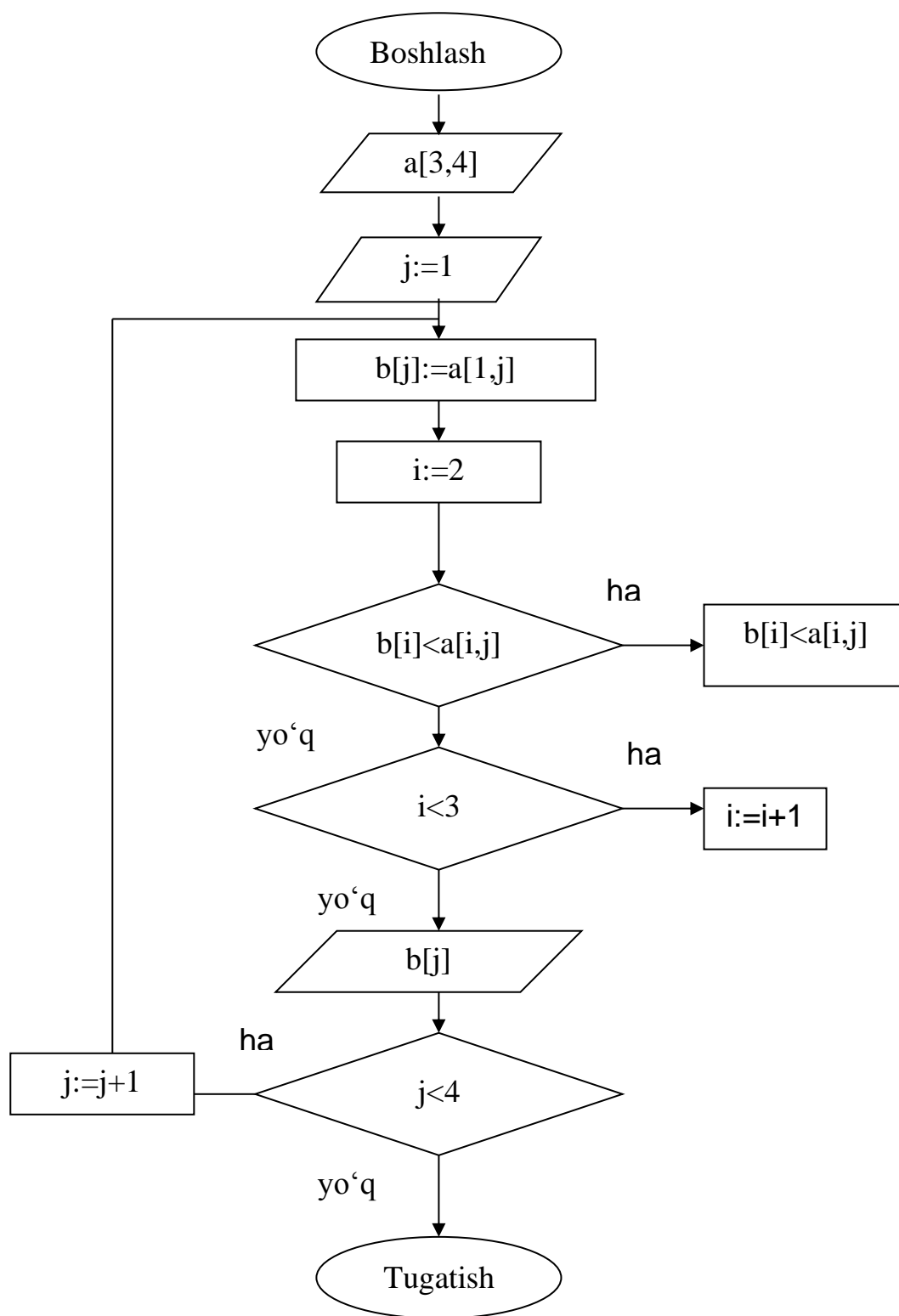
8-qadam. b[j] chop etilsin

9-qadam. agar 2<3 bo'lsa, u holda i:=i+1 deb olinsin va 7-qadam o'tilsin aks holda 10-qadamga o'tilsin.

10-qadam. agar j<4 bo'lsa, u holda j:=j+1 deb olinsin va 5-qadam o'tilsin aks holda 11-qadamga o'tilsin.

11-qadam. algoritm tugadi

Blok-sxema



1.5-chizma.

Mustaqil bajarish uchun topshiriqlar

Quyidag berilgan masalalarning algoritmi va blok sxemasini tuzing.

1. To'rtta son berilgan. Shu sonlarning o'rtta arifmetigi va o'rtta geometrigini hisoblovchi.

$$2. k = \ln \left| (y - \sqrt{|x|}) * \left(x - \frac{y}{z + x^2 / 4} \right) \right|.$$

3. Berilgan uchta sondan eng kattasini topuvchi.

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

$$5. y = \begin{cases} \sqrt[3]{x^2 + a^2}, & \text{agar } x^2 + a^2 < 4, \\ x + a, & \text{agar } x^2 + a^2 = 4, \\ \ln x, & \text{agar } x^2 + a^2 > 4. \end{cases}$$

6. 1 dan 100 gacha bo'lgan sonlarning yig'indisini topuvchi.

$$7. p = \prod_{k=1}^3 \prod_{n=3}^5 \frac{\sqrt{n^k + 1}}{\ln n}$$

8.

$$s = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \dots$$

$$9. y = \sum_{i=1}^5 \frac{ilz_i}{2^i}$$

10. $A(5)$ vektor elementlarini kamayib borish tartibida joylashtirish.

11. $A(M, N)$ matritsa berilgan. Har bir satrdagi eng kichik elementlar orasidan eng kattasini va u joylashgan tartib raqamini aniqlash.

12. $y = \sqrt{x + 2x}$ funksiyaning qiymatini $-2 \leq x \leq 2$ oraliqda $\Delta x = 0,4$ qadam bilan,

$3 \leq x \leq 5$ oraliqda $\Delta x = 0,5$ qadam bilan, $6 \leq x \leq 10$ oraliqda $\Delta x = 1$ qadam bilan hisoblash.

II- BOB. DELPHI DASTURLASH TILI

2.1. Delphi dasturlash tili tarixi

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 Delphining paydo bo`lishi dasturlashni rivojlantirish tarixida yorqin ko`rinish bo`ldi. Delphi dasturlash tilining 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, Delphi 9.0, Delphi 10.0.

Dastur yuqori unumdorlikka va tezlikka ega bo`lgan kompilyatorga ega. Har bir yangi versiyasi yaratilganda dasturlash tiliga qandaydir qulaylik yoki imkoniyatlilik kiritildi. Delphi dasturlash tili vizual instrumentlar yig`indisiga ega bo`lib, u esa dasturlash jarayonini yanada osonlashtiradi. Dasturchi dastur tuzish jarayonida komponentlar palitrasidagi tayyor tugmachalarni (ob'ektlarni) dasturga joylashtirishi mumkin. Maxsulotning standart turi ikkita bazaviy sinflardan tashkil topgan. Dasturlash tili qobig`i korporativ ma'lumotlar bazasiga ulanishni qo`llab quvvatlovchi, dastur tuzishni tezlashtiruvchi RAD (Rapid Application Development) vizual asboblar 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 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 (доступ) 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.

Hozirgi kunda juda ko'plab algoritmik tillar mavjud. Bu tillar ichida Pascal tili universal tillardan biri bo'lib, boshqa tillarga qaraganda imkoniyati ko'pligi va kengroqligi bilan farq qiladi. So'nggi yillarda Pascal tili juda takomillashib, tobora ommaviylashib bormoqda. Pascal tili Shvetsariyalik olim N.Virt tomonidan yaratilib, keyinchalik Borland korporatsiyasi tomonidan rivojlantirildi. Pascal tilida dastur tuzish uchun Turbo Pascal va Delphi dasturlash vositalari mavjud. Bu dasturlash vositalari zamonaviy kompyuter texnologiyasining barcha talablarini o'z ichiga olgan va unda dastur tuzish uchu hamma qulayliklar yaratilgan.

Delphi dasturlash tili bir necha muhim ahamiyatga ega bo'lgan texnologiyalar kombinatsiyasini o'zida mujassam etgan:

- yuqori darajali mashinali kodda tuzilgan kompilyator;
- ob'ektga yo'naltirilgan komponentalar modellari;
- dastur ilovalarini visual tuzish;
- ma'lumotlar bazasini tuzish uchun yuqori masshtabli vosita.

Delphi – Windows muhitida ishlaydigan dastur tuzish uchun qulay bo'lgan vosita bo'lib, kompyuterda dastur yaratish ishlarini avtomatlashtiradi, xatoliklarni kamaytiradi va dastur tuzuvchi mehnatini yengillashtiradi.

Delphida dasturlash tili zamonaviy visual loyihalash texnologiyasi asosida obektga yo'naltirilgan dasturlash nazariyasini hisobga olgan holda tuziladi. Delphi sistemasi Turbo Pascal 7.0. tilining rivoji bo'lgan Object Pascal dasturlash tilini ishlatadi.

2.2. Delphi dasturlash tilida konsol ilovasida ishlash

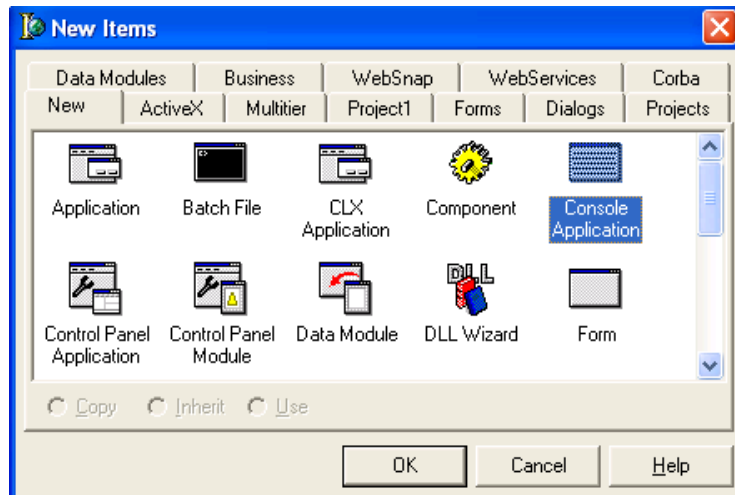
Delphi dasturlash tilida konsol ilovalarini turli usullarda yaratish mumkin. Ulardan eng oson usuli quyidagicha:

1. Delphi muhiti ishga tushiriladi;

“Пуск→Программы→Borland Delphi 7→ Delphi 7”

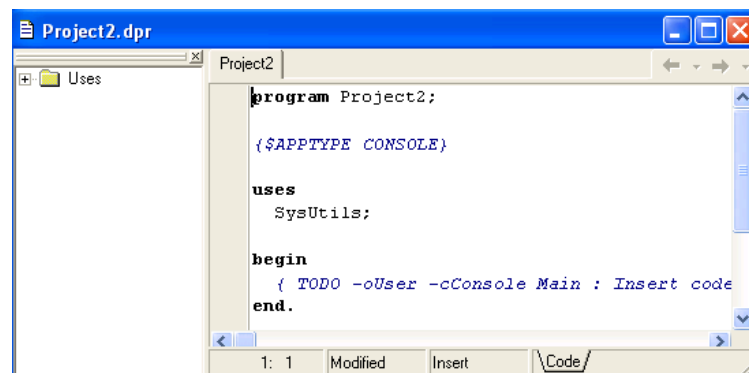
2. Bosh menyudan File punktini ochib, u yerdan New, keyin esa Other buyruqlari tanlanadi; “File→New→Other”

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



2.1-rasm.

1. Natijada ekranda loyiha oynasi ochiladi (.dpr kengaytmali nom bilan).



2.2-rasm.

Begin – end ichiga olingan

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

Izoh o‘rniga loyiha faylining dastur kodi kiritiladi. Misol uchun berilgan ikki sonni yig‘indisini hisoblovchi dasturini ko‘rishimiz mumkin:

```
program Project2;  
{$APPTYPE CONSOLE}  
uses  
SysUtils;  
var  
s,a,b:real;
```

```

begin
write('a='); readln(a);
write('b='); readln(b);
s:=a+b;
write('s=',s:0:0);
readln;
  { TODO -oUser -cConsole Main : Insert code here }
end.

```

```

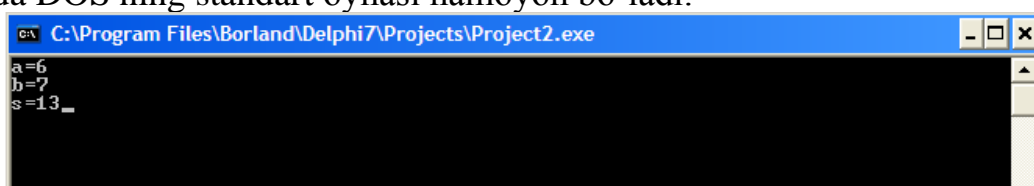
program Project2;

{$APPTYPE CONSOLE}

uses
  SysUtils;
var
  s,a,b:real;
begin
  write('a='); readln(a);
  write('b='); readln(b);
  s:=a+b;
  write('s=',s:0:0);
  readln;
  { TODO -oUser -cConsole Main : Insert code here }
end.

```

Tuzilgan dasturni ishga tushirishdan oldin uni saqlash kerak. Uni saqlash file=>Save All buyrug‘ini berish lozim. Har bir loyihani alohida yangi papkaga saqlash tafsiya qilinadi. Loyiha faylini saqlashda, unga nom berilmasa ProjectN.dpr nomli fayl nomini tafsiya qiladi. Loyihani saqlab bo‘lgandan so‘ng, uni bajarishga beramiz. Buning uchun bosh menyudan quyidagi buyruqni berish lozim: Run=>Run yoki F9 funksional tugmachasini bosish kerak bo‘ladi. Dastur normal ishga tushgandan so‘ng ekranda DOS ning standart oynasi namoyon bo‘ladi.



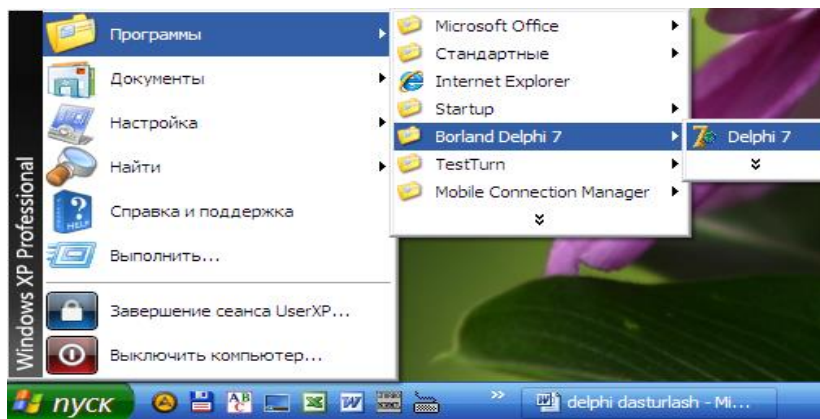
2.3-rasm.

2.3. Delphi dasturlash tili muhiti

Delphi dasturlash tilini ishga tushirishning bir nechta usullari mavjud:

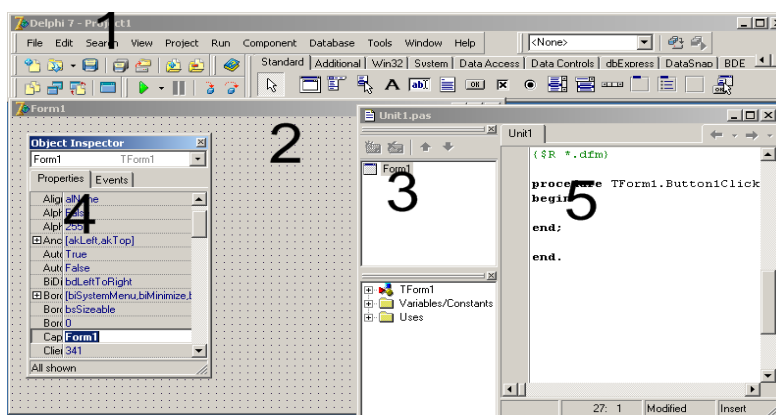
- foydalanuvchining ish stolidagi yorliq yordamida;
- quyi masalalar paneli yordamida;
- bosh menyu yordamida.

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



2.4-rasm.

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



2.5-rasm.

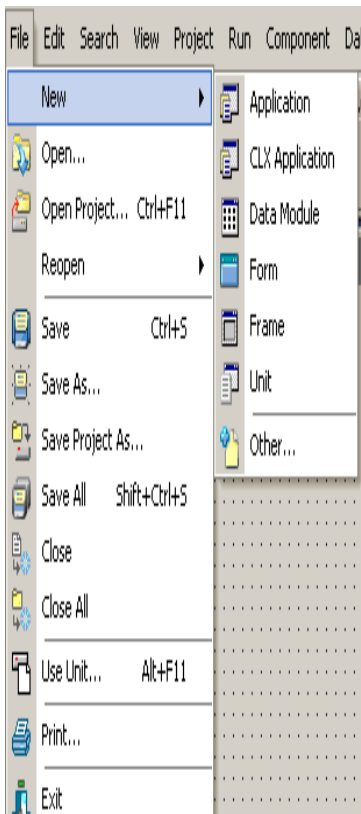
- 1- Menyular satri
- 2- Forma oynasi
- 3- Ob'ektlar daraxti
- 4- **Object Inspektori**
- 5- Dasturlash maydonchasi

Menyular satrining ko‘rinishi odatdagi Office dasturlarini singari sarlavha satridan keyin joylashgan. Delphi dasturining o‘n bitta menyusi bo‘lib ular quyidagilardir:



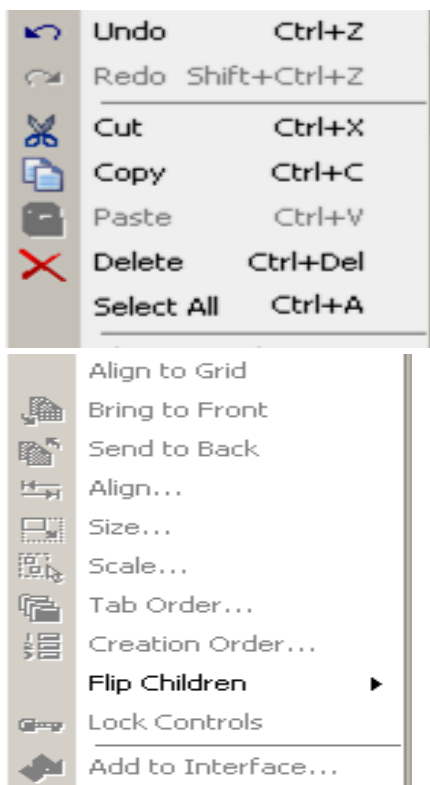
2.6-rasm.

1. File menyusi. File menyuning bandlari quyidagi bo‘limlardan iborat: New- Yangi loyiha yaratish. Agar siz yangi forma yaratmoqchi bo‘lsangiz, fayl(file) menyusining yangi(new) bandiga sichqoncha tugmasini olib borishingiz bilan sizga bir nechta variantni tanlash imkoniyati tug‘iladi:



Application- yangi forma yaratish;
Application CLX- yangi CLX forma yaratish;
Data module- axborotlar moduli;
Form- Yangi forma yaratish;
Frame- Yangi oyna yaratish;
Unit – Yangi modul yaratish;
Open – Mavjud faylni ochish;
Open project– project ko‘rinishidagi faylni ochish;
Reopen – Oldin ochilgan fayllarni qayta ochish;
Save – oldin saqlangan faylni qayta saqlash;
Save As – dasturni kompyuter xotirasiga saqlash;
Save project As- dasturni exe fayl ko‘rinishiga o‘tkazish;
 ▪ **Save All-** barcha dasturlarni saqlash;
 ▪ **Close –** faylni yopish;
Close All- ochilgan fayllarni yopish;
Use Unit – modulni yuklash;
Print- chop etishga berish;
Exit- dasturdan chiqish.

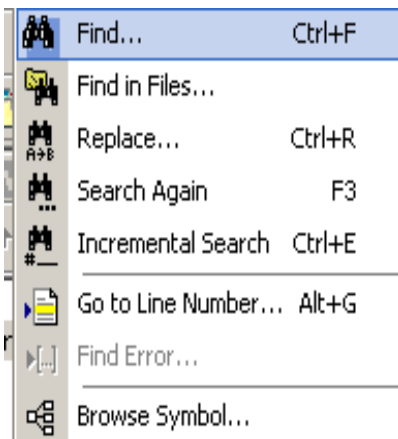
2. Edit menyusi. Edit menyusining bandlari asosan ma’lumotlarni o‘chirish, qayta tiklash, ma’lumotlarni joyini almashtirish va o‘lchamlarini kattalashtirish yoki kichiklashtirishga yordam beradi. Bular quyidagilar:



Undo- oxirgi bajarilgan buyruqni bekor qilish;
Redo- oxirgi bajarilgan buyruqqa qaytish;
Cut- belgilangan ma’lumotni qirqib, kompyuter xotirasiga olish;
 ▪ **Copy-** belgilangan ma’lumotdan nusxa olish ;
Paste- nusxa olingan ma’lumotni joylashtirish;
Delete- belgilangan ma’lumotlarni o‘chirish;
Select All- matnni belgilash.
Align to Grid- To‘r to‘qish;
Bring to Front- oldinga o‘tkazish;
 ▪ **Send to Back-** oxiriga jo‘natish;
 ▪ **Align-** tanlash;
 ▪ **Size-** o‘lcham;
 ▪ **Scale-** ko‘rinish;
 ▪ **Tab Order...-** tartiblash;
 ▪ **Creation Order-** yaratish va tartiblash
 ▪ **Flip Children-** o‘zgartirish;
 ▪ **Lock Controls-** boshqarishni quliflash;
 ▪ **Add to Interface...-** qo‘shimcha imkoniyat.

Yuqorida keltirilgan buyruqlarni klaviatura yordamida ham bajarish mumkin. Bular quyidagilar: bajarilgan buyruqni bekor qilish Ctrl+Z, oxirgi bajarilgan buyruqqa qaytish Shift+Ctrl+Z, belgilangan ma'lumotlarni qirqib olish Ctrl+X, belgilangan ma'lumotdan nusxa olish Ctrl+C, nusxa olingan ma'lumotni matnga joylashtirish Ctrl+V, belgilangan ma'lumotlarni o'chirish Ctrl+Delete, matni belgilash Ctrl+A tugmachalari yordamida bajariladi.

3. Search (qidirish) menyusi. Menyu bandlari matndagi harflarni, so'zlarni, simvollarni hamda kompyuter xotirasidagi fayllarni qidirishimiz va ularni boshqasiga almashtirishda xizmat qiladi. Demak, ular quyidagilar:



Find- fayldagi ma'lumotlarni qidirish;

Find in Files- kompyuter xotirasidagi fayllarni qidirish;

Replace- matndagi biror so'z, harf yoki belgini boshqasiga almashtirish ;

Search Again- izlashni davom ettirish;

Incremental Search- qidirish;

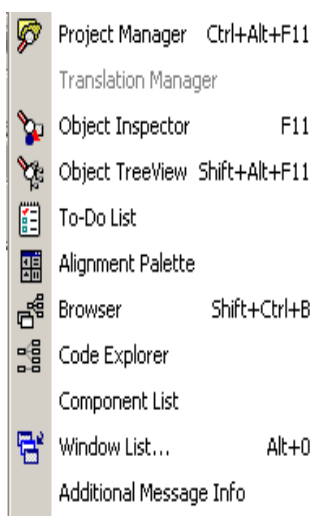
Go to Line Number- nomer satri bo'yicha qidirish;

Find Error – xatoliklarni qidirish;

Browse Symbol...- belgilarni qidirish.

Menyuning buyruqlarini klaviatura yordamida ham bajarish mumkin, ya'ni fayldagi ma'lumotlarni qidirish Ctrl+F, matndagi biror so'z, harf yoki belgini boshqasiga almashtirish Ctrl+R, izlashni davom ettirish F3, nomer satri bo'yicha qidirish Alt+G tugmalar juftligi orqali amalga oshiriladi.

4. View menyusi. View menyusining bandlari orqali dastur oynasi ko'rinishini o'zgartirishda, biror ko'rinishni yo'qotish yoki paydo qilishda foydalaniladi. Bular quyidagilar:



Project Manager- loyiha menijeri;

Translation Manager- tarjima menijeri;

Object Inspector- Object Inspektorini bekor qilish yoki o'rnatish;

Object TreeView- ob'ektlar daraxtini bekor qilish yoki o'rnatish;

To-Do List- ro'yxati;

Alignment Palette- palitraga o'tish ;

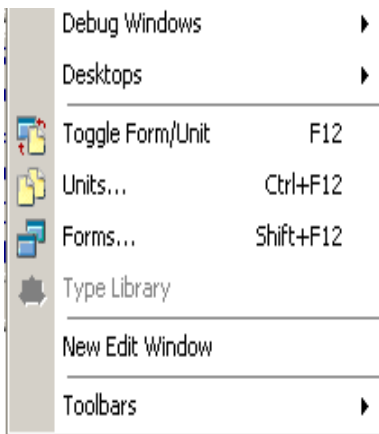
Browser- komponentalarni tartiblash;

Code Explrer- forma oynasini tartiblash;

Windows List...- oynalarning ro'yxatini chiqarish;

Additional Message Info- ma'lumot jo'natish;

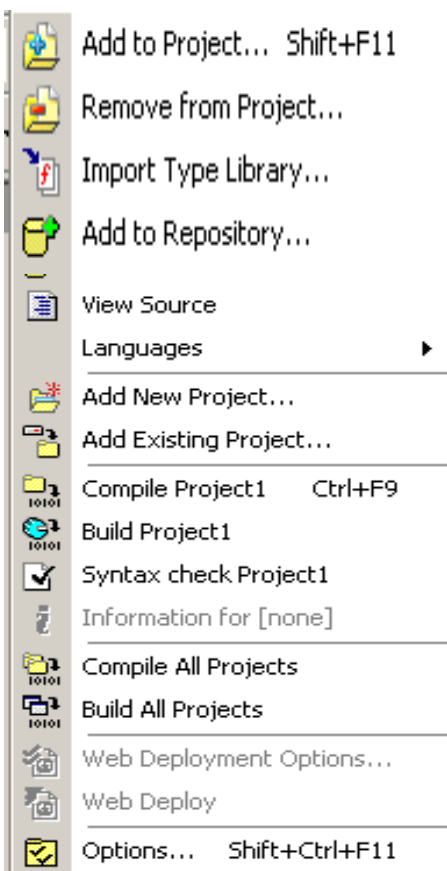
Debug Windows- ishchi oynaga o'tish;



Desktop- oynalarni ochish;
Toggle Form/Unit- dasturlash maydoniga yoki forma oynasiga o'tish;
Units...- tuzilish;
Forms...- forma oynasiga o'tish;
- Type Library- kutubxona turi;
New Edit Windows- yangi dasturlash maydoniga o'tish;
Toolbars- dasturlash oynasidagi yorliqlarni qo'shish yoki yo'qotish.

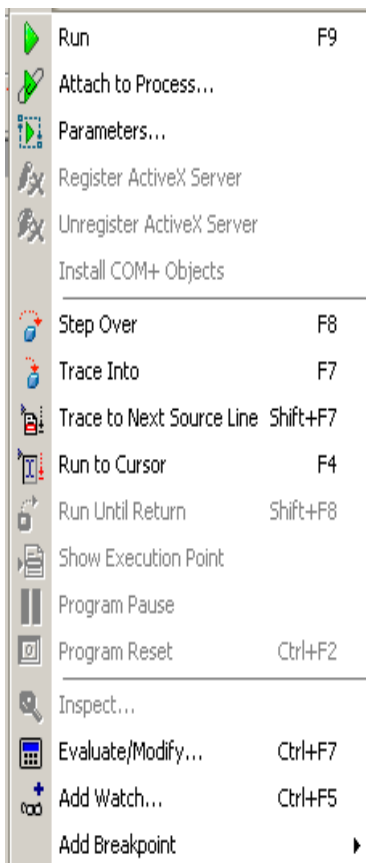
Yuqorida keltirilgan buyruqlarni ba'zi birlarini klaviatura yordamida ham bajarish mumkin. Bular quyidagilar: loyiha menijeri ro'yxatini ko'rish Ctrl+Alt+F11, **Object Inspektorini** yo'qotish yoki o'rnatish F11, ob'ektlar daraxtini yo'qotish yoki o'rnatish Shift+Alt+F11, komponentalarni tartiblash Shift+Ctrl+B, oynalarning ro'yxatini chiqarish Alt+O, dasturlash maydoniga yoki forma oynasiga o'tish F12, dastur tuzilishini ko'rish Ctrl+F12, forma oynasiga o'tish Shift+F12 tugmalar yordamida amalga oshiriladi.

5. Project menyusi. Foydalanuvchi menyuning bu bandlardan loyiha oynasi ko'rinishini o'zgartirishda, biror ko'rinishni yo'qotishda, paydo qilishda, dasturni faollashtirishda foydalanadi. Bular quyidagilar:



Add to project- qo'shimcha loyiha qo'shish;
Remove from Project- qo'shimcha loyihani o'chirish;
Import Type Library- dastur kutubxonasiga murojat;
Add to Repository- saqlab fayllarni ochish;
View Source- dasturlash maydoniga o'tish;
Languages- qo'shimcha tilni o'rnatish;
Add New Project- yangi loyiha qo'shish;
Add Existing Projekt- navbatdagi loyihani qo'shish;
Compile Project1- dasturni ishchi holatga o'tkazish;
Build Project1- komponovka qilish;
Syntax Check Project1- loyihadagi xatoliklarni tekshirish;
Information for [none]- ma'lumot uchun;
Compile All Projects- barcha loyihalarni yig'ish;
Build All Projects- loyihalarni birlashtirish;
Web Deployment Options...- web sahifaga bog'lash;
Web Deploy- web sahifaga o'tish;
Options...- o'zgartirish.

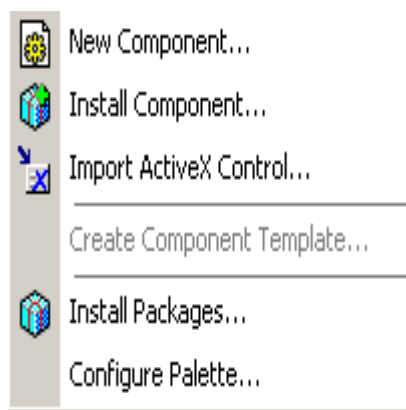
6. Run menyusi.



Run- dasturni faollashtirish;
Attach to Process...- faollashgan dasturlarni aniqlash;
Parameters...-parametr;
Register ActiveX Server- serverni ro'yxatga olish;
Unregister ActiveX Server- ro'yxatga olingan serverni bekor qilish;
Install COM+ Objects- COM+ obektni o'rnatish;
Step Over- ishlashga ruxsat berish;
Trace Into- qator yo'nalishlarni belgilash;
Trace to Next Source Line- navbatdagi qatorga o'tish;
Run to Cursor- tuzilgan dasturni ishlatish;
Run Until Return- faollashgan dasturni qaytarish;
Show Execution Point- ko'rsatilgan jarayonni bajarish;
Program Pause- dasturni to'xtatish;
Program Rest- bir maydondan ikkinchisiga o'tish;
Inspect- tekshirish;
Evaluate/Modify- hisoblashni almashtirish;
Add Watch- tekshirishning qo'shimcha imkoniyatlari;
Add breakpoint- to'xtatishning qo'shimcha imkoniyatlari.

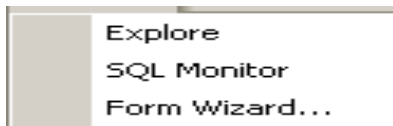
Bu yerda ham yuqorida aytib o'tilgan buyruqlarni ba'zi birlarini klaviatura yordamida bajarish mumkin. Bular quyidagilar: qo'shimcha loyiha qo'shish Shift+F11, dasturni ishchi holatga o'tkazish Ctrl+F9, dastur maydonini o'zgartirish Shift+Ctrl+F11, dasturni faollashtirish F9, ishlashga ruxsat berish F8, qator yo'nalishlarni belgilash F7, tuzilgan dasturni ishlatish F4, faollashgan dasturni qaytarish Shift+F8, bir maydondan ikkinchisiga o'tish Ctrl+F2, hisoblashni almashtirish Ctrl+F7, tekshirishning qo'shimcha imkoniyatlari Ctrl+F5 tugmalar yordamida amalga oshiriladi.

7. Component menyusi. Dasturchi menyuning bu bandlardan yangi loyiha oynasini ustida amallar bajarishda, ma'lumotlar ombori yaratishda, standart komponentalar palitrasini o'zgartirishda foydalanadi. Bular quyidagilar:



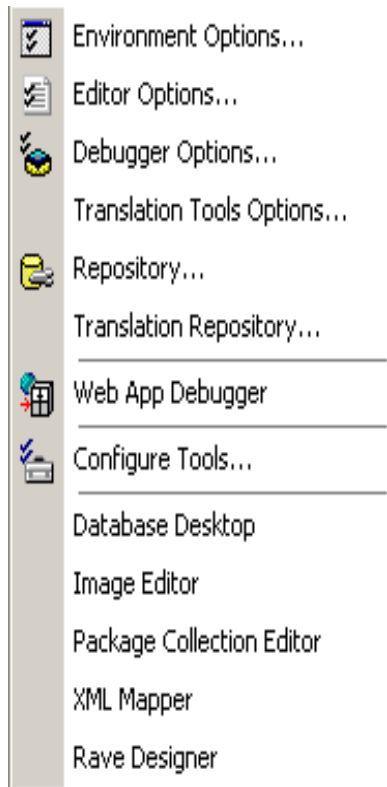
New component- Yangi loyiha yaratish;
-Install component- Loyha o'rnatish;
Import ActiveX Control...- ActiveX dasturini boshqarish uchun olish;
Create Component Template- Shablon komponentalarini yaratish;
Install Packages...- Paketlarni o'rnatish;
Configure palette...- komponentalar palitrasini qo'shish va o'chirish.

8. Database menyusi.



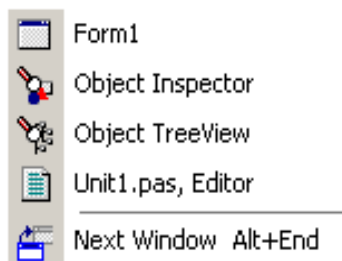
Explore- o‘tkazish;
SQL Monitor- SQL bandiga o‘tish;
Form Wizard...- forma yaratish.

9. Tools menyusi.



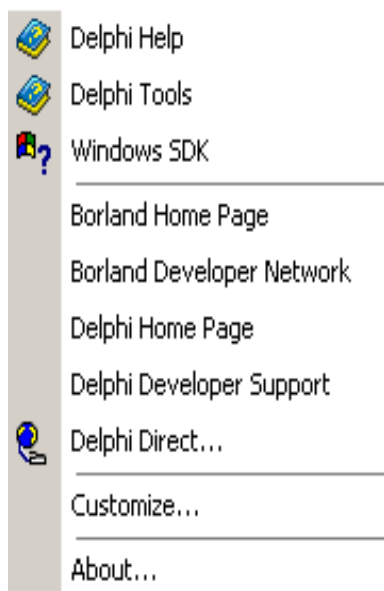
Environment Options...- standart komponentalar palitrasini o‘zgartirish ;
Editor Options...- dasturlash maydonining parametrlarini o‘zgartirish;
Debugger Options...- o‘zgartirish va tanlash;
Translation Tools Options...- o‘lcham va ranglarni tanlash;
Repository...- saqlash;
Translation Repository- tarjimini saqlash;
Web App Debugger- web sahifaga bog‘lash;
Regenerate CORBA IDL Files...- fayllarga bog‘lash;
Configure Tools...- ishchi muhitini sozlash;
Databse desktop- Databse desktop menyusini yuklash;
Image Editor- Image Editor dasturini yuklash;
Package Collection Editor - Package Collection Editor menyusini yuklash;
-XML Mapper- XML menyusini yuklash;
Rave Designer- Rave Designer menyusini yuklash;

10. Window(oyna) menyusi. Menyuning bu bandlarida forma oynasini yuklash, **Object Inspektor**iga oid amallar bajarish uchun xizmat qiladi. Ularning vazifalar quyidagilardan iborat:



Form1-Form1 oynasini yuklash;
Object Inspector- **Object Inspektor**iga o‘tish;
Object TreeView- ob’ektlar daraxtiga o‘tish;
Unit1, pas, Editor- Unit1 dasturlash maydoniga o‘tish;
Next Window – Navbatdagi oynaga o‘tish.

11. Help menyusi. Delphi dasturlash tilidan foydalanish uchun ma’lumotlarni ushbu menyu bandlari orqali olish mumkin:



Delphi help- Delphi dasturlash tilida foydalanish uchun ko'rsatma;

Delphi Tools- Delphi dasturlash tili uskunasi;

Windows SDK- dastur bilan ishlash uchun yordam berish;

Borland Home Page- Borland uy sahifasi;

Borland Developer Network- Borlandga oid aloqa;

Delphi Home Page- Delphining uy sahifasi;

Delphi Developer Support- dasturdan foydalanish uchun yordam olish;

Delphi Direct- Delphi dasturini tuzilishi;

Customize- tuzatish;

About- dastur haqida ma'lumot.

Forma oynasida ilovalar yaratiladi. Object Inspector oynasi obyekt xossalari taxrirlash uchun xizmat qiladi. Obyekt xossalari bu – obyektga berilgan xarakteristika bo'lib, uning ko'rinishi, joylashishi va holatidir. Masalan, Width va Height xossalari forma o'lchamini, top va Left esa formaning ekrandagi holatini, Caption – esa sarlavha matnini aniqlaydi va nomini o'zgartiradi.

Vizual dasturlash texnologiyasida obyekt deganda muloqot oynasi va boshqarish elementlari (kiritish va chiqarish maydoni, buyruq tugmalari, pereklyuchatellar va boshqa) tushuniladi.

Delphi dasturlash tilida ikkita o'zaro ta'sir etuvchi bir-biri bilan bog'liq jarayon asosida tashkil qilinadi: dasturni visual loyihalash jarayoni; dastur kodlarini kiritish (yozish) jarayoni.

Kodlarni yozish uchun maxsus kod oynasi mavjud bo'lib, u dastur matnini kiritish va taxrirlash uchun mo'ljallanganidir. Bu kodlarni yozish oynasida dasturlash Pascal tilining rivoji bo'lgan va kengaytirilgan Object Pascal tilida tuziladi.

Kodlarni yozish oynasi, boshlanishda o'z ichiga hali bo'sh formani akslantiruvchi dastur matnini yozib chiqaradi. Delphi tilida dasturlash forma oynasini tashkil etishdan boshlanadi.

Oddiy dastur ilovasini yaratish ketma-ket File→New→ Application buyrug'ini berish bilan boshlanadi. Bu buyruqni berishdan oldin ikkita asosiy ishni bajarish lozim: papka tashkil etish, tizimni to'g'rilash.

Delphi muhitini standart nastroykasiga o'zgartirish kiritish uchun Tols→Environment Options menyu buyrug'ini berish va muloqot darchasidan kerakli o'zgartirishlarni bajarish lozim.

Delphi dasturlash muhitida ishlash jarayonida quyidagi kengaytmali fayllar ishlatiladi: loyiha fayli, kengaytmasi **.dpr**; pascal moduli fayli, kengaytmasi **.pas**; komponentalar joylashgan fayl, kengaytmasi **.dcl**; formalar joylashgan fayl, kengaytmasi **.dfm**; ma'lumotlar bazasi fayli, kengaytmasi **.dbf**.

Tayyorlanadigan Delphi dastur uchta asosiy etapdan o'tadi: kompilyatsiya, kompanovka, bajarish.

Kompilyatsiya etapidan tayyorlangan dastur matni Object Pascal tiliga o'tkaziladi. Kompanovkada esa kerakli qo'shimcha yordamchi dasturlar unga birlashtiriladi. F9 tugmasini bosish bilan Save UnitAs dialog oynasi paydo bo'ladi va sizdan Unit.pas moduli uchun fayl nomini va joylashadigan papkani ko'rsatishingizni so'raydi. Agar joy ko'rsatilmasa Delphi avtomatik ravishda dasturingizni Bin papkasiga joylashtiradi. Dastur kompilyatsiya qilinishi paytida Delphi sistemasi .pas, .dfm va .dcu kengaytmali modullar tuzadi. .pas kengaytmali fayllar kodlarni yozish oynasida kiritilgan dastur matnini, .dfm forma oynasi tashkil etuvchilarini, .dcu kengaytmali fayl esa .pas va .dfm kengaytmali fayllarning birgalikdagi mashina kodiga o'tkazilgan variantini saqlaydi. Bu .dcu kengaytmali fayl kompilyator tomonidan tashkil qilinadi va yagona ishchi (bajariluvchi) .exe kengaytmali fayl tashkil qilishga baza yaratadi.

2.4. Loyiha va modul tushunchasi

Delphi dasturi – bu bir necha bir biri bilan bog'liq fayllar. Har qanday dastur .dpr kengaytmali loyiha fayli va bir yoki bir necha .pas kengaytmali modullardan tashkil topadi. Loyiha fayli dasturchi tomonidan kiritilmaydi, u foydalanuvchining ko'rsatmalari asosida avtomatik ravishda Delphi sistemali dasturi tomonidan tuziladi. Loyiha fayli matnini ko'rish uchun Project/View Source buyryg'ini berish zarur. Loyiha matni umumiy holda quyidagicha bo'lishi mumkin.

Program Project1;

Uses

Forms,

Unit1 in 'Unit1.pas' {Form1}

*{\$R *.res}*

Begin

Application.Initialize;

Application.CreateForm(Tform,Form1);

Application.Run; End.

Loyiha nomi dasturchi tomonidan loyihani saqlash vaqtida beriladi va Delphi muhitida bajariluvchi fayl, ya'ni kengaytmasi .exe bo'lgan faylni tashkil qilishni aniqlaydi. Loyiha faylidan keyin ishlatiladigan modullar: standart modullar Forms va Unit1 joylashadi. **{\$R *.res}** direktivasi kompilyatorga ishlatilishi kerak bo'lgan resurs fayllari, masalan dasturlarni e'lon qilish kerakligini bildiradi. Yulduzcha belgisi resurs faylining kengaytmasi .res ekanligini bildiradi. Bosh modulning bajariluvchi qismi Begin .. End operatorlari orasiga joylashadi.

Loyihani saqlash. Ilovani kompilyatsiya qilish va ishga tushirish. Loyihani saqlashda Delphi bir necha fayl tashkil qiladi. Ayrim loyiha butunloyihani tavsiflashni, boshqalari forma va dastur modulini tavsiflashni o'z ichiga oladi. Agar hali saqlanmagan loyiha bo'lsa Fayl (File) menyusidan Soxranit proyekt (Save Project) buyrug'i beriladi va dastur moduli va proyekt nomi beriladi.

Modul tushunchasi. Modul – bu biror bir dastur. Modullar standart loyihasi ega. Delphi tizimini ishga tushirgandan keyin modul strukturasi quyidagi ko'rinishda bo'ladi.

Unit unit1;

Interface

Uses

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

Type

TForm1 = class(TForm)

Private

{ Private declarations }

Public

{ Public declarations } end;

Var

Form1: TForm1;

Implementation

*{ \$R *.dfm } End.*

Unit – modulning sarlavhasi;

Interface – modulning interfeysi, ya'ni dastur va boshqa modullar uchun ochiq ko'rinarli qismining boshlanishini bildiradi. Bu qismda o'zgarmaslar, kattaliklar tiplari, protsedura va funksiyalar aniqlanib ko'rsatilgan bo'ladi, lekin ularning butun ko'rinishi keyingi yopiq qismda beriladi.

Implementation – modulning dastur va modullar uchun yopiq, ya'ni ko'rinmaydigan qismi boshlanishini bildiradi. Bu yerda interfeys qismida aniqlangan protsedura va funksiyalar yana bir marta ko'rsatilishi shart (ularning sarlavhalari bir xil bo'lishi kerak).

Asosiy qismi Begin yozuvidan keyin boshlanadi, agar bu qism mavjud bo'lmasa Begin ham bo'lmaydi. Bu qismda boshqaruvni asosiy programmaga o'tkazishgacha qadar bajarilishi lozim bo'lgan operatorlar ro'yxati turadi.

Sinflar va obektlar. Object Pascal obyektga yo'naltirilgan dasturlash tilidir. Obyektga yo'naltirilgan tilga yig'ilgan imkoniyatlarga dasturlash tilining obyekt modellari deyiladi. Object Pascalda obyekt modellari ishlatilishining amaliy natijasi komponentlarni yaratish va ularni qo'llab quvvatlashdir.

Ma'lumki, Delphi dasturiy vositasi Object Pascal obyektga yo'naltirilgan dasturlash tilini ishlatadi. Obyektga yo'naltirilgan dasturlash (OYD) – bu dastur ishlab chiqish usuli bo'lib, uning asosida real dunyo obyekt va uning holatini ifodalovchi ma'lum strukturaga ega obyekt tushunchasi yotadi. Object Pascalda obyekt modelining qo'llanilishi natijasi bu komponentlarni qo'llash va yaratishdir. Object Pascal obyekt modelining asosi sinf va obyekt tushunchalaridir.

Sinf- bu Object Pascalda maxsus turlar bo'lib, o'zida maydon, usullar va xossalarni mujassamlashtiradi.

Pascal sinfik tili dasturchiga o'zining murakkab ma'lumotlar turlari - yozuvlar (records) kiritishga imkon beradi. Obyektli dasturlash konsepsiyasiga asoslangan Delphi tili, sinflar kiritishga imkon beradi. Sinf murakkab struktura bo'lib, ma'lumotlar ta'riflaridan tashqari, prosedura va funksiyalar ta'riflarini o'z ichiga oladi.

Obyekt –bu sinfning real nusxasi bo'lib, ma'lumotlar va funksiyalardan tashkil topadi. U dasturning Var bo'limida e'lon qilinadi.

Obyektlar sinflar vakillari sifatida dasturda var bo‘limida ta’riflanadi, masalan:
var

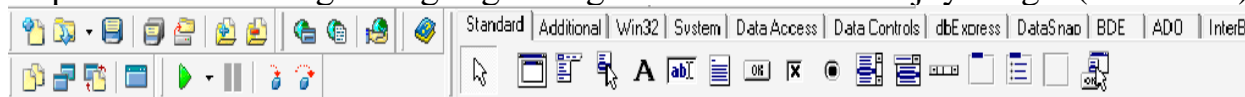
student: TPerson; professor: TPerson;

Delphida obyekt – bu dinamik strukturadir. O‘zgaruvchi-obyekt ma’lumotlarni emas, obyekt ma’lumotlariga ilovani o‘z ichiga oladi. Shuning uchun dasturchi bu ma’lumotlarga xotiradan joy ajratishni ko‘zda tutishi lozim.

Joy ajratish sinf maxsus usuli – konstruktor yordamida amalga oshiriladi. Bu usul odatda Create (yaratish) nomiga ega bo‘ladi. Sinf ta’rifida konstruktor uchun procedure so‘zi o‘rniga constructor so‘zi ishlatiladi.

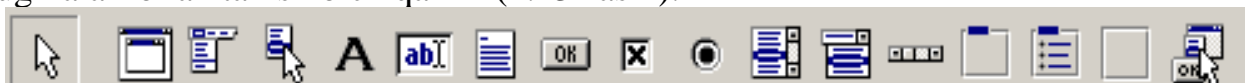
2.5. Delphida komponentalar ro‘yxati

Delphi dasturlash tilida uskunalar paneli va komponentalar ro‘yxati komponentalar bajaradigan vazifalariga qarab guruhlariga ajratilgan. Har bir guruh komponentalari o‘zinig nomiga ega bo‘lgan alohida bo‘limda joylashgan(2.22-rasm).















2.7-rasm.

Dastur oynasining chap tomonida uskunalar paneli va o‘ng tomonida komponentalar palitrasi joylashgan bo‘ladi. Biz hozir standart komponentalar palitrasi tugmalari bilan tanishib chiqamiz (2.23-rasm).



2.8-rasm.

-  **MainMenu** – formada ustun va satr bo‘yicha kontekstli menyu hosil qiladi;
-  **PopupMenu** – formada ustun bo‘yicha kontekstli menyu hosil qiladi;
-  **Label** – formada yozuv hosil qiladi;
-  **Edit** – formada bir satrli matn kiritish maydonini hosil qiladi;
-  **Memo** – formada ko‘p satrli matn maydonini 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 hosil qiladi;
-  **List Box** – formada bir nechta satrlar ro‘yxatini tuzish imkoniyatini hosil qiladi;
-  **Combo Box**–bir nechta satrdan birortasini tanlash imkoniyatini hosil qiladish;
-  **Scroll Bar** – formada siljitish vositasini yaratish imkoniyatini hosil qiladi;
-  **GroupBox** – formada bir nechta variantdan birortasini tanlash imkoniyatini hosil qiladi;

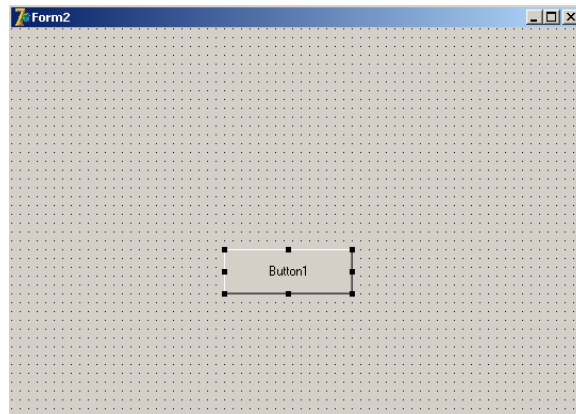


Radio Group – formada bir nechta variantdan birortasi tanlanadigan ro‘yxatni tuzish imkoniyatini hosil qiladi;

Panel – formada panelni yaratishda imkoniyatini hosil qiladi.

Standard komponentalar palitrasi:

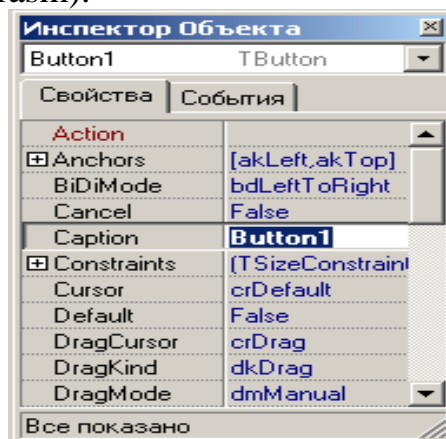
1. Button tugmasi. Button tugmasini bosilishi natijasida kutilishi lozim bo‘lgan jarayonlar (masalan, hisoblashlar yoki bajarilishi lozim bo‘lgan operatsiyalar) bajarilishga xizmat qiladi.



2.9-rasm.

Button tugmasini ekranga joylash uchun Delphi dasturlash tilining **Standart** komponentalar palitrasidan **OK** piktogrammasi belgilanib forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda tugmacha qo‘yilishi lozim bo‘lgan joy ajratiladi. Natijada, **Button1** tugmasi hosil qilinadi. Ikkinchi marta tanlansa, **Button2** va hokazo **ButtonN** ko‘rinishda davom etadi.

Agar, siz tugmachadagi **Button1** so‘zini boshqa so‘zga almashtirmoqchi bo‘lsangiz, **Object Inspektorining Caption** xossasidan foydalanib o‘zgartirish kiritishingiz mumkin(2.10-rasm).



2.10-rasm.

Tugmadagi so‘zni kattalashtirish yoki kichiklashtirishga to‘g‘ri kelsa, **Object Inspektorining Font-{Tfon}** bandi tanlanadi va bu yerdan kerakli o‘lchamlarni tanlash mumkin.

Yuqorida paydo bo‘lgan ko‘rinishdan nafaqat **Button** tugmasi ustida amallar bajaribgina qolmasdan, boshqa tugmalar ustida ham bajarish, shu bilan bir qatorda dasturlash maydoniga turli xil bezaklar berishda ham foydalanishimiz mumkin.

Demak, **Button** tugmasi bosilishi natijasida kutilishi lozim bo‘lgan jarayonlar bajarilishga xizmat qilalar ekan. Biz bu tugmaga doir bir nechta misollar ko‘rib chiqamiz.

1-misol. Dasturdan chiqish buyrug‘ini bajaruvchi dastur tuzing.

Standard komponentalar palitrasidan **Button1** tugmasini hosil qilamiz va sichqoncha tugmasini ikki marta bosgan holda **Until1.pas** maydoniga **form1.close** buyrug‘i kiritiladi.

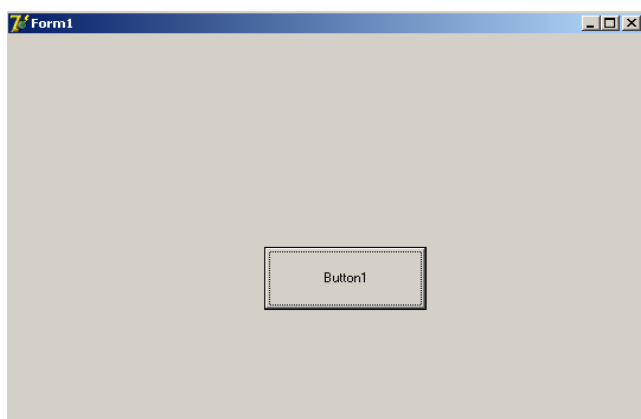
```
{ $R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);

begin
form1.close;
end;

end.
```

Yuqoridagi topshiriq bajarilgandan so‘ng, bu dasturni birinchi marotaba **File**→**Save As**, Ikkinchi marotaba **File**→**Save project As** ketma-ketlikni bajariladi va dasturdan chiqib, **Project1** faylni qayta yuklaymiz. Natijada, quyidagi ko‘rinish paydo bo‘ladi(2.11-rasm):



2.11-rasm.

Agar siz tugmacha ustiga sichqoncha chap tugmasini bir marta bossangiz hosil qilingan dasturdan chiqishga muvassar bo‘lasiz.

2. Label komponentasi. **Label** tugmasi nafaqat ma’lumotlarni ekranga joylashtirish uchun xizmat qiladi, balki dastur natijalarini chiqarishda ham ishlatish mumkin. Buning uchun dasturda **Labeln.caption:='Dastur natijasi'**; buyrug‘i berilishi lozim.

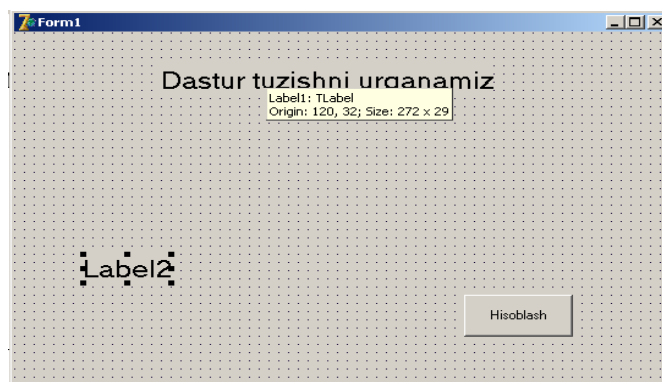
Label tugmasini **Forma** oynasiga joylashtirish uchun **Delphi** dasturlash tilining **Standart** komponentalar palitrasidan **A** piktogrammasi belgilanib, forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda tugma qo‘yilishi lozim bo‘lgan joy ajratiladi. Natijada, **Label1** tugmasi hosil qilinadi. Ikkinchi marta tanlansa **Label2** va hokazo **LabelN** ko‘rinishda davom etadi.

2-misol. **Label** tugmasi yordamida ma’lumotlarni ekranga joylashtirish va dastur natijasini chiqarish yo‘lini ko‘rib chiqamiz.

1. **Standart** komponentalar palitrasidan **A** piktogrammasi belgilanadi va forma ustiga sichqoncha tugmachasini bosgan holda tugma qo‘yilishi lozim bo‘lgan joy ajratiladi. Natijada, **Label1** tugmasi hosil bo‘ladi, **Object Inspektorining Caption** bandidagi **Label1** so‘zini o‘chirib “Dastur tuzishni o‘rganamiz” jumlasini kiritiladi.

2. Ikkinchi marotaba xuddi shu yo‘lni tanlab, **Label2** tugmasi hosil qilinadi.

3. **Button1** tugmasini hosil qilib, **Caption** bandiga “Hisoblash” jumlasini kiritiladi.



2.12-rasm.

4. **Button1(hisoblash)** bandiga sichqoncha tugmasi ikki marta bosilib, dasturlash maydoniga **Label2.caption:= 'Salom yoshlar: jumlasini kiritiladi'**.

```

Form1: TForm1;

implementation

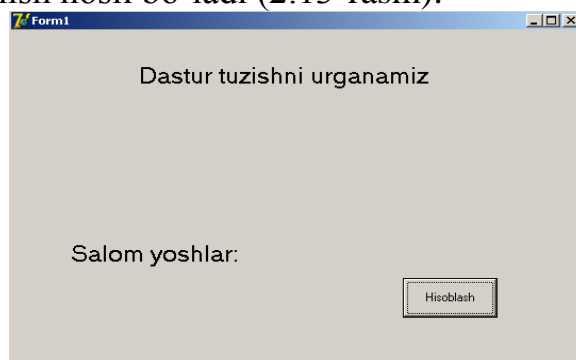
{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin
Label2.Caption:='Salom yoshlar:';
end;

end.

```

5. Yuqoridagi amallar ketma-ketligi kiritilgandan so'ng, **F9** tugmasi bosiladi. Natijada, quyidagi ko'rinish hosil bo'ladi (2.13-rasm):



2.13-rasm.

3. Edit (kiritish) qatori. **Edit** kiritish qatori matnni bir qatordan kiritish va uni tahrirlash uchun ishlatilgina qolmasdan balki ma'lumotlarni chiqarishda ham xizmat qiladi. Matn kiritish va chiqarish qatorini ekranga joylash uchun **Delphi**ning **Standard** komponentalar palitrasidan **ab** piktogrammasi belgilanib forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda matn kiritilishi lozim bo'lgan joy ajratiladi. Natijada, **Edit1** matn kiritish maydoni hosil qilinadi. Matnni kiritish dastur ishchi holatiga o'tilganda bajariladi.

3-misol. **Edit** matn kiritish va chiqarish qatoridan foydalanishni ko'rib chiqamiz.

1. **Standard** komponentalar palitrasidan **ab** piktogrammasi belgilanib, **forma** ustiga kelinadi va sichqoncha tugmachasini bosgan holda **Edit1** matn kiritilishi va ikkinchi marotaba **Edit2** chiqarilishi lozim bo'lgan joy ajratiladi.

2. **Standard** komponentalar palitrasidan **Button1** tugmasini hosil qilinadi.

3. Natijada, hosil bo'lgan **Button1** tugmasi ustiga sichqoncha tugmachasini ikki marta bosib, dasturlash maydoni hosil qilinadi.

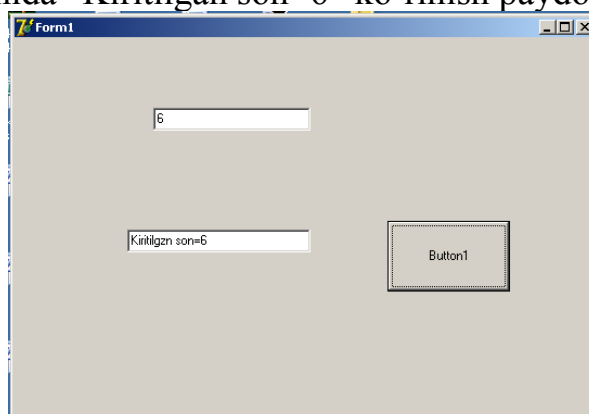
4. Demak, hosil bo‘lgan maydonga quyidagi dastur kodi kiritiladi:

```
var  
m:integer;  
begin  
m:=strtoint(edit1.Text);  
edit2.text:=edit2.text+'Kiritilgan son='+inttostr(m);  
end;
```

```
Form1: TForm1;  
  
implementation  
{ $R *.dfm }  
  
procedure TForm1.Button1Click(Sender: TObject);  
var m:integer;  
begin  
m:=strtoint(edit1.Text);  
edit2.text:=edit2.text+'Kiritilgan son='+inttostr(m);  
end;  
end.
```

5. Dastur bajarilish uchun **F9** tugmasi tanlanadi.

6. Shundan so‘ng, hosil bo‘lgan katakka, ya’ni **Edit1** maydoniga 6 sonini kiritiladi. Natijada, **Edit2** maydonida “Kiritilgan son=6” ko‘rinish paydo bo‘ladi (2.14-rasm).



2.14-rasm.

4. CheckBox tugmasidan foydalanish. **CheckBox** – tugmasi yordamid amallar bajarish yoki aksincha bajarilgan amallarni bekor qilish uchun xizmat qiladi. **CheckBox** tugmasini forma oynasiga joylash uchun **Delphining Standard** komponentalar palitrasidan piktogrammasi belgilanib, forma ustiga kelinadi va lozim bo‘lgan joy ajratiladi. Natijada, **CheckBox1** tugmasi hosil qilinadi.

4-misol. CheckBox1 tugmasidan o‘chirish yoki belgilashda foydalanish.

1. **Standard** komponentalar palitrasidan **ab** piktogrammasi belgilanib, forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda **Edit1** matn kiritilishi lozim bo‘lgan joy ajratiladi.

2. **Standard** komponentalar palitrasidan piktogrammasi belgilanib forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda **CheckBox1** tugmasi hosil qilinadi.

3. **Standard** komponentalar palitrasidan **Label1** va **Button1** tugmalari hosil qilinadi.

4. Demak, hosil bo‘lgan **Button1** tugmasi ustiga sichqoncha tugmachasini ikki marta bosib, dasturlash maydonni hosil qilinadi.

5. Dasturlash maydoniga quyidagi dastur kodi kiritiladi:

```
var  
a,s:real;
```

begin

```
a:=strtofloat(edit1.Text);
```

```
s:=a; if CheckBox1.Checked then s:= s*2;
```

```
label1.Caption:='s='+floattostr(s); end;
```

```
{ $R *.dfm }
```

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var
```

```
a,s:real;
```

```
begin
```

```
  a:=strtofloat(edit1.Text);
```

```
  s:=a;
```

```
  if CheckBox1.Checked then s:= s*2;
```

```
  label1.Caption:='s='+floattostr(s);
```

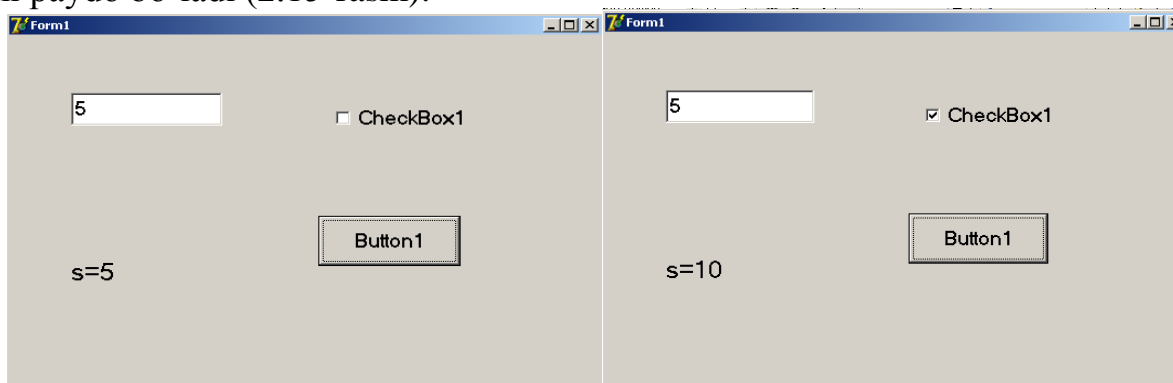
```
end;
```

```
end.
```


6. Dasturning bajarilishini amalga oshirish uchun **F9** tugmasi tanlanadi.

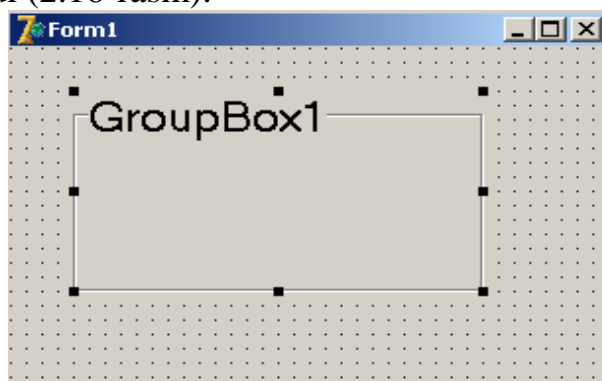
7. Hosil bo‘lgan katakchaga (Edit1) maydoniga 5 sonini kiritiladi va **Button1** tugmasi tanlansa oynada 5 soni paydo bo‘ladi.

8. Agar **Edit** maydoniga 5 sonini kiritib, **CheckBox1** tugmasi tanlansa, oynada 10 soni paydo bo‘ladi (2.15-rasm).




2.15-rasm.

5. GroupBox tugmasidan foydalanish. **GroupBox** – tugmasi yordamida formadagi bir nechta tugmalarni bir-biri bilan o‘zaro bog‘lash uchun xizmat qiladi. Tugmalar bilan bog‘lanish hosil qilish uchun, **Delphining Standard** komponentalar palitrasidan  piktogrammasi belgilanib, forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda joylashtirish lozim bo‘lgan joyi ajratiladi. Natijada, **GroupBox1** tugmasi forma oynasiga joylashtiriladi (2.16-rasm).




2.16-rasm.

6. ScrollBar tugmasidan foydalanish. **ScrollBar** – tugmasi formada siljitish vositasini yaratishda uchun xizmat qiladi. Siljitish vositasini forma oynasiga

joylashtirish uchun **Delphining Standard** komponentalar palitrasidan  tugmasi belgilanib, forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda tanlash lozim bo'lgan joy ajratiladi. Natijada, **ScrollBar1** siljitish tugmasi hosil qilinadi. Siljitish dastur ishchi holatiga o'tilganda bajariladi.

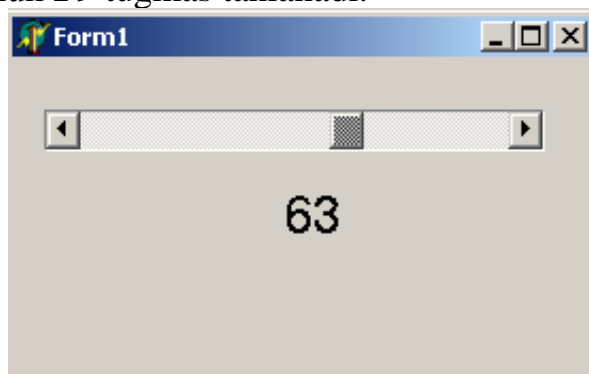
5-misol. ScrollBar tugmasidan foydalanish.

1. **Standard** komponentalar palitrasidan  piktogrammasi belgilanib forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda **ScrollBar1** maydonini hosil qilamiz.
2. **Standard** komponentalar palitrasidan **Label1** tugmasi hosil qilinadi.
3. **Object Inspektorining Caption** xossasidan **Label1** yozuvi o'rniga "0" sonini kiritamiz.
4. **ScrollBar1** tugmasini ustiga sichqoncha tugmachasini ketma-ket ikki marta bosiladi va dasturlash maydoniga quyidagi dastur kodi kiritiladi:


Label1.Caption:=IntToStr(ScrollBar1.Position);

```
{ $R *.dfm }  
procedure TForm1.ScrollBar1Change(Sender:  
begin  
Label1.Caption:=IntToStr(ScrollBar1.Position);  
end;  
end. |
```



5. Natijani ko'rish uchun **F9** tugmas tanlanadi.



2.17-rasm.

7. RadioButton tugmasidan foydalanish. **RadioButton** – tugmasidan bir nechta variantdan birortasini tanlash imkoniyatini yaratish uchun xizmat qiladi. Variantdan birortasini tanlash tugmachasini forma oynasiga joylashtirish uchun **Delphining Standard** komponentalar palitrasidan  piktogrammasi belgilanib, forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda tanlash lozim bo'lgan joy ajratiladi. Natijada, **RadioButton1** tanlash tugmasi hosil qilinadi. Tanlash dastur ishchi holatiga o'tilganda bajariladi.

6-misol. "RadioButton" tugmasidan foydalanish.

1. **Standard** komponentalar palitrasidan  piktogrammasi belgilanib forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda **GroupBox1** maydoni hosil qilinadi.
2. **Standard** komponentalar palitrasidan  piktogrammasi belgilanib, **GroupBox1** forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda **RadioButton1** va ikkinchi marotaba **RadioButton2** tanlash lozim bo'lgan joy ajratiladi.
3. **Standard** komponentalar palitrasidan **Label1** tugmasi hosil qilinadi.

4. **Object Inspektorining Caption** xossasidan **Label1** yozuvi o‘rniga “Dastur” so‘zini kiritiladi.

5. **RadioButton1** tugmasini ustiga sichqoncha tugmacahsini ketma-ket ikki marta bosiladi va dasturlash maydoniga quyidagi dastur kodi kiritamiz:

Label1.Enabled := False;

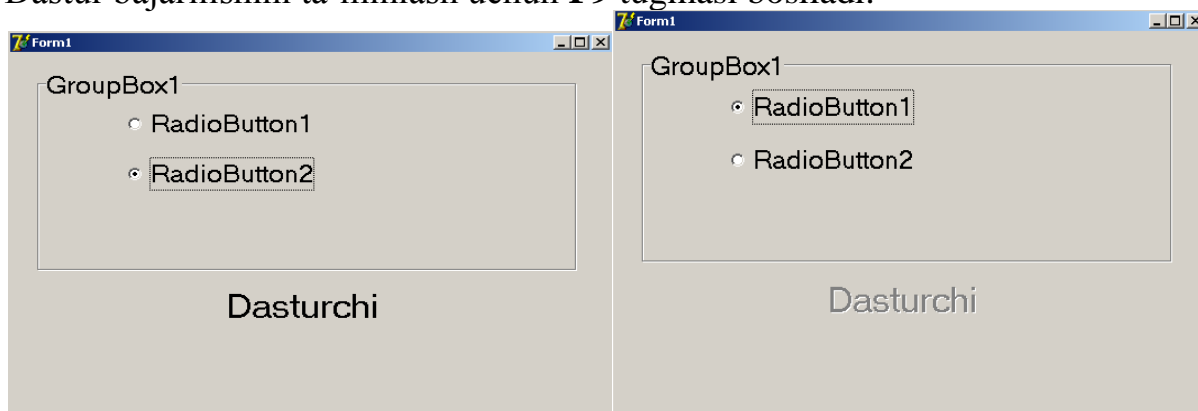
6. **RadioButton2** tugmasini ustiga sichqoncha tugmacahsini ketma-ket ikki marta bosiladi va dasturlash maydoniga quyidagi dastur kodi kiritiladi:

Label1.Enabled := True;

```
{ $R *.dfm}


procedure TForm1.RadioButton2Click(Sender:
begin
Label1.Enabled := True;
end;
procedure TForm1.RadioButton1Click(Sender:
begin
Label1.Enabled := False;
end;
end. |
```

7. Dastur bajarilishini ta’minlash uchun **F9** tugmasi bosiladi.




2.18-rasm.

8. Agar siz **RadioButton2** tugmasini tanlasangiz dastur so‘zi faollashadi va aksincha **RadioButton1** tanlasangiz dastur xira holga keladi.

8. RadioGroup tugmasidan foydalanish. RadioGroup – tugmasi formada bir nechta variantdan birortasi tanlanadigan ro‘yxatni tuzish uchun xizmat qiladi. Tanlash tugmasini forma oynasiga joylashtirish uchun **Delphi**ning **Standard** komponentalar palitrasidan  piktogrammasi belgilanib, forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda tanlash lozim bo‘lgan joy ajratiladi. Natijada, **RadioGroup1** tanlash tugmasi hosil qilinadi. Tanlash dastur ishchi holatiga o‘tilganda bajariladi.

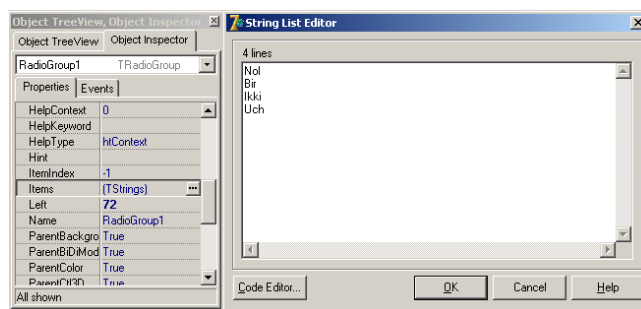
7-misol. RadioGroup tugmasidan foydalanish.

1. **Standard** komponentalar palitrasidan  piktogrammasi belgilanib, forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda **RadioGroup1** maydoni hosil qilinadi.

2. **Standard** komponentalar palitrasidan **Label1**, **Button1** va **Edit1** tugmalari hosil qilinadi.

3. **RadioGroup1** maydoniga sichqoncha tugmasi bir marta bosiladi.

4. **Object Inspektorining Items** xossasidan **TStrings** yozuvi ustiga sichqoncha tugmasi bir marta bosiladi, natijada quyidagi rasm paydo bo‘ladi(2.19-rasm):



2.19-rasm.

5. Hosil bo‘lgan maydonga “Nol, Bir, Ikki, Uch” yozuvlari kiritiladi.

6. **RadioGroup1** tugmasini ustiga sichqoncha tugmasi ketma-ket ikki marta bosiladi va dasturlash maydoniga quyidagi dastur kodi kiritiladi:

Label1.Caption:=IntToStr(RadioGroup1.ItemIndex);

7. **Button1** tugmasini ustiga sichqoncha tugmasi ketma-ket ikki marta bosiladi va dasturlash maydoniga quyidagi dastur kodi kiritiladi:

Edit1.Text:=IntToStr(RadioGroup1.ItemIndex);

```

implementation

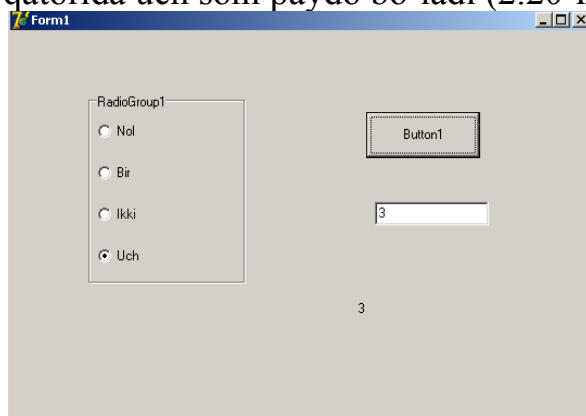
({$R *.dfm})

procedure TForm1.RadioGroup1Click(Sender: T
begin
  Label1.Caption:=IntToStr(RadioGroup1.ItemI
end;
  procedure TForm1.Button1Click(Sender: T
begin
  Edit1.Text:=IntToStr(RadioGroup1.ItemIndex
end;
end.


```

8. Dasturni ishchi holatga o‘tkazish uchun F9 tugmasi tanlanadi.


9. Dastur ishchi holatiga o‘tilganda so‘ng, to‘rtinchi doirani belgilab, **Button1** tugmasi tanlansa, **Edit** qatorida uch soni paydo bo‘ladi (2.20-rasm).



2.20-rasm.

9. Panel tugmasidan foydalanish. Panel – tugmasi formada panelni yaratish uchun xizmat qiladi. Siljitish vositasini forma oynasiga joylashtirish uchun **Delphining Standard** komponentalar palitrasidan  piktogrammasi belgilanib, forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda tanlash lozim bo‘lgan joy ajratiladi. Natijada, **Panel1** tugmasi hosil qilinadi.

8-misol. Panel tugmasidan foydalanish.

1. **Standard** komponentalar palitrasidan  piktogrammasi belgilanib, forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda **Panel1** maydonini hosil qilinadi.
2. **Standard** komponentalar palitrasidan **Edit1** tugmasi hosil qilinadi.
3. **Panel1** maydoni ustiga **Button1** va **Button2** tugmalari hosil qilinadi.
4. **Button1** tugmasini ustiga sichqoncha tugmacahsini ketma-ket ikki marta bosiladi va dasturlash maydoniga quyidagi dastur kodi kiritiladi:

Form1.close;

5. **Button2** tugmasini ustiga sichqoncha tugmasini ketma-ket ikki marta bosiladi va dasturlash maydoniga quyidagi dastur kodi kiritiladi:

edit1.Text:=' ';

edit1.SetFocus;

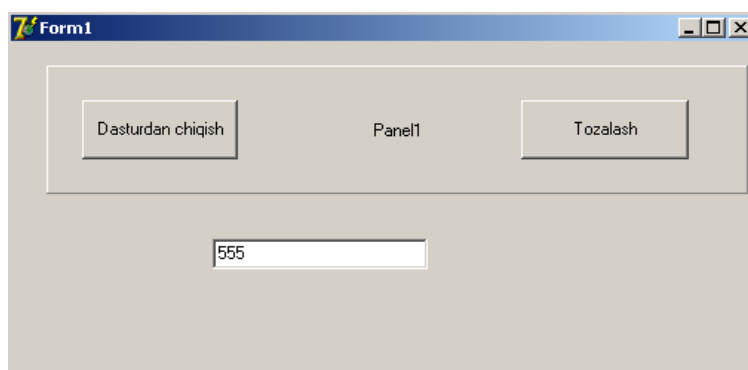
```

($R *.dfm)

procedure TForm1.Button1Click(Sender: TObject)
begin
Form1.close;
end;
    procedure TForm1.Button2Click(Sender: TObject)
begin
edit1.Text:=' ' ;
    edit1.SetFocus;
    end;
end.



```

6. Dasturni ishchi holatga o'tkazish uchun F9 tugmasi tanlanadi.



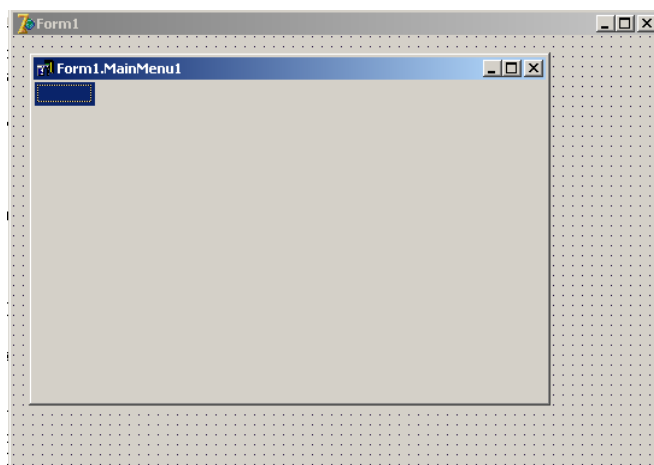
2.21-rasm.

7. Hosil bo'lgan dasturda **Button1** tugmasini bir marta bossangiz dasturdan chiqasiz. Agar **Button2** tugmachasini tanlasangiz **Edit** qatoriga kiritilgan matinni o'chiradi.

10. MainMenu va PopupMenu tugmalarida foydalanish. **MainMenu** va **PopupMenu** tugmalari menyu yaratishda foydalaniladi. **PopupMenu** tugmasi ustun bo'yicha menyu yaratish uchun mo'ljalangan bo'lsa, **MainMenu** tugmasi esa ustun va satrlar bo'yicha menyular yaratishda foydalaniladi. Menyu yaratish tugmalaridan foydalanish uchun dastur oynasning **Standard** komponentalar palitrasidan  yoki  piktogrammasi belgilanib, forma ustiga kelinadi va sichqoncha tugmasini bosgan holda joylashtiriladi.

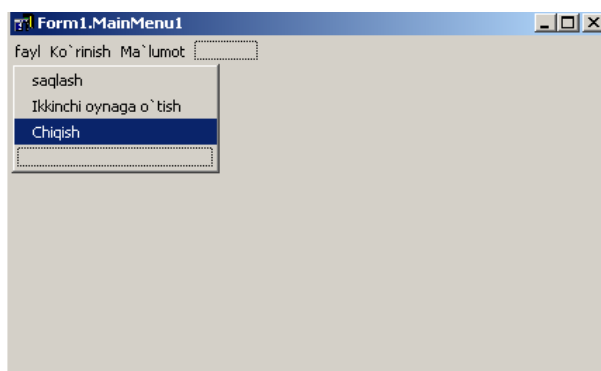
9- misol. **MainMenu** tugmasi yordamida dasturdan chiqish menyusini yaratish.

1. **Standard** komponentalar palitrasidan **MainMenu** tugmasi tanlanadi.
2. Hosil bo'lgan tugma ustiga sichqoncha tugmasini ketma-ket ikki marta bosiladi. Natijada, **Form1 MainMenu1** maydoni paydo bo'ladi.



2.22-rasm.

3. **Object Inspektorining Caption** bandiga “Fayl” jumlasini kiritiladi.
4. Fayl menyusiga bir nechta bandlarni kiritishingiz mumkin. Misol tariqasida “saqlash”, “boshqa oynaga o‘tish”, “Chiqish”. Fayl menyusining gorizantal yo‘nalishi bo‘yicha ham yorliqlar yaratish mumkin. Masalan: “Ko‘rinish”, “Ma’lumot”.



2.23-rasm.

5. Forma oynasining menyu bo‘limlarining har bir bandga kerakli dasturlar tuzib foydalanish mumkin. Misol sifatida “Chiqish” bandini bosganda dasturdan chiquvchi dastur yaratish uchun “Chiqish” bandining dasturlash maydoniga quyidagi dastur kodi kiritiladi: *form1.close;*

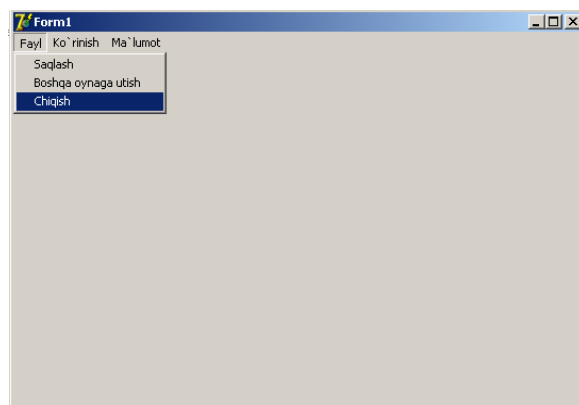
```

procedure TForm1.ChiqishiClick(Sender: TObject)
begin
    form1.close;
end;

end.


```

6. Ma’lumotlar kompyuter xotirasiga saqlanadi.
 1. File-Save Ass...
 2. File-Save Project Ass..
7. **F9** tugmasi tanlanadi.



2.24-rasm.

Yuqoridagi ketma-ketlik amallari bajarilgandan so‘ng, fayl menyusning “Chiqish” bandiga sichqoncha tugmachasini bir marotaba bosamiz va dasturdan chiqishimiz mumkin.

11. Memo tugmasidan foydalanish. **Memo** tugmasi matnlarni bir va bir necha qator qilib chiqarish hamda sodda matn muharrirlari yaratish uchun ishlatiladi. Matnlarni bir necha qator qilib chiqarish uchun, dastur oynasining **Standart** komponentalar palitrasidan  piktogrammasi belgilanib, forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda joylashtiriladi. Natijada, **Memo1** matn chiqarish maydoni hosil qilinadi. Bu matn chiqarish maydoni dasturda natijalarni chiqarishda qo‘l keladi. Natijani chiqarishda u dastur ichida quyidagicha ishlatiladi:

Memo1.Lines.add('Yechim='+S);

Agar biron bir masalani son qiymatini chiqarish uchun esa quyidagicha:

Memo1.Lines.add('Yechim='+floattostr(s));

Memo maydonini tozalash uchun esa natijani chiqarishdan oldin modulda *Memo1.Clear;* buyrug‘ini berish bilan amalga oshiriladi.

10- misol. Haftada ish va dam olish kunlarini chiqaruvchi dastur tuzish.

Bu masalani dasturini tuzishimiz uchun, bitta **Button1** va **Memo1** tugmasidan foydalaniladi.

1. **Button1** va **Memo1** tugmalari hosil qilinadi.
2. **Button1** tugmasini ustiga sichqoncha tugmachasini ketma-ket ikki marta bosiladi.
3. Hosil bo‘lgan dasturlash maydoniga quyidagi dastur kodi kiritiladi:

begin

Memo1.Clear;

Memo1.Lines.add('Ish kunlari');

Memo1.Lines.add('Dushanba');

Memo1.Lines.add('Seshanba');

Memo1.Lines.add('Chorshanba');

Memo1.Lines.add('Payshanba');

Memo1.Lines.add('Juma');

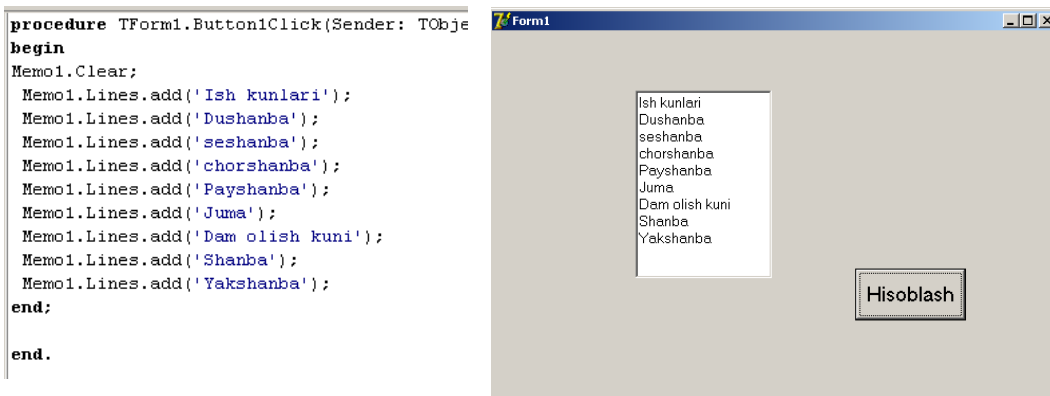
Memo1.Lines.add('Dam olish kuni');

Memo1.Lines.add('Shanba');

Memo1.Lines.add('Yakshanba');


end; end.

4. **F9** tugmasi tanlanadi.



2.25-rasm.

5. Yuqoridagi bosqichlardan so‘ng, hisoblash tugmasi bosiladi. Natijada, 2.25-rasm ko‘rinishidagi oyna paydo bo‘ladi.

12. List Box tugmasi. **List Box** tugmasi formada bir nechta satrlar ro‘yxatini tuzish imkonini yaratadi. Satrlar ro‘yxatini tuzish tugmasini hosil qilish uchun dastur oynasining **Standart** komponentalar palitrasidan  piktogrammasi belgilanib, forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda joylashtiriladi. Natijada, **List Box1** satrlar ro‘yxatini tuzish imkonini yaratiladi. Satrlar ro‘yxatini tuzish dastur ishchi holatiga o‘tilganda bajariladi.

11-misol. List Box1 satrlar ro‘yxatini tuzish dasturi.

1. **Standart** komponentalar palitrasidan **Button1**, **Edit1** va **List Box1** tugmalari hosil qilinadi.

2. **Object Inspektoridan** **Button1** matn o‘rniga “O‘tkazish” matniga almashtiriladi.

3. Dastur maydoni ustiga sichqoncha tugmasini ketma-ket ikki marta bosiladi va dasturlash maydoniga o‘tiladi.

4. Dasturlash maydoniga quyidagi dastur kodi kiritiladi: **ListBox1.Clear;**

5. **Button1** tugmasini dasturlash maydoniga quyidagi:

begin

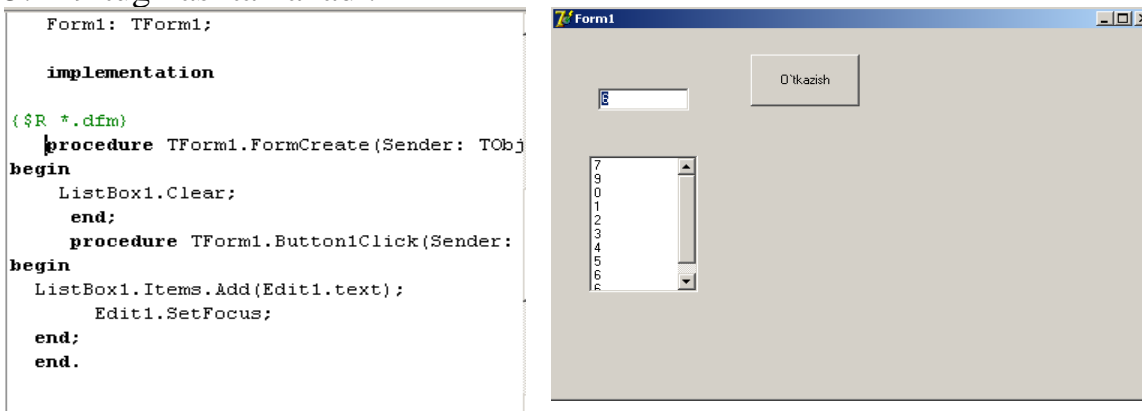
ListBox1.Items.Add(Edit1.text);

Edit1.SetFocus;

end;


dastur kodi kiritiladi.

5. **F9** tugmasi tanlanadi.



2.26-rasm.

Hosil bo‘lgan dasturda siz **Edit** qatoriga qiymat kiritib o‘tkazish tugmasini tanlasangiz, qiymatlar **ListBox** maydonida namoyon bo‘ladi (2.26-rasm).

13. ComboBox tugmasi. **ComboBox** tugmasi formada bir nechta satrlar ro'yxatini tuzish imkonini yaratad. Satrlar ro'yxatini tuzish tugmasini hosil qilish uchun **Standard** komponentalar palitrasidan  piktogrammasi belgilanib, forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda joylashtiriladi. Natijada, **ComboBox1** satrlar ro'yxatini tuzish imkonini yaratiladi. Satrlar ro'yxatini tuzish dastur ishchi holatiga o'tilganda bajariladi.

12-misol. ComboBox1 satrlar ro'yxatini tuzish dasturi.

1. **Standard** komponentalar palitrasidan **Button1** va **ComboBox1** tugmalari hosil qilinadi.

2. **Object Inspektoridan** **Button1** matn o'rniga "O'tkazish" matni bilan almashtiriladi.

3. Dastur maydoni ustiga sichqoncha tugmasini ketma-ket ikki marta bosiladi va dasturlash maydoniga o'tiladi.

4. Dasturlash maydoniga quyidagi:

ComboBox1.Clear;

dastur kodi kiritiladi.

5. **Button1** tugmasi dasturlash maydoniga quyidagi:

begin

ComboBox1.Items.Add(ComboBox1.text);

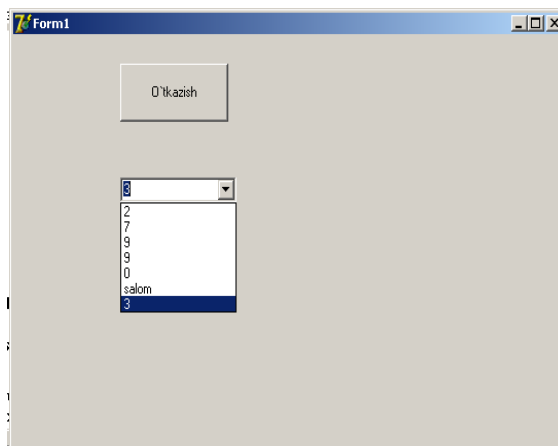
ComboBox1.SetFocus;

end;

dastur kodi kiritiladi.

6. F9 tugmasi tanlanadi.

```
implementation
($R *.dfm)
  procedure TForm1.Button1Click(Sender: TObject)
  begin
    ComboBox1.Items.Add(ComboBox1.text);
    ComboBox1.SetFocus;
  end;
  procedure TForm1.FormCreate(Sender: TObject)
  begin
    ComboBox1.Clear;
  end;
end.
```

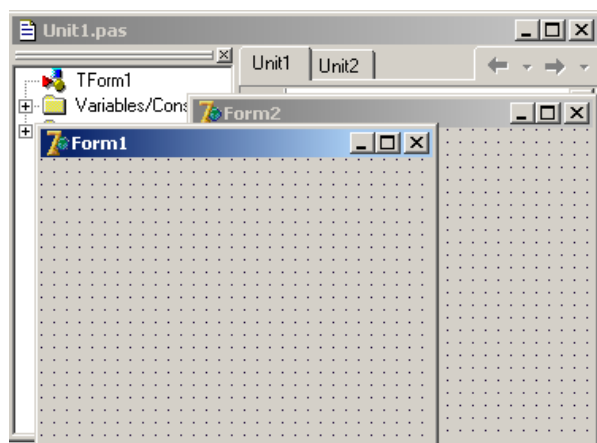


2.27-rasm.

Hosil bo'lgan ko'rinishga ma'lumot kiritib o'tkazish tugmasini tanlasangiz sonlar ketma-ketligi paydo bo'ladi (2.27-rasm).

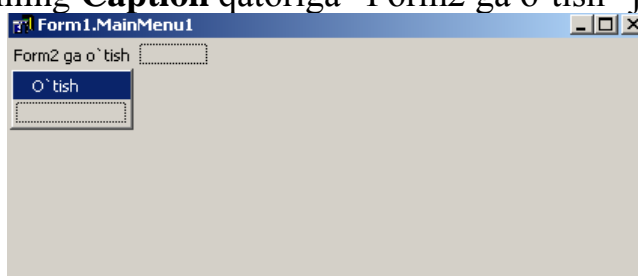
13-Misol. Bir forma oynasidan ikkinchisiga yoki ikkinchi forma oynasidan birinchi forma oynasiga o'tuvchi dastur tuzing.

1. Menyular satridan **form1** va **form2** oynalari hosil qilinadi.



2.28-rasm.

2. **Form1** oynasiga **Standard** komponentalar palitrasidan **MainMenu** tugmasi tanlanadi.
5. Hosil bo'lgan tugma ustiga sichqoncha tugmasi ketma-ket ikki marta bosiladi. Natijada, **Form1 MainMenu1** maydoni paydo bo'ladi.
6. **Object Inspektorining Caption** qatoriga "Form2 ga o'tish" jumlasini kiritiladi.



2.29-rasm.

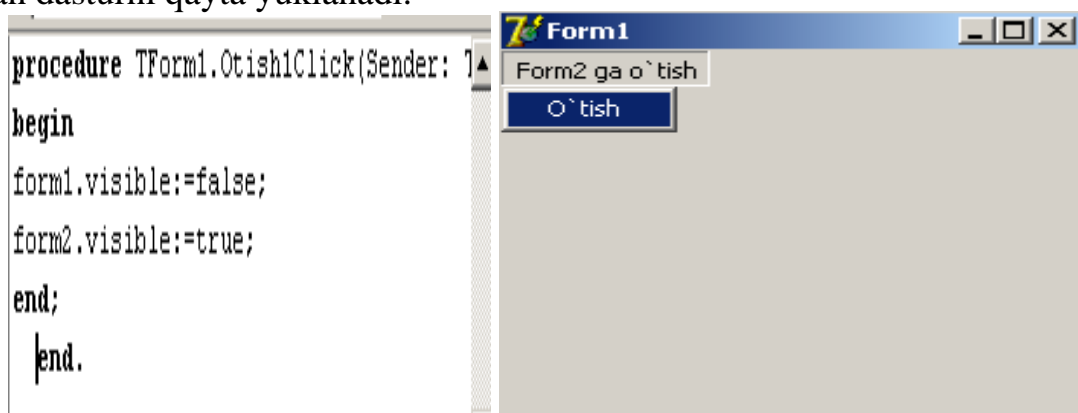
7. Demak, hosil bo'lgan ikkinchi qatorga "O'tish" jumlasini kiritiladi.
8. "O'tish" tugmasini ustiga ketma-ket ikki marta bosiladi.
9. Dasturlash maydoniga quyidagi dastur kodi kiritiladi:

begin

form1.visible:=false;

form2.visible:=true; end;

10. Dasturni biror nom bilan kompyuter xotirasiga saqlanadi va dasturdan chiqib, tuzilgan dasturni qayta yuklanadi.



2.30-rasm.

- form2** dan **form1** ga o'tuvchi dastur tuzish uchu yuqoridagi topshiriqlarni **form2** oynada bajariladi va quyidagi dastur kodi kiritiladi:

begin

```
Form2.visible:=false;  
Form1.visible:=true;  
End;
```

Mustaqil bajarish uchun topshiriqlar

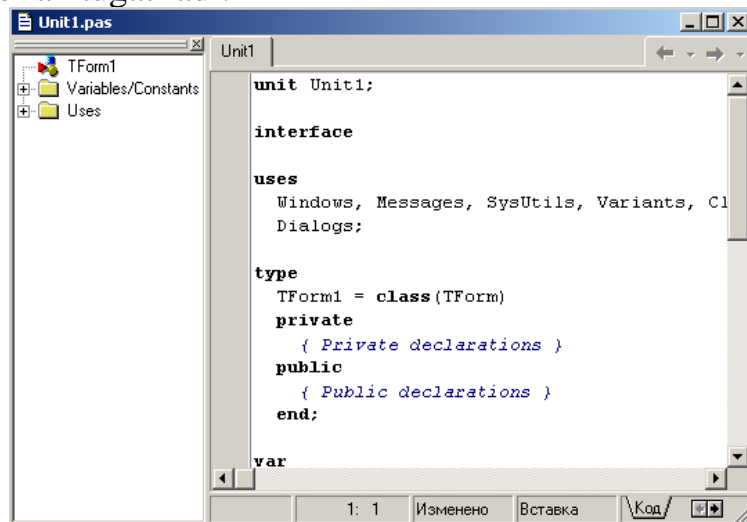
1. Additional komponentalar palitrasidagi tugmalardan foydalanishni o'rganig.
2. Win32 komponentalar palitrasidagi tugmalardan foydalanishni o'rganig.
3. System komponentalar palitrasidagi tugmalardan foydalanishni o'rganig.

III-BOB. DELPHI TILIDA DASTURLASH

3.1. Dasturlash tili elementlari

Delphi dasturlash tilida har qanday dasturni tuzish uchun **Unitn.pas** maydoniga dastur kodi kiritiladi. **Delphi** dasturlash tilida dastur kodini kiritishda **Unitn.pas** maydonidagi buyruqlarni o'chirmasdan dastur kodini kiritishni maslahat beramiz. Dastur kodi **Begin** va **End** buyruqlar orasiga kiritiladi.

Delphi tilida ixtiyoriy masalani dasturini kiritishda hamisha **begin**lar sonidan **end**lar soni bitta ortiqcha bo'ladi. Masalan 3 ta **begin** bo'lsa 4 ta **end** buyrug'i kiritiladi va oxirgi **end** nuqta bilan tugatiladi.



3.1-rasm.

Shunday qilib, **Delphi** dasturlash tilida dastur strukturasi quyidagi ko'rinishda ega:

Var - asosiy dasturda muomulada 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 novbatida **Delphi** dasturlash tilida, tilga xos bo'lgan asosiy tushunchalar asosida hosil qilinadi.

Delphi tili alifbosi.

Tanlab olingan algoritm asosida qo'yilgan masalani kompyuterda yechish uchun har qanday algoritmik tilning o'z alifbosi, buyruqlar majmuasi va maxsus dasturlar jamlangan kutubxonasi bo'lishi zarur. Delphi dasturlash tili alifbosini uch qismga bo'lish mumkin:

- A dan Z gacha bo'lgan lotin alifbosi harflari va bo'sh joyni aniqlovchi belgi ;
- 0 dan 9 gacha bo'lgan raqamlar;
- maxsus belgilar.

Maxsus belgilar o'z novbatida uch turga bo'linadi:

- 1) +, -, *, / - arifmetik amallar, qo'shish, ayirish, ko'paytirish va bo'lish;
- 2) Solishtirish amallari belgilari;

3) Tinish yoki ajratish belgilari., | . | ; | : | (|) | [|] | { | } | ` | := |

3.1-jadval

Delphi belgisi	Matematik ko‘rinishi	Ma’nosi
=	=	teng
<>	≠	teng emas
<	<	kichik
<=	≤	kichik yoki teng
>	>	katta
>=	≥	katta yoki teng

Foydalanuvchi tomonidan bajarilishi lozim bo‘lgan ma’lum harakatni, elektron hisoblash mashinalariga maxsus so‘zlardan tashkil topgan operatorlar yordamida yetkazib, amalga oshirish mumkin. Demak, kompyuter uchun operator bajarilishi so‘zsiz shart bo‘lgan buyruqdir.

Operatorlar algoritmik tillarda asosiy tushuncha bo‘lib, o‘z novbatida ikki guruhga bo‘linadi: oddiy va murakkab operatorlar.

Oddiy operatorlar jumlasiga **shartsiz o‘tish (goto)**, **o‘zlashtirish(:=)** va hakozolar kabi operatorlar kiritilishi mumkin.

Murakkab operatorlar bir necha oddiy operatorlarni o‘z ichiga olishi mumkin.

Delphi dasturlash tilida standart funksiyalar

3.2-jadval

t/r	Matematik ko‘rinishi	Delphi dasturlash tilida
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	$\log_a x$	ln(x)/ln(a)

Matematik ko‘rinishdagi masalalarni Delphi dasturlash tilida yozish.

Boshqa dasturlash tillari kabi Delphi dasturlash tilining ham o‘z tili mavjud. Biror bir masalani dasturini tuzish uchun, bu masalani dastur tiliga o‘tkazishga to‘g‘ri keladi. Bulardan bir nechtasini ko‘rib o‘tamiz:

3.3-jadval

t/n	Matematik ko‘rinishi	Delphi dasturlash tilida yozilishi
1	6,38	6.38
2	$\frac{3}{4}$	3/4
3	a+b	a+b
4	x^2	sqr(x) yoki x*x
5	\sqrt{x}	sqrt(x)
6	a^x	exp(x*ln(a))
7	$\sqrt[x]{a}$	exp((1/x)*ln(a))
8	$\log_a x$	ln(x)/ln(a)
9	$\frac{x+y}{a_1} \cdot \frac{a_2}{x-y}$	(x+y)/a1*a2/(x-y)
10	$\sqrt{x^2+b} - b^2 \sin\left(\frac{x+a}{x}\right)$	Sqrt(sqr(x)+b)-sqr(b)*sin((x+a)/x)

Delphi dasturlash tilida o‘zgaruvchilarni tavsiflash.

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

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

- **real** 2.9e -39 dan 1.7e38 gacha bo‘lgan diapozondagi butun va kasr sonlar ;
- **single** 1.5e -45 dan 3.4e38 gacha bo‘lgan diapozondagi butun va kasr sonlar ;
- **double** 5.0e -324 dan 1.7e308 gacha bo‘lgan diapozondagi butun va kasr sonlar ;
- **extended** 3.4e -4932 dan 1.1e4932 gacha bo‘lgan diapozondagi butun va kasr sonlar ;

Dasturlash tilining kalit so‘zlari

3.4-jadval

1	And	Va
2	Array	Massiv
3	Begin	Boshlash
4	Case	Variant
5	Const	o'zgarmas
6	Div	qoldiqsiz butun bo'lish
7	Do	Bajarish
8	Downto	gacha kamaytirish
9	To	gacha ko'paytirish
10	Else	aks holda
11	End	Tamom
12	File	Fayl
13	For	Uchun
14	Function	Funksiya
15	Goto	ga o'tish
16	If	Agar
17	In	ga tegishli
18	Label	Belgi
19	Mod	butun qoldikli bo'lishdagi qoldiq
20	Procedure	Prosedura

O'zgartirish funksiyalari

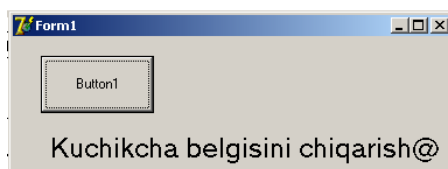
3.5-jadval

1	Chr(n)	n kodli simvol
2	IntToStr(k)	butun k ni tasvirlovchi satr
3	FloatToStr (n)	haqiqiy n ni tasvirlovchi satr
4	FloatToStrF(n, f , k,m)	haqiqiy n ni tasvirlovchi satr, bunda: f – format; k – aniqlik; m – kasr qismidagi raqamlar soni
5	StrToInt (s)	satrni butun songa o'tkazish
6	StrToFloat (s)	satrni haqiqiy songa o'tkazish
7	Round(n)	haqiqiy sonni yaxlitlash
8	Trunc(n)	haqiqiy son kasr qismini olib tashlash
9	Frac(n)	kasrli sonning kasr qismi
10	Int (n)	sonning butun qismi.

Delphi dasturlash tilida belgilarning ishlatilishi.

Bizga ma'lumki dastur tuzish jarayonida dasturga ayrim belgilarni ishlatishga to'g'ri keladi. Delphi dasturlash tilida belgilarni dasturga chiqarish uchun, #-belgisi qo'yilib sonlardan foydalaniladi. Misol tariqasida @- belgisini dasturga chiqarish uchun 64 sonidan foydalanamiz. Bu quyidagicha:

“label1.caption:='Kuchukcha belgisini chiqarish'+#64;



3.2-rasm.

Belgi va harflarni tasniflash

3.6-jadval

Simvol	Belgi	Simvol	Belgi	Simvol	Belgi	Simvol	Belgi
9	Abzats tashlash	11	Yangi qator tashlash	13	Abzats oxiri		
32	Bo'sh joy	42	*	52	4	62	>
33	!	43	+	53	5	63	7
34	k	44	,	54	6	64	@
35	#	45	-	55	7	65	A
36	\$	46	,	56	8	66	V
37	%	47	/	57	9	67	S
38	&	48	0	58	:	68	D
39	'	49	1	59	;	69	Ye
40	(50	2	60	<	70	F
41)	51	3	61	=	71	G
72	N	87	W	101	e	114	r
73	I	88	X	102	f	115	S
74	J	89	Y	103	g	116	t
75	K	90	Z	104	h	117	u
76	L	91	[105	i	118	V
77	M	92	\	105	i	119	W
78	N	93]	106	j	120	X
79	O	94	^	107	k	121	U
80	R	95	_	108	I	122	Z
81	Q	96	'	109	m	123	[
82	R	97	a	110	n	124	
83	S	98	b	111	o	125]
84	T	99	s	112	P	126	~
85	U	100	d	113	q	127	
192	A	201	Y	210	T	219	Ы
193	B	202	K	211	U	220	
194	V	203	L	212	F	221	E
195	G	204	M	213	X	222	Yu
196	D	205	N	214	S	223	Ya
197	Ye	206	O	215	Ch	224	A
198	J	207	P	216	Sh	225	b
199	3	208	R	217	ИЦ	226	v
200	I	209	S	218	'	227	g
228	d	235	l	242	t	249	щ

229	e	236	m	243	u	250	‘
230	j	237	n	244	f	251	Ы
231	3	238	o	245	x	252	
232	i	239	p	246	ts	253	e
233	y	240	r	247	ch	254	yu
234	k	241	s	248	sh	255	ya

Mustaqil bajarish uchun topshiriqlar

Quyidagi ifodalarni Delphi dasturlash tilida yozing:

1. 6,38; 2) $\frac{ab}{c} + \frac{c}{ab}$; 3. $\frac{x+y}{a_1} \frac{a_2}{x-y}$; 4. $(1+x)^2$; 5. $\sqrt{1+x}$; 6. $|a+bx|$; 7. $\sin 8$; 8.

$\cos^2 x^8$; 9. $\operatorname{tg} x$; 10. $\log_2 \frac{x}{5}$; 11. x^{-1} ; 12. x^4 ; 13. x^{-2} ; 14. x^{100} ; 15. 2^{1+x} ; 16. $x^{\sqrt{2}}$; 17.

$\sqrt[3]{1+x}$; 18. $\sqrt[8]{x^8 + 8^x}$; 19. $\frac{xyz - 3,3|x + \sqrt[4]{y}|}{10^7 + \sqrt{\lg 4!}}$; 20. $\frac{-\beta + \sin^2 \pi^4}{\cos 2 + |\operatorname{ctg} y|}$;

20.

21.

22.

$$2 + \frac{x^2}{\sqrt{2}} + \frac{|y^3|}{\sqrt{2}} + \frac{z^4(\ln(x)+1) \cdot \sqrt{2}}{\sqrt{3}}; \quad \ln\left|(y - \sqrt{|x|}) \cdot \left(x - \frac{y}{z + x^2/4}\right)\right|; \quad \frac{2 \cos\left(x - \frac{\pi}{6}\right)}{\frac{1}{2} + \sin^2 y};$$

23.

24.

25.

$$\frac{a^2 x + e^{-x} \cos bx}{bx - e^{-x} \sin bx + 1}; \quad e^{-ax} \frac{x + \sqrt{a+x}}{x - \sqrt{|x-b|}}; \quad \ln(y^{-\sqrt{|x|}}) \cdot (\sin(x) + e^{(x+y)});$$

3.2. Chiziqli jarayonlarni dasturlash

Chiziqli algoritmlarda asosan hech qanday shart tekshirilmaydi va jarayonlar tartib bilan ketma-ket bajariladi. Demak, chiziqli algoritmlar sodda hisoblashlar yoki amallar ketma-ketligidir. 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.

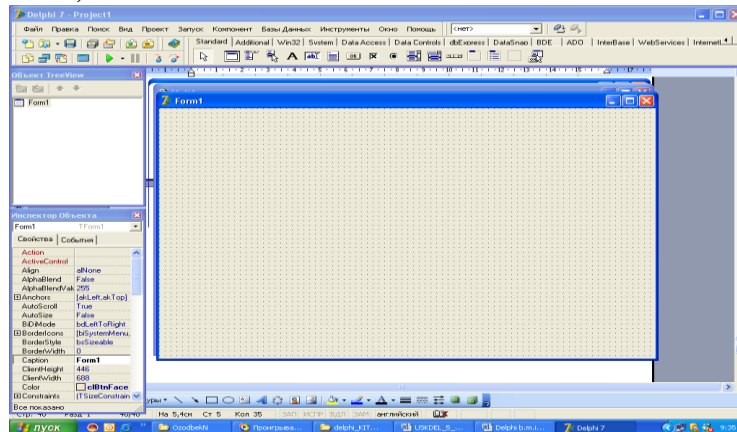
Chiziqli tuzilishdagi algoritmnini tuzish masalani yechish uchun kerak bo‘ladigan boshlang‘ich ma’lumotlarni tashkil qiluvchi o‘zgaruvchilar nomi, ularning turi va o‘zgarish oralig‘ini aniqlashdan boshlanadi. Boshlang‘ich ma’lumotlar aniqlangandan so‘ng, oraliq va yakuniy natijalar, o‘zgaruvchilarining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

1–misol. Ikki sonning yig‘indisini hisoblash dasturi.

Berilgan masalani **Delphi** dasturlash tilida dasturlash uchun, **Delphi** dasturini yuklab yaratiladigan dasturni loyihasini tuzib olish kerak, aniqrog‘i formaga kerakli tugmalar, menyular, yozuvlar va shunga o‘xshash tugmalarni joylashtirib olgach, shu yaratilgan tugmalarni har biriga tegishli buyruqlarni kiritish zarur.

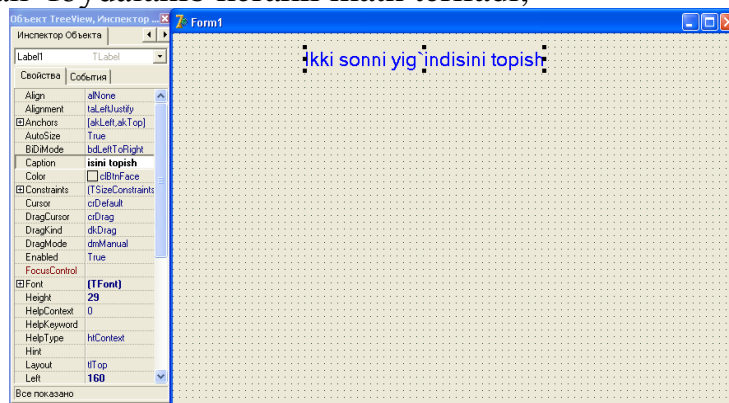
Keltirilgan masalani dasturini ifodalash uchun, **Delphi** dasturlash tilida quyidagi bosqichlar amalga oshiriladi:

1. Dastur ishga tushiriladi;



3.3-rasm.

2. Hosil bo'lgan oynada matni ekranga joylash uchun **Delphi** dasturlash tilining **Standart** komponentalar palitrasidan **A** piktogrammasi belgilanib, forma oynasi ustiga kelinadi va sichqoncha tugmachasini bosgan holda matn joylashtirilishi lozim bo'lgan joy ajratiladi. Natijada, **Label1** matn maydoni hosil qilinadi hamda parametrlar va xossalari bo'limidan foydalanib kerakli matn teriladi;



3.4-rasm.

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 matni 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. **Edit** matn kiritish qatorini ekranga joylash uchun **Delphi**ning **Standart** komponentalar palitrasidan **ab** piktogrammasi belgilanib, forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda matn kiritilishi lozim bo'lgan joy ajratiladi. Natijada, **Edit1** matn kiritish maydoni hosil qilinadi. Matni kiritish dastur ishchi holatiga o'tilganda bajariladi.



3.5-rasm.

Edit2 matn kiritish maydoni ham xuddi shu tartibda bajariladi.



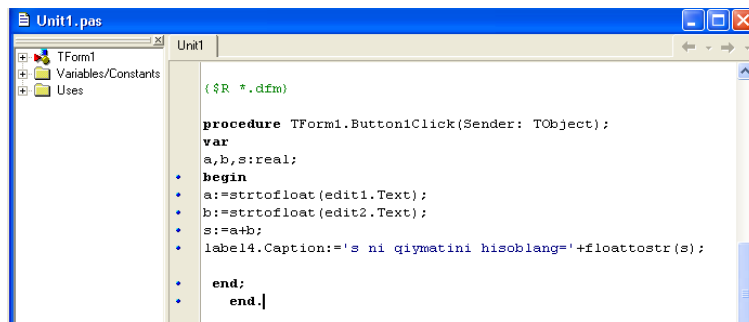
3.6-rasm.

5. **Button** tugmasini ekranga joylash uchun **Delphining Standart** komponentalar palitrasidan palitrasidan **OK** piktogrammasi belgilanib, forma ustiga kelinadi va sichqoncha tugmachasini bosgan holda tugma qo'yilishi lozim bo'lgan joy ajratiladi. Natijada, **Button1** tugmasi hosil qilinadi. Tugmaning nomi **Caption** xossasidan foydalanib o'zgartiriladi.



3.7-rasm.

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



3.8-rasm.

7. **Button1** (Hisoblash) tugmasining dasturlash maydochasiga quyidagi dastur kodi kiritiladi:

var

a,b,s:real;

begin

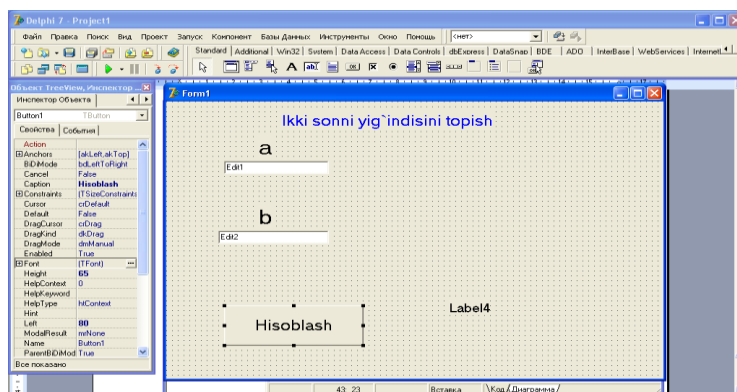
a:=strtofloat(edit1.Text);

b:=strtofloat(edit2.Text);

s:=a+b;

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

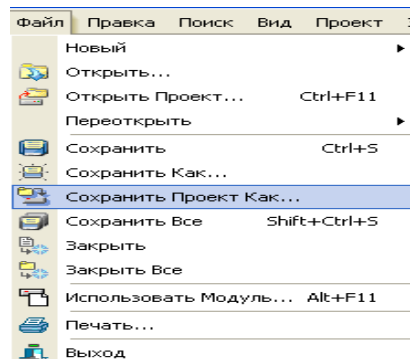
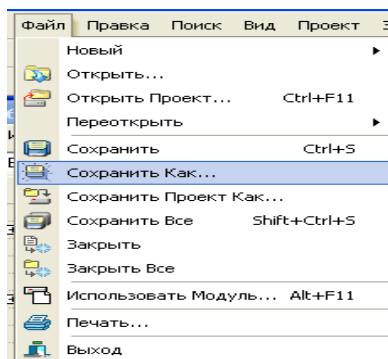
8. Dasturlash maydoniga, dastur kodini kiritib bo'lgandan so'ng, **F12** tugmasini tanlanadi.



3.9-rasm.

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

ikkinchi marta



3.10-rasm.

10. Buyruqlari yordamida kompyuter xotirasiga saqlanadi. Yuqoridagi ketma-ketlikdan so'ng, dasturni kompilatsiya qilishimiz kerak(dasturni ishchi holatga keltirish). Kompilatsiya ishini asboblardagi quyidagi:



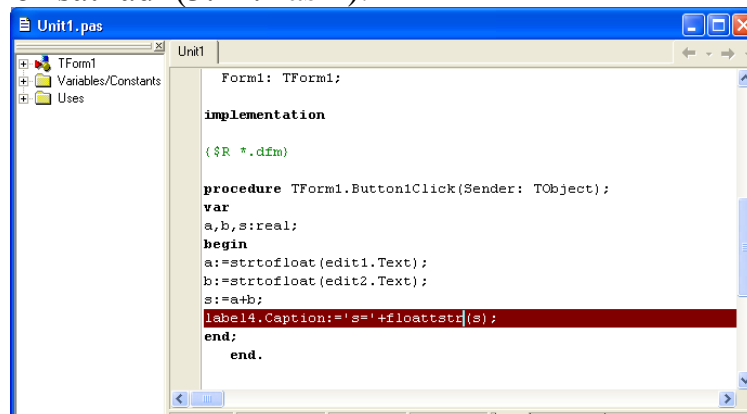
tugma yordamida amalga oshiriladi.

Dastur faollashgandan so‘ng, oynaning ko‘rinish holati quyidagicha bo‘ladi (3.11-rasm):



3.11-rasm.

Agar dasturning biror joyida sintaksis xatolik bo‘lsa, quyidagi ko‘rinishga o‘xshash xatolik ko‘rsatiladi (3.12.-rasm):



3.12-rasm.

2-misol. Bitta oynada ikki sonning yig‘indisini, ayirmasini, ko‘paytmasini va bo‘linmasini hisoblash dastur tuzish.

Formani loyihalashtirish oynasini yuqoridagi masaladan foydalangan holda, **Label1, Label2, Label3, Label4, Edit1, Edit2, Button1, Button2, Button3, Button4** tugmalarni hosil qilinadi, parametrlar va hodisalar bo‘limidan tugmalarga kerakli ma‘lumotlar kiritiladi (Masalan, **Button1** tugmasiga yig‘indi. va h.k.) ..“Yig‘indi” belgisi yozilgan tugmaning dasturlash maydoniga quyidagi:

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

“Ayirma” belgisi yozilgan tugmaning dasturlash maydoniga quyidagi:

```
var
```

```

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

```

“Ko‘paytma” belgisi yozilgan tugmaning dasturlash maydoniga quyidagi:

```

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

```

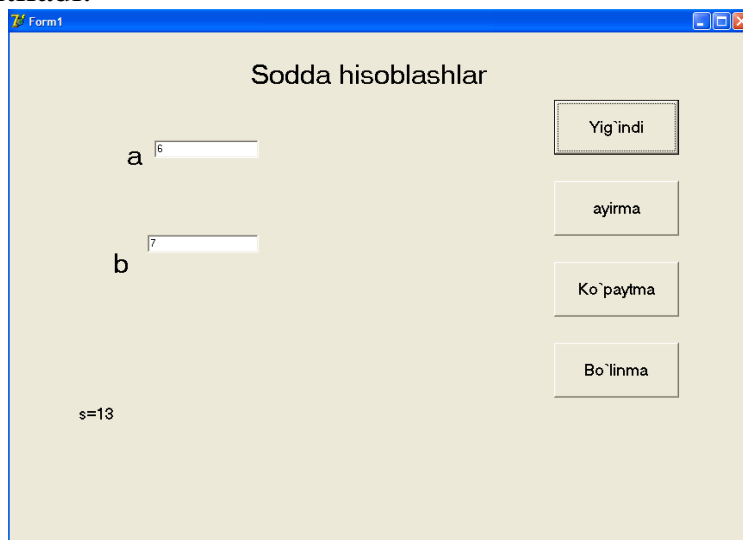
“Bo‘linma” belgisi yozilgan tugmaning dasturlash maydoniga quyidagi:

```

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

```

dastur kodlari kiritiladi.



3.13-rasm.

3- misol. Bitta oynada konusning yon sirti, to‘la sirti va hajmini hisoblash dasturini hosil qilish.

$$1. S_{yon} = \pi Rl;$$

$$2. S_{to'la} = S_{asos} + S_{yon} = \pi R(R + l);$$

$$3. V = \frac{1}{3} \pi R^2 h.$$

Standart komponentalar palitrasidan **Edit1**, **Edit2**, **Edit3**, **Button1**, **Button2**, **Button3** tugmalari hosil qilinadi va **Button1** tugmasining dasturlash maydoniga quyidagi:

```
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;
```

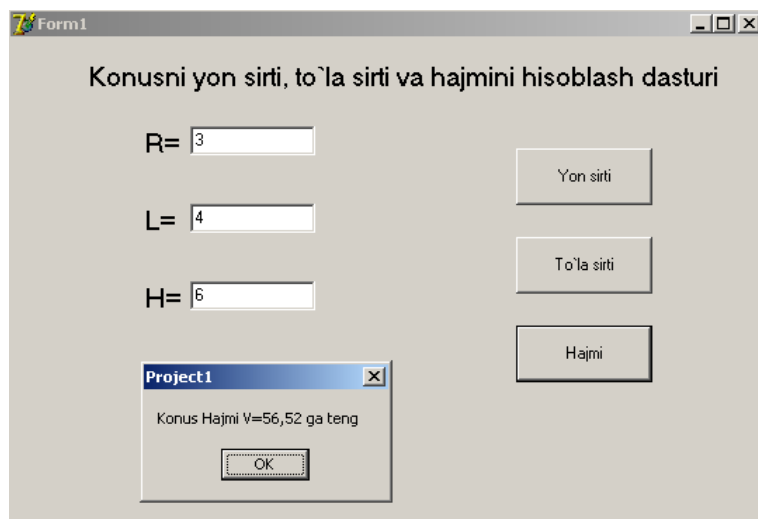
Button2 tugmasining dasturlash maydoniga quyidagi:

```
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;
```

Button3 tugmasining dasturlash maydoniga quyidagi:

```
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;
```

dastur kodlari kiritiladi.



3.14-rasm.

Mustaqil bajarish uchun topshiriqlar

1. Koordinatalar tekisligida ikkita nuqta orasidagi masofani aniqlovchi dastur tuzing.
2. To'rtta son berilgan. Shu sonlarning o'rtta arifmetigi va o'rtta geometrigini hisoblovchi dastur tuzing.
3. Uchburchakning a,b,c tomonlari berilgan. Shu uchburchakka ichki chizilgan aylanani radiusini hisoblovchi dastur tuzing:

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

bunda r – uchburchakning yarim perimetri.

4. Uchburchakning berilgan uchta burchaklari va tomoni orqali uning yuzini hisoblash dasturini tuzing:

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

5. Rombning tomoni va o'tkir burchagi berilgan, uning yuzasini hisoblash dasturini tuzing: $S = a^2 \sin \alpha$, bunda a – rombning tomoni, α -esa o'tkir burchagi.
6. Arifmetik progressiyaning birinchi hadi va ayirmasi berilgan. Uning 30-chi hadi va dastlabki 40 ta hadining yig'indisini hisoblash dasturini tuzing.
7. Kesmaning koordinatalari berilgan. Shu kesmani o'rtasining koordinatalarini hisoblash dasturini tuzing.
8. 1 dan m gacha natural sonlar berilgan. Shu sonlarning arifmetik progressiya yordamida yig'indisini hisoblash dasturini tuzing.

Quyidagi misollarni hisoblash dasturini tuzing:

$$9. a = \ln(y^{-\sqrt{|x|}}) * (\sin(x) + e^{(x+y)}).$$

$$10. b = \sqrt{c (\sqrt{y} + x^2)} \cdot (\cos(x) - |c - y|).$$

$$11. c = \arctg(x) - \frac{3}{5} e^{xy} + 0.5 \frac{|x + y|}{(x + y)^b}.$$

$$12. d = \frac{e^{|x-y|} \cdot \text{tg}(z)}{\text{arctg}(y) + \sqrt{x}} + \ln(x).$$

$$13. e = \frac{(\cos(x) - \sin(y))^3}{\sqrt{\text{tg}(z)}} + \ln^2(x * y * z).$$

$$14. f = y^x + \sqrt{|x| + e^y} - \frac{z^3 \cdot \sin^2(y)}{y + z^2 / (y - x)}.$$

$$15. g = \frac{1 + \cos(x + y)}{|e^x - 2y / (1 + x^2 * y^2)|} * x^3 + \arcsin(y).$$

3.3. Tarmoqlanuvchi jarayonlarni dasturlash

Delphi dasturlash tilida tarmoqlanish **If** operatori bilan amalga oshiriladi. **If** operatori- bu shartli operator hisoblanadi. Delphi 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:

3.7-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) AND (6<50) -mantiqiy ifoda rost (True),
 (20>0) OR (20<0.5) -mantiqiy ifoda rost (True),
 (10<8) AND (10<15) -mantiqiy ifoda yolg‘on (False),
 NOT(100>3) -mantiqiy ifoda yolg‘on (False).

O‘zlashtirish mantiqiy ifodalarni biror bir mantiqiy o‘zgaruvchiga yuborish ham mumkin. Masalan: F:=(A<B) AND (A<C);

Bu yerda, ikkala shart bajarilgandagina F mantiqiy o‘zgaruvchi “chin” (True) qiymatni qabul qiladi. Aks holda, “yolg‘on ” (False) qiymatni qabul qiladi. Delphi dasturlash 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 bo'lganda bajariladigan operatorlar>  
end;
```

Bu yerda If -agar; then -u holda; else -aks holda ma'nosini bildiruvchi xizmatchi (kalit) so'zlar. Birinchi ko'rinishdagi shartli operatorida, 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 ekanligi 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 ekanligi kelib chiqadi.

Yuqorida bayon qilingan fikrlarni tekshirib ko'rish uchun **Delphi** dasturlash tilida shartli operatorlaridan foydalangan holda bir nechta misollarga dasturlar tuzamiz.

1-misol. Berilgan ikki sonlarni bo'linmasini hisoblash dasturini tuzing.

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

Standart komponentalar palitrasidan **Edit1**, **Edit2**, **Button1**, **Label1**, **Label2**, **Label3**, **Label4** tugmalari hosil qilinadi va **Button1** tugmasining dasturlash maydoniga quyidagi:

```
var a,b,c:real;
```

```
begin
```

```
a:=strtofloat(edit1.text);
```

```
b:=strtofloat(edit2.text);
```

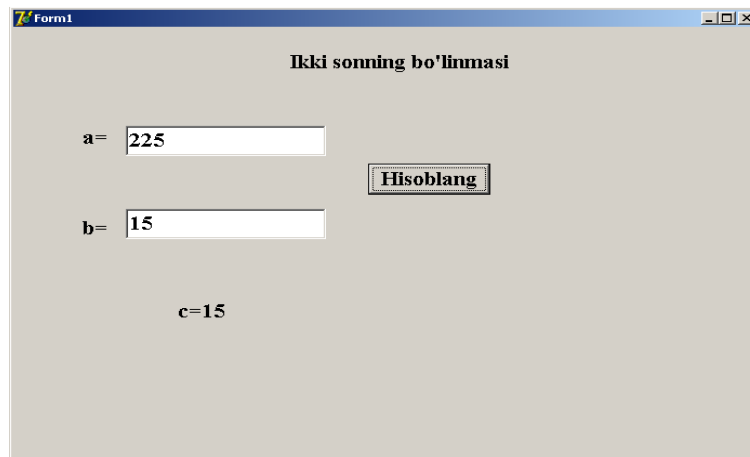
```
if b=0 then label5.Caption:='Bo'luvchi nolga teng'
```

```
else begin c:=a/b;
```

```
label5.Caption:='' + floattostr(c);
```

```
end; end;
```

dastur kodi kiritiladi.



3.15- rasm.

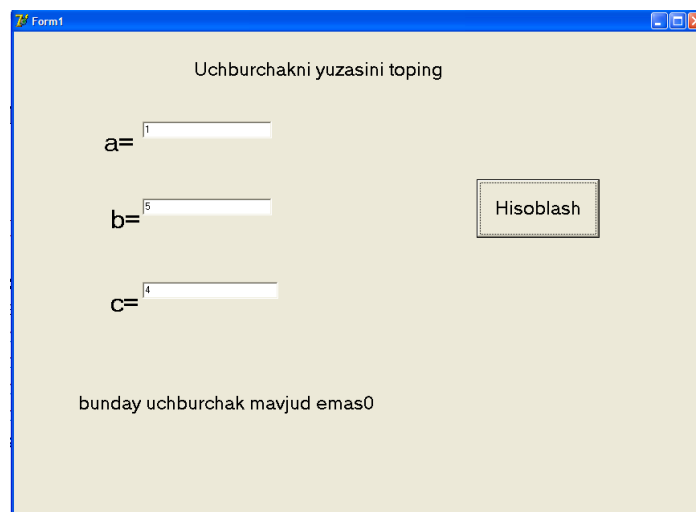
2- misol. Geron formulasidan foydalanib uchburchakning yuzasini hisoblash dasturini tuzing. $S = \sqrt{p(p-a)(p-b)(p-c)}$

Standart komponentalar palitrasidan **Button1**, **Label1**, **Label2**, **Label3**, **Label4**, **Label5**, **Edit1**, **Edit2**, **Edit3** tugmalari hosil qilinadi va **Button1** tugmasining dasturlash maydoniga quyidagi:

```

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 mavjud emas'+floattostr(s)
Else begin
if a+c<=b then
label5.Caption:='bunday uchburchak mavjud emas'+floattostr(s)
else begin
if b+c<=a then
label5.Caption:='bunday uchburchak mavjud 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;
dastur kodi kiritiladi.

```

3.17-rasm.

3-misol. Kvadrat tenglamani yechish dasturi.

Standart komponentalar palitrasidan **Button1**, **Button2**, **Label1**, **Label2**, **Label3**, **Label4**, **Label5**, **Edit1**, **Edit2**, **Edit3** tugmalari hosil qilinadi va **Button1** tugmasining dasturlash maydoniga quyidagi:

var

a,b,c,d,x1,x2:real;

begin

a:=strtofloat(edit1.Text);

b:=strtofloat(edit2.Text);

c:=strtofloat(edit3.Text);

if a=0 then label1.Caption:='kvadrat tenglama emas'

else begin

*d:=b*b-4*a*c;*

if d<0 then label1.Caption:='kvadrat tenglama yechimga ega emas'

else begin

*if d=0 then label1.Caption:='x1='+floattostr(-b/2*a)*

else

*label1.Caption:='x1='+floattostr((-b+sqrt(d))/2*a)+' '+x2='+floattostr((-b-
sqrt(d))/2*a);*

end;

end;

Button2 tugmasining dasturlash maydoniga quyidagi:

begin

edit1.Text:=' ';

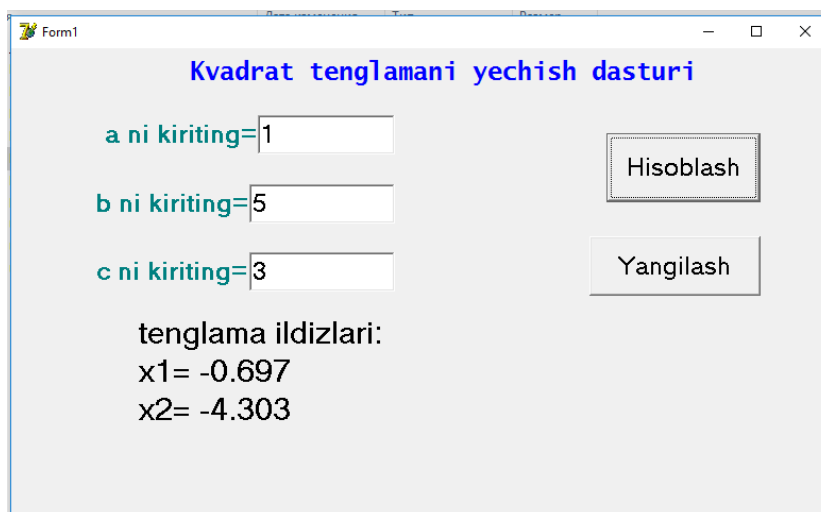
edit2.Text:=' ';

edit3.Text:=' ';

label2.Caption:=' ';

end;

dastur kodlari kiritiladi.



3.18-rasm.

4- misol. Ushbu funksiya qiymatini hisoblash dasturini tuzing.

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

Standart komponentalar palitrasidan **Button1**, **Label1**, **Label2**, **Edit1** tugmalarini hosil qilamiz va **Button1** tugmasining dasturlash maydoniga quyidagi *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);

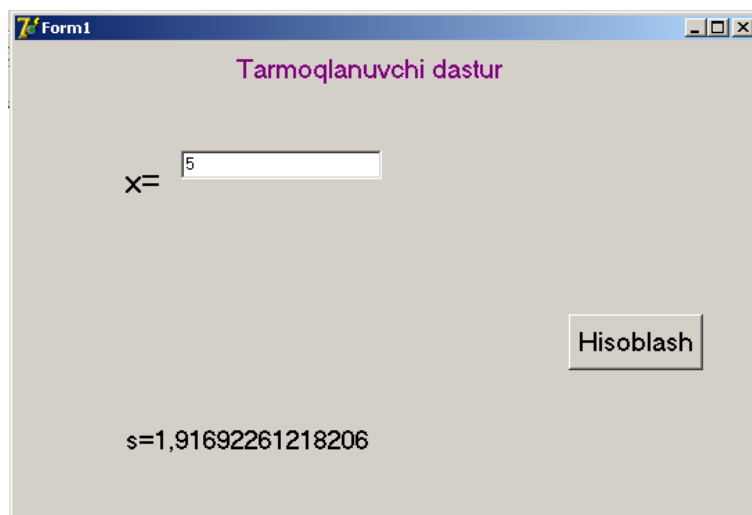
goto 3;

2:s:=ln(x+1.8);

3:label2.Caption:= 's=' + floattostr(s);

end;

dastur kodini kiritamiz.



3.19-rasm.

Agar yuqoridagi hisoblash natijasini uch xonagacha yaxlitlab chiqarish talab etilsa quyidagi buyruq beriladi: `label2.Caption:='s='+floattostrF(s, ffixed, 4, 3);` ko'rinishda yoziladi.

Mustaqil bajarish uchun topshiriqlar

1. a, b va c haqiqiy sonlar berilgan bo'lsin. $a < b < c$ munosabat o'rinlimi?
2. Berilgan ikkita sondan eng kattasini topuvchi dastur tuzing.
3. $y = \frac{\sqrt{x}}{b}$ ni hisoblovchi dastur tuzing.
4. Berilgan uchta sondan eng kattasini topuvchi dastur tuzing.
5.
$$y = \begin{cases} \sin^2 x & \text{agar } |x| < \frac{\pi}{4}, \\ \sin(\text{tg}x), & \text{agar } |x| = \frac{\pi}{4}. \end{cases}$$
6.
$$y = \begin{cases} \sqrt[3]{x^2 + a^2}, & \text{agar } x^2 + a^2 < 4, \\ x + a, & \text{agar } x^2 + a^2 = 4, \\ \ln x, & \text{agar } x^2 + a^2 > 4. \end{cases}$$
7. Uchta a, b va c haqiqiy sonlar berilgan bo'lsin. Tomonlari shu sonlarga teng uchburchak mavjudmi? Mavjud bo'lsa, uning perimetri va yuzi topilsin.
8. Uchta o'zaro har xil sonlarning yig'indisi birdan kichik bo'lsa, berilgan sonlarning eng kichigi, aks holda eng kattasi topilsin.
9. M(x,y) nuqtaning markazi va raduisi berilgan aylanaga tegishlilikini aniqlovchi dastur yarating.
10. Agar 3 ta har xil x, y, z butun sonlar yig'indisi 1 dan kichik bo'lsa, u holda bu uchta sondan eng kichigini qolgan ikkitasining yarim yig'indilari bilan almashtiring aks holda x va y lardan kichigini qolgan ikkitasi yarim yig'indilari kvadrati bilan almashtiruvchi dastur tuzing.
11. $\min(x, y)$ ni topuvchi dastur tuzing.
12. $\max(x, y) + \min(x, y)$ ni topuvchi dastur tuzing.
13. $\max(x, y, z)$ ni topuvchi dastur tuzing.

14. $\max(x+y+z, xyz)$ ni topuvchi dastur tuzing.
 15. $\min(x+y/2+z/3, x-2y+z, x-y-z)$ ni topuvchi dastur tuzing.

3.4. Takrorlanuvchi jarayonlarni dasturlash

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. Delphi dasturlash 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 $i:=m_1$ to m_2 do S;

Bu yerda i -sikl parametri; m_1, m_2 -i parametrining boshlang'ich va oxirgi qiymati bo'lib, ular o'zgarmas son yoki ifoda bo'lishi mumkin; S-sikl tanasi bo'lib, bir nechta operatorlardan tashkil topishi mumkin. Agar sikl tanasi bir necha operatoridan iborat bo'lsa, ular **begin** va **end** ichiga olinadi.

1-misol. Delphi dasturlash tilida 1 dan n gacha ketma-ket butun natural sonlarni yig'indisini hisoblaydigan dastur tuzing.

```
var i,n,s: integer;
    begin
        s:=0;
        For i:=1 to n do s:=s+i;
    labeln.Caption:= 's=' + floatostr(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 $i:=m_1$ downto m_2 do S;

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 yerda 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.

2-misol. 1,2,...,100 sonlar yig'indisini hisoblash dasturini tuzing.

```
Const kn=100;
Var i: Integer; s: Real;
begin
    s:=0; i:=0;
    While i<=kn do begin i:=i+1; s:=s+i;
```

```
end;
labeln.Caption:='s'+floattostr(s);
end;
```

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

```
Repeat
  S  Until B
```

Bu yerda 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.

3-Misol. 1,2,...,100 sonlar yig'indisini hisoblash dastursini tuzing.

```
Const kn=100;
Var i: Integer; s: Real;
Begin
  s:=0; i:=0;
  Repeat
    i:=i+1; s:=s+i;
  Until I>kn;
  labeln.Caption:='s'+floattostr(s);
end;
```

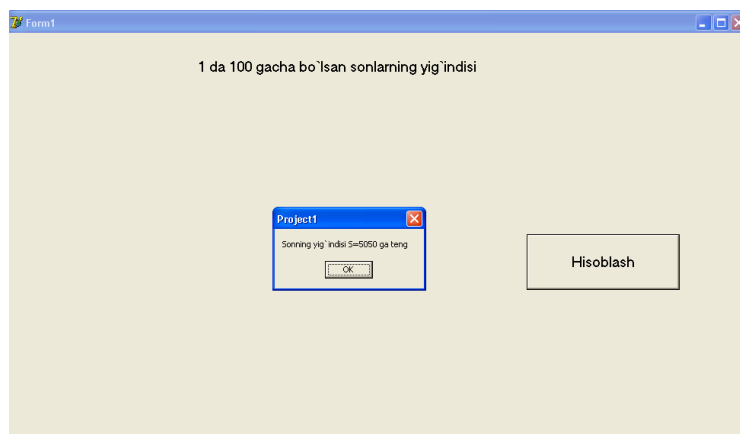
Odatda WHILE operatori REPEAT operatoriga nisbatan ko'p ishlatiladi. Bunga sabab ko'pchilik masalalarda sikl tugallanish sharti sikl boshlanmasdan tekshirish maqsadga muvofiqdir. Zarur bo'lsa siklni umuman bajarmasdan o'tish mumkin. Ko'pchilik masalalarni yechishda tuzilgan dasturda ichma-ich joylashgan sikllar tashkil etishga to'g'ri keladi. Bunday sikllarga murakkab sikllar deyiladi.

Yuqorida bayon qilingan fikirlarni tekshirib ko'rish uchun Delphi tilining vizvual loyihasida, sikl operatorlaridan foydalangan holda bir nechta dasturlar tuzamiz.

4-misol. 1 dan 100 gach ketma- ket keluvchi natural sonlarning yig'insisini hisoblash dasturi.

Standart komponentalar palitrasidan **Button1**, **Label1** tugmalarini hosil qilamiz va **Button1** tugmasining dasturlash maydoniga quyidagi:

```
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;
dastur kodi kiritiladi.
```



3.20-rasm.

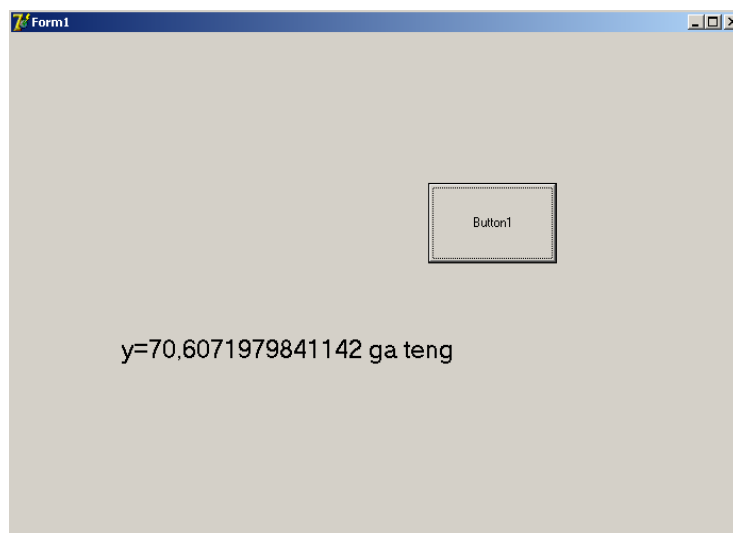
1 dan 100 gacha ketma- ket keluvchi natural sonlarning o'zini chiqarish dasturi quyidagicha bo'ladi:

```
var
i,s:integer;
begin
s:=0;
for i:=1 to 100 do
label1.Caption:= label1.Caption+inttostr(i)+#13; end;
```

5-misol. Ushbu $y = \prod_{i=1}^6 \sum_{k=1}^5 \frac{k+i}{2k+3i}$ jarayonni mos oyna va dastur modulini tuzing.

Standart komponentalar palitrasidan **Button1**, **Label1** tugmalarini hosil qilamiz va **Button1** tugmasining dasturlash maydoniga quyidagi:

```
Var
k,i:integer;
y,p:real;
begin
y:=1;
for i:=1 to 6 do begin
p:=0;
for k:=1 to 5 do begin
p:=p+(k+i)/(2*k+3*i);
end;
y:=y*p;
Label1.caption:='y='+floattostr(y)+' ga teng';
end;
end;
dastur kodini kiritamiz.
```



3.21-rasm.

6-misol. 1 dan 20 gacha ketma- ket keluvchi natural sonlarning kvadratini hisoblash dasturini tuzing.

Standart komponentalar palitrasidan **Button1**, **Button2**, **ListBox1** tugmalarini hosil qilamiz va **Button1** tugmasining dasturlash maydoniga quyidagi:

var

r:array of integer;

i:Integer;

begin

ListBox1.Items.Clear;

SetLength(r,10);

for i:=0 to High(r) do begin

*r[i]:=i*i;*

ListBox1.Items.Add(IntToStr(i)+' ni kvadrati =' +IntToStr(r[i]));

end;

2. “Button2” tugmasining dasturlash maydoniga quyidagi:

type

TDynArr=array of integer;

var

r:TDynArr;

i:Integer;

begin

ListBox1.Items.Clear;

SetLength(r,10);

*for i:=0 to High(r) do r[i]:=i*i;*

SetLength(r,20);

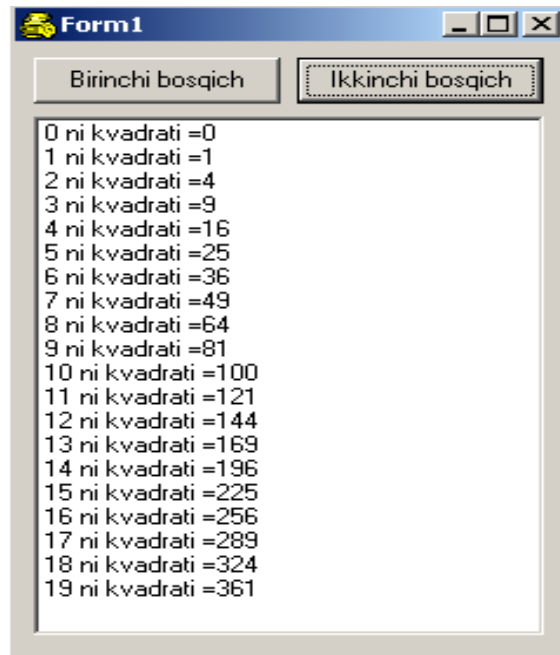
*for i:=10 to High(r) do r[i]:=i*i;*

for i:=0 to High(r) do

ListBox1.Items.Add(IntToStr(i)+' ni kvadrati =' +IntToStr(r[i]));

end;

dastur kodini kiritamiz.

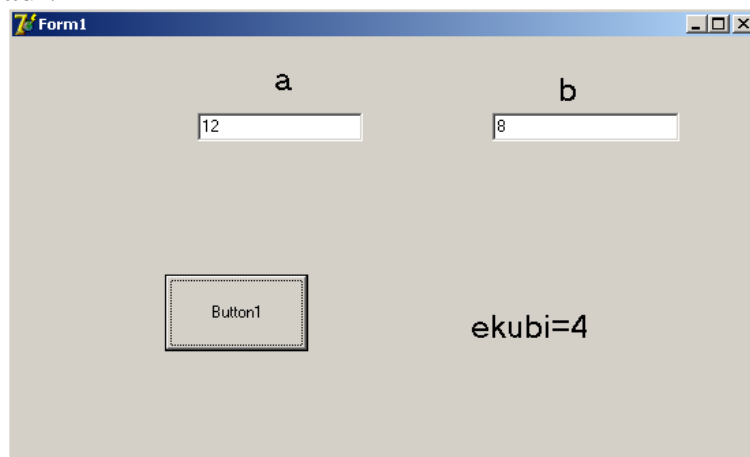


3.22-rasm.

7-masala. Berilgan ikkita sonni EKUB(eng katta umumiy bo‘luvchi) ni topish dasturi.

Standart komponentalar palitrasidan **Edit1, Edit2, Button1, Label1, Label2, Label3** tugmalari hosil qilinadi **Button1** tugmasining dasturlash maydoniga quyidagi:

```
var
a,b,c,d:real;
begin
a:=strtofloat(edit1.text);
b:=strtofloat(edit2.text);
while a<>b do
begin
if a>b then a:=a-b else b:=b-a;
label3.Caption:='ekubi'+floattostr(a);
end;
end;
dastur kodi kiritiladi.
```



3.16-rasm.

8-masala. O‘nlik sanoq sistemasidagi sonni ikkilik, sakkizlik, o‘n oltilik sanoq sistemalariga o‘tkazish.

Standart komponentalar palitrasidan **Button1**, **Label1**, **Label2**, **Edit1** tugmalari hosil qilinadi va **Button1** tugmasining dasturlash maydoniga quyidagi:
Begin

```
E:= strtoint(edit1.text);edit2.text:="";  
If E>1 then  
Begin I:=0;  
Repeat  
I:=I+1;m:=E mod 2;b[I]:=m;E:=E div 2;  
If E<2 then begin I:=I+1;b[I]:=E;end;  
Until E<2;  
For j:=i downto 1 do edit2.text:=edit2.text+inttostr(b[j]);  
End else edit2.text:=inttostr(E);
```

end;

4. **Button2** tugmasining dasturlash maydoniga quyidagi:
begin

```
E:= strtoint(edit1.text);edit2.text:="";  
If E>=8 then  
Begin I:=0;  
Repeat  
I:=I+1;m:=E mod 8;b[I]:=m;E:=E div 8;  
If E<8 then begin I:=I+1;b[I]:=E;end;  
Until E<8;  
For j:=i downto 1 do edit2.text:=edit2.text+inttostr(b[j]);  
End else edit2.text:=inttostr(E);
```

end;

5. “**Button3**” tugmasining dasturlash maydoniga quyidagi:
begin

```
E:= strtoint(edit1.text);edit2.text:="";  
If E>=10 then  
Begin  
k:=E div 10;l:=E-k+10;  
If l>=6 then  
begin I:=0;  
Repeat  
I:=I+1;m:=E mod 16;b[I]:=m;E:=E div 16;  
If E<16 then begin I:=I+1;b[I]:=E;end;  
Until E<16;  
For j:=i downto 1 do edit2.text:=edit2.text+inttostr(b[j]);  
end;  
S:=inttostr(k);edit2.text:=s+a[l];  
End else edit2.text:=inttostr(E);
```

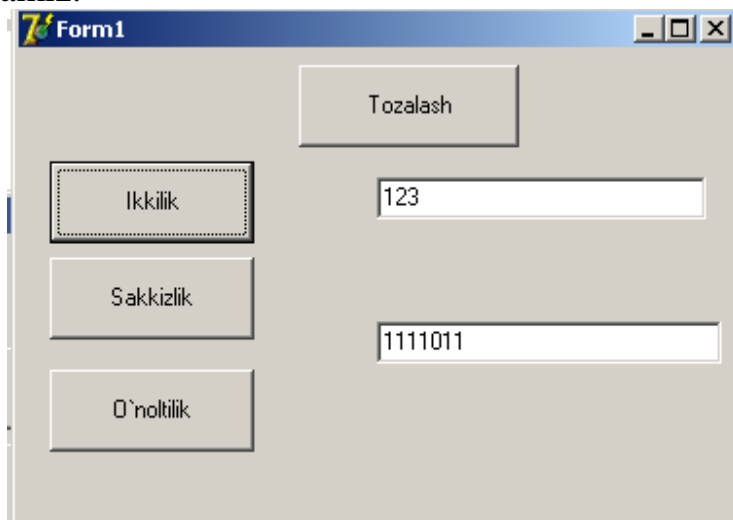
end;

6. **Button4** tugmasining dasturlash maydoniga quyidagi:
begin
Edit1.clear;

```

Edit2.clear;
end;
dastur kodini kiritamiz.

```



3.23-rasm.

9-misol. a^k ning oxirgi raqamini toppish dasturini tuzish.

Standart komponentalar palitrasidan **Button1**, **Label1**, **Label2**, **Label3**, **Label4**, **Edit1**, **Edit2** tugmalarini hosil qilinadi va **Button1** tugmasining dasturlash maydoniga quyidagi kdlar kiritiladi:

```

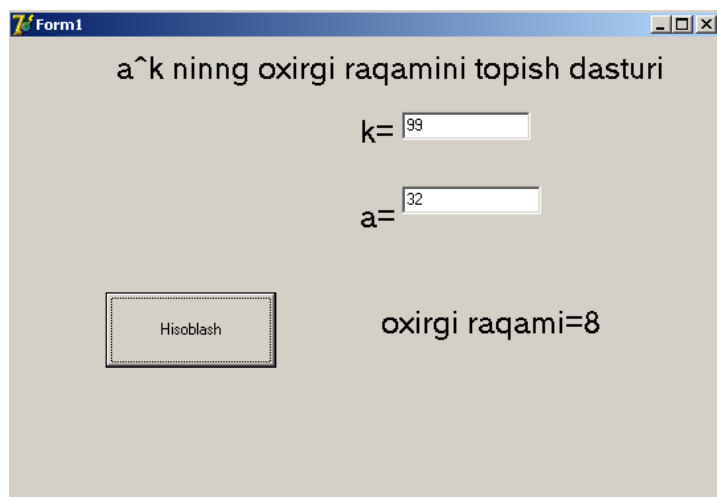
Var
g,a,b,c,d,e:Real;
begin
a:=strtofloat(Edit1.text);
g:=strtofloat(Edit2.text);
e:=frac(g/10)*10;
if e=2 then begin
b:=a/4;
c:=frac(b)*4;
if c=0 then d:=6 else
if c=1 then d:=2 else
if c=2 then d:=4 else d:=8;
Label2.Caption:='oxirgi raqami='+floattostr(d);
end;
if e=3 then begin
b:=a/4;
c:=frac(b)*4;
if c=1 then d:=3 else
if c=2 then d:=9 else
if c=3 then d:=7 else d:=1;
Label2.Caption:='oxirgi raqami='+floattostr(d);
End;
if e=4 then begin
b:=a/2;
c:=frac(b)*2;

```

```

if c=0 then d:=6 else d:=4;
Label2.Caption:='oxirgi raqami='+floattostr(d);
end;
If e=5 then begin d:=5;
Label2.Caption:='oxirgi raqami='+floattostr(d);
end;
  If e=6 then begin d:=6;
Label2.Caption:='oxirgi raqami='+floattostr(d);
end;
if e=7 then begin
b:=a/4;
c:=frac(b)*4;
if c=0 then d:=1 else
if c=1 then d:=7 else
if c=2 then d:=9 else d:=3;
Label2.Caption:='oxirgi raqami='+floattostr(d);
end;
if e=8 then begin
b:=a/4;
c:=frac(b)*4;
if c=0 then d:=6 else
if c=1 then d:=8 else
if c=2 then d:=4 else d:=2;
Label2.Caption:='oxirgi raqami='+floattostr(d);
end;
if e=9 then begin
b:=a/2;
c:=frac(b)*2;
if c=0 then d:=1 else d:=9;
Label2.Caption:='oxirgi raqami='+floattostr(d);
End;
If e=1 then begin d:=1;
Label2.Caption:='oxirgi raqami='+floattostr(d);
end;
end;

```



3.24-rasm.

Mustaqil bajarish uchun topshiriqlar

1. 1 dan 100 gacha bo‘lgan sonlarning yig‘indisini va ko‘paytmasini toping.
2. $s=1-2+3-4+5- \dots$ qator yig‘ingisini berilgan hadlar uchun hisoblang.
3. 2,3,4,5,6,7,8,9 sonlariga ko‘paytirilganda raqmlarining yig‘indisi o‘zgamaydigan ikki xonali sonlarni toping.
4. Raqamlari yig‘indisi berilgan butun songa teng uch xonali barcha sonlarni toping.
5. Barcha shunday uch xonali sonlarni topingki, ularni kvadratga ko‘targanda uchata raqami bir xil bo‘lsin, nol bundan mustasno.
6. Shunday to‘rt xonali sonlarni topingki, 133ga bo‘lganda qoldiq 125ga va 134ga bo‘lganda qoldiq 111ga teng bo‘lsin.
7. Raqamlari yig‘indisi berilgan butun songa teng to‘rt xonali barcha sonlarni toping.
8. Berilgan natural l_0 sonni bo‘luvchilari yig‘indisini toping.
9. Berilgan N sonidan oshmaydigan barcha pifagor natural sonlarni toping. (Agar berilgan uch xonali sonning dastlabki ikkita raqami yig‘indisining kvadrati uchinchi raqamiga teng bo‘lsa, bunday natural son pifagor soni deb ataladi)
10. Birinchi ikkita raqamlari yig‘indisi oxirgi ikkita raqamlari yig‘indisiga teng bo‘lgan barcha to‘rt xonali sonlarni toping.
11. Berilgan natural sonni bo‘luvchilari yig‘indisini toping.
12. N dan oshmaydigan barcha natural pifagor sonlarini toping (Agar ikkita raqamining yig‘indisining kvadrati uchunchisining kvadratiga teng bo‘lsa, bunday uchta xonali natural son pifagor soni deb ataladi).

$$13. s = \prod_{k=1}^6 k^2$$

$$14. s = \sum_{k=2}^{10} \sqrt{k}$$

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

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

$$17. p = \prod_{n=1}^6 \sum_{m=1}^{10} \frac{\sqrt{n+m}}{n^{m+1}}$$

$$18. y = \sqrt{3 + \sqrt{6 + \dots \sqrt{96 + \sqrt{99}}}}$$

19.

$$s = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \dots$$

20.

$$s = \frac{1}{2*3} + \frac{2}{3*4} + \frac{3}{4*5} + \dots$$

3.5. Massivlar ustida amallar

Ko'p hollarda jadval yoki matritsalar ko'rinishidagi ma'lumotlar bilan ish yuritish kerak bo'ladi. Jadvalda ma'lumotlar juda ko'p bo'lgani sababli, ularning har bir yacheykasidagi sonni mos ravishda bitta o'zgaruvchiga qiymat qilib berilsa, ular ustida ish bajarish ancha noqulayliklarga olib keladi. Shu sababli 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 sababli, 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:array[1..10] of real; C,A1,D1:Array[1..10,1..20] of real;

bu yerda A va B massivlari 10 tadan elementga ega. C,A1,D1 massivlari esa 10x20=200 tadan elementga ega.

Massivlarni e'lon qilishdan maqsad massiv elementlari uchun kompyuter xotirasidan joy ajratishdir. Yuqoridagi ma'lumotlarni tasdiqlash uchun massivlarga oid bir nechta misollar ko'rib o'tamiz.

1-misol. Guruhlar ro'yxatini chiqarish dasturi.

Standart komponentalar palitrasidan **Button1**, **Label1** tugmalarini hosil qilamiz va **Button1** tugmasining dasturlash maydoniga quyidagi:

Const NT = 6;

var

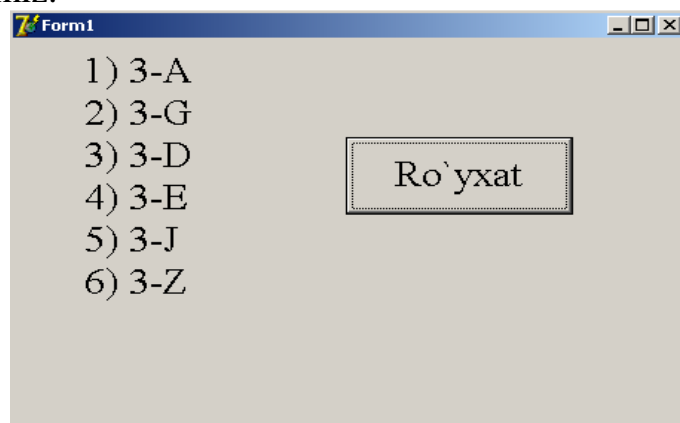
Massiv: array[1..NT] of string[10] =

('3-A','3-G','3-D','3-E','3-J','3-Z');

```

procedure TForm1.Button1Click(Sender: TObject);
var
st:string; // massiv ro'yxati
i:integer; // indeks, massiv elementlarining nomeri
begin
// formalashgan ro'yxat formada joylashish bo'yicha
for i:=1 to NT do
st:= st + IntToStr(i)+ ' ) '+ Massiv[i] + #13;
Label1.Caption:= st; // ro'yxatni chiqarish
end;
dastur kodini kiritamiz.

```

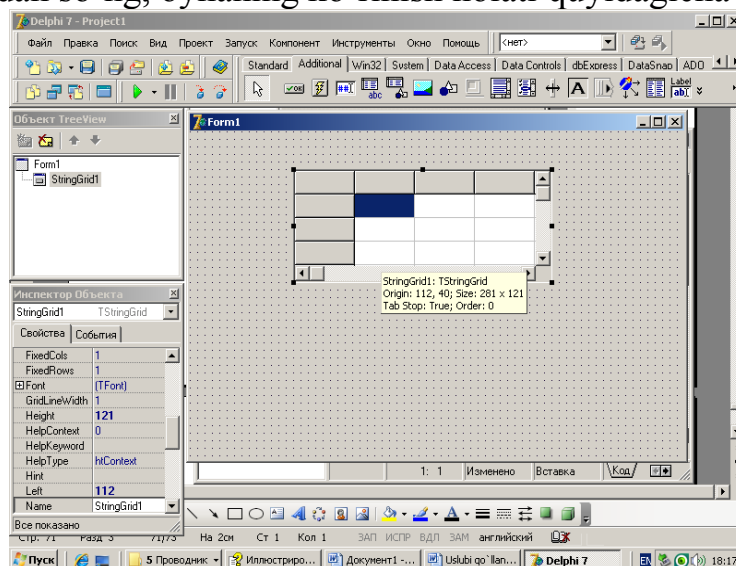


3.25-rasm.

2-misol. A(5) massiv elementlarini yig'indisi va o'rta arifmetigini topish dasturini tuzish.

Standart komponentalar palitrasidan **Button1**, **Additional** komponentalar palitrasidan **StringGrid1** tugmalari hosil qilinadi.

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



3.26-rasm.

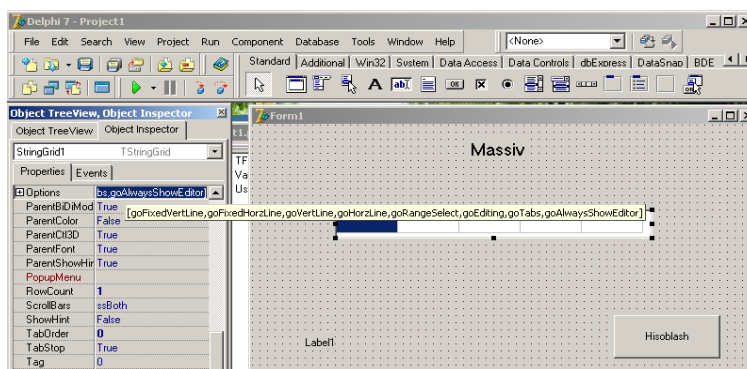
Hosil bo'lgan jadvaldan 5 ta ustun, 1 ta satr yarating. Bu jadvalni yaratish uchun parametrlar va xossalr bo'limidan **ColCount** hodisalar bandiga 5 raqamini kiritamiz. Satr bo'yicha o'zgartirish kiritish uchun **RowCount** bandiga 1 raqamini kiritamiz.

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

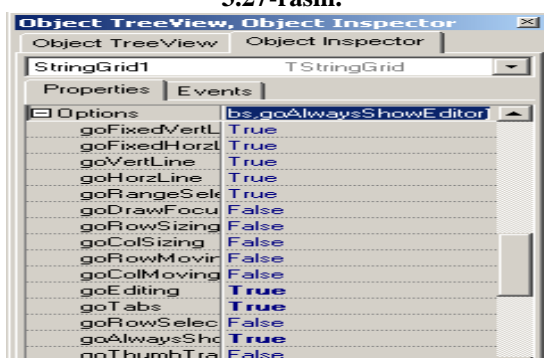
3.8-jadval

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).



3.27-rasm.



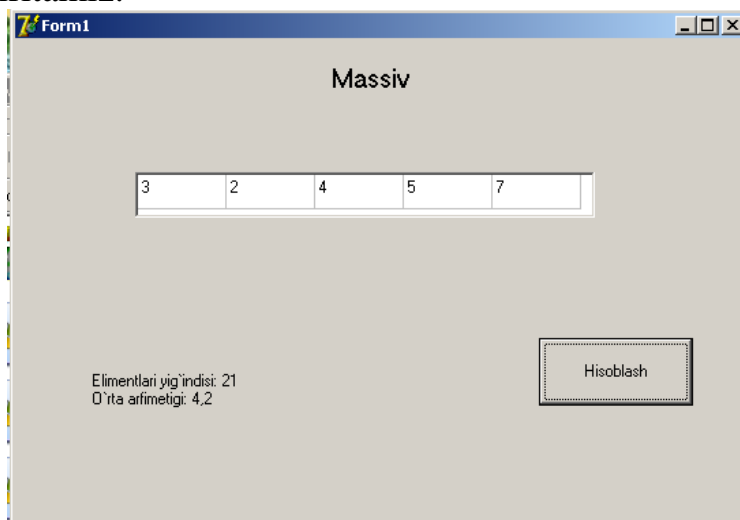
3.28-rasm.

Hodisalar bo‘limining **goEditing** bandida **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. Yuqorida bajarilgan algoritmdan so‘ng, “Button1” tugmasining dasturlash maydoniga quyidagi:

```

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 :=
'Elementlari yig'indisi: ' + IntToStr(summ)
+ #13+ 'O'rta arifmetigi: ' + FloatToStr(sr);
end;
dastur kodlarini kiritamiz.

```



3.29-rasm.

2-usul.

1-masalani ikkinchi usul bilan hisoblaymiz.

Standart komponentalar palitrasidan **Button1**, **Button2**, **Memo1**, **ComboBox1** tugmalari hosil qilinadi va **Forma1** oynasining dasturlash maydoniga quyidagi:

```

var
Form1: TForm1;
a:Array[1..5] of integer;
i:integer;
s,s1,s2:real;
procedure TForm1.FormCreate(Sender: TObject);
begin
i:=0;
ComboBox1.Clear;

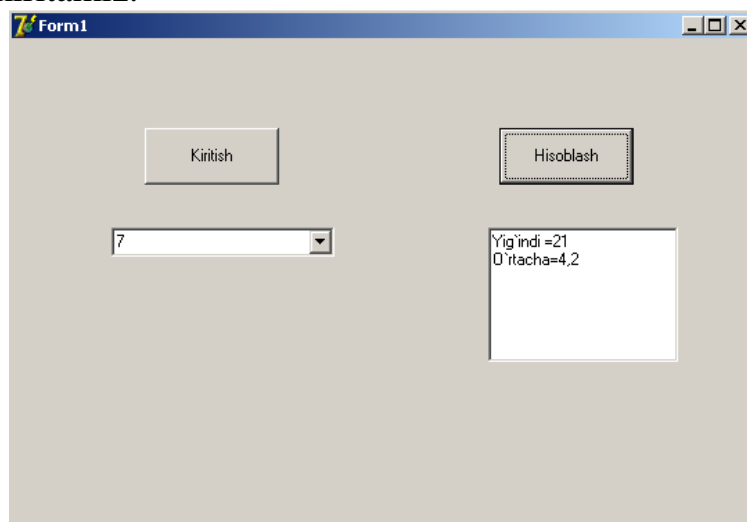
```



```

end;
“Button1” tugmasining dasturlash maydoniga quyidagi:
begin
ComboBox1.Items.Add(ComboBox1.text);
  i:=i+1;
a[i]:=StrToInt(ComboBox1.text);
  ComboBox1.SetFocus;
end;
“Button2” tugmasining dasturlash maydoniga quyidagi:
begin
  S:=0;
  For i:=1 to 5 do
    s:=s+a[i];
  Memo1.Clear;
  Memo1.Lines.add('Yig'indi =' + floattostr(s));
  for i:=1 to 5 do
    s1:=s/5;
  Memo1.Lines.add('O'rtacha=' + floattostr(s1));
end;
dastur kodlarini kiritamiz.

```



3.30-rasm.

3-usul.

1-masalani uchunchi usul bilan hisoblaymiz.

Standart komponentalar palitrasidan **Button1**, **Button1**, **Memo1**, **Edit 1**, va **ListBox1** tugmalarini hosil qilamiz va **Form1** oynasining dasturlash maydoniga quyidagi:

```

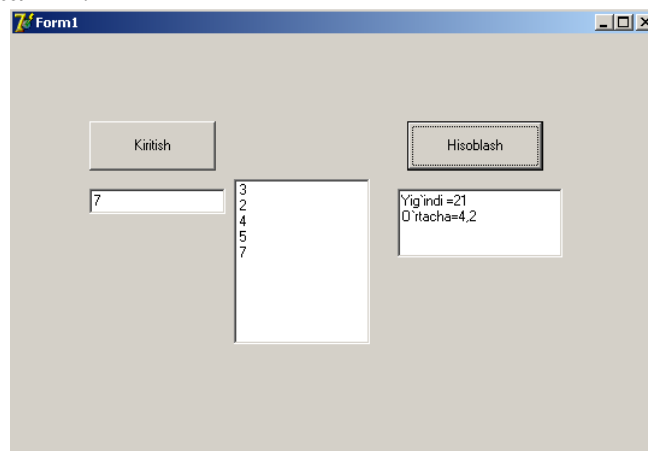
var
  Form1: TForm1;
  a:Array[1..5] of integer;
  i:integer;
  key,s,s1:real;
procedure TForm1.FormCreate(Sender: TObject);
begin

```

```

i:=0;
ListBox1.Clear;
end;
“Button1” tugmasining dasturlash maydoniga quyidagi:
begin
ListBox1.Items.Add(Edit1.text);
    i:=i+1;
    a[i]:=StrToInt(Edit1.text);
    Edit1.SetFocus;
end;
“Button2” tugmasining dasturlash maydoniga quyidagi:
begin
    S:=0;
    For i:=1 to 5 do
        s:=s+a[i];
    Memo1.Clear;
    Memo1.Lines.add('Yig'indi =' + floattostr(s));
    for i:=1 to 5 do
        s1:=s/5;
    Memo1.Lines.add('O'rtacha=' + floattostr(s1));
end;
dastur kodlarini kiritamiz.

```



3.31-rasm.

3-misol. A(10) vektorning toq va juft o‘rinlarida joylashgan elementlarini B(K) vektorga ketma-ket yozish dasturini tuzing.

Standart komponentalar palitrasidan **RadioButton1**, **RadioButton2**, **Additional** komponentalar palitrasidan **BitBtn1**, **StringGrid1**, **StringGrid2** tugmalari hosil qilinadi va **BitBtn1** tugmasining dasturlash maydoniga quyidagi:

```

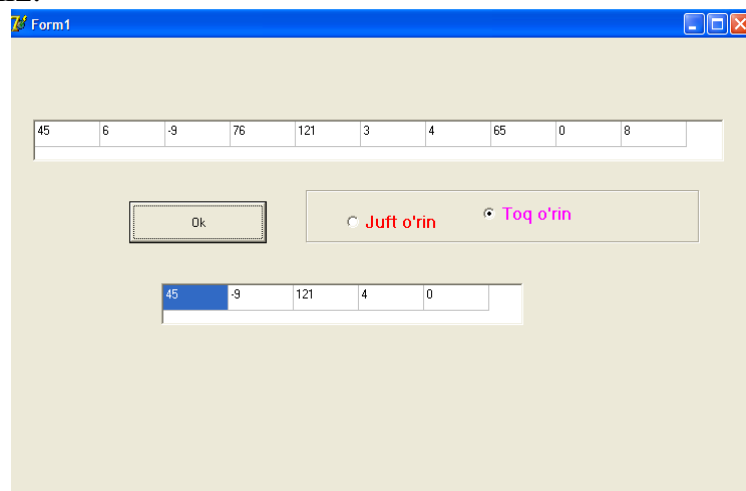
var i,k:integer;
begin
k:=0;
if RadioButton1.Checked then
begin
for i:=0 to 9 do begin

```

```

    if odd(i) then      begin
        StringGrid2.Cells[k,0]:=StringGrid1.Cells[i,0];
        k:=k+1;
    end;
end;
end
    else
    begin
    for i:=0 to 9 do
    begin
    if not odd(i) then      begin
        StringGrid2.Cells[k,0]:=StringGrid1.Cells[i,0];
        k:=k+1;
    end;
end;
end;
end;
dastur kodini kiritamiz.

```



3.32-rasm.

4-misol. $A(5,5)$ massiv elementlarini yig'indisi va o'rta arifmetigini topish dasturi.

Standart komponentalar palitrasidan **Button1**, **Memo1**, **Additional** komponentalar palitrasidan **StringGrid1** tugmalarini hosil qilamiz va **Button1** tugmasining dasturlash maydoniga quyidagi:

```

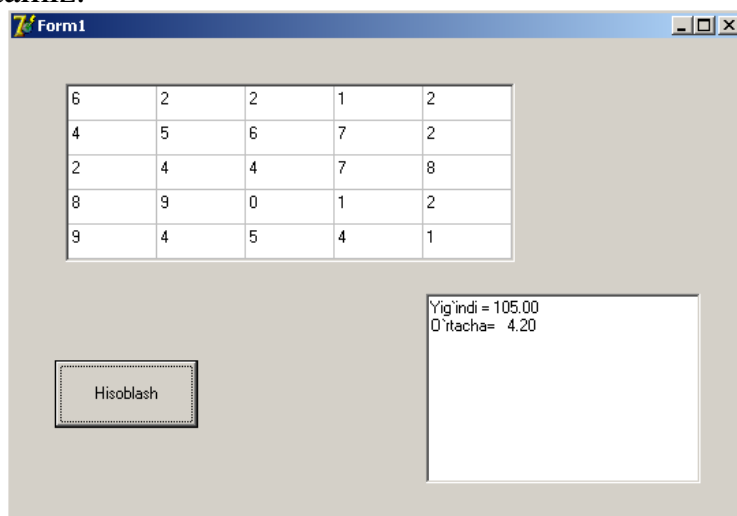
Var
i,j,cod:integer;
    A:array[1..5,1..5] of Real;
    S:real;
s1:String;
begin
    For i:=1 to 5 do
    For j:=1 to 5 do
    Val(StringGrid1.cells[i-1,j-1],a[i,j],cod);

```

```

S:=0;
For i:=1 to 5 do
  For j:=1 to 5 do
    s:=s+a[i,j];
    Str(s:7:2,s1);
    Memo1.Clear;
    Memo1.Lines.add('Yig'indi =' +s1);
    s:=s/25;
  Str(s:7:2,s1);
  Memo1.Lines.add('O'rtacha=' +s1);
end;
dastur kodini kiritamiz.

```



3.33-rasm.

5-misol. $A(N,N)$ kvadrat matritsani musbat va manfiy elementlarini mos ravishda 1 va 0 sonlari bilan almashtirish dasturi.

Standart komponentalar palitrasidan **Button1**, **Button2**, **Edit1** va **Additional** komponentalar palitrasida **StringGrid1** tugmalarni hosil qilinadi va **Button1** tugmasining dasturlash maydoniga quyidagi:

```

var col:integer;
begin
  col:=strtoint(Edit1.Text);
  StringGrid1.ColCount:=col+1;
  StringGrid1.RowCount:=col+1;
end;

```

dastur kodi kiritiladi.

Button2 tugmasining dasturlash maydoniga quyidagi:

```

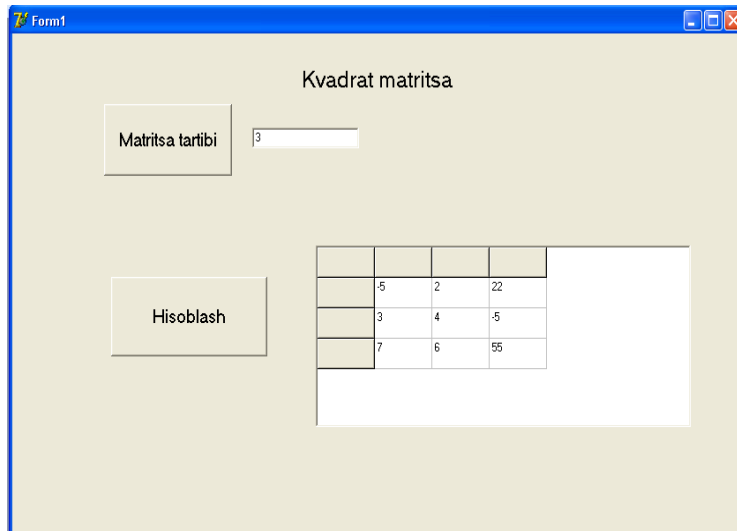
Var
  i,k:integer;
begin
  for i:=1 to strtoint(edit1.Text) do
    for k:=1 to strtoint(edit1.Text) do
      begin
        if strtoint(StringGrid1.Cells[i,k])<=0 then StringGrid1.Cells[i,k]:='0'

```

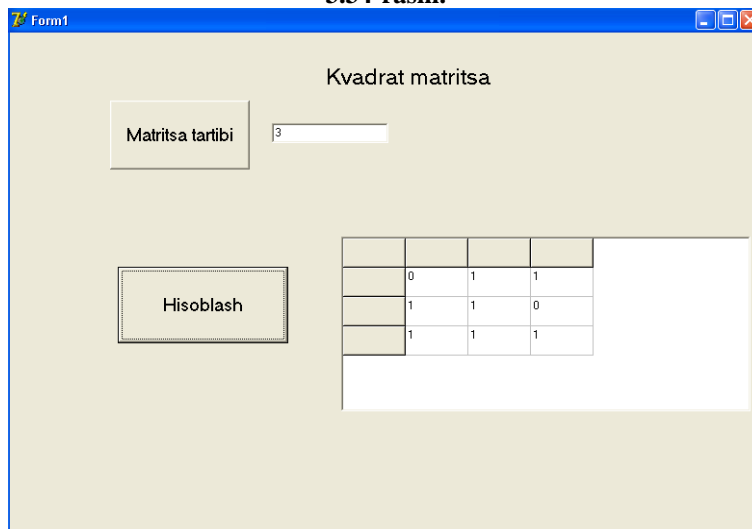
```

else
StringGrid1.Cells[i,k]:='1'
end;
dastur kodi kiritiladi.

```



3.34-rasm.



3.35-rasm.

6-misol. $c_{ij} = a_{ij} + b_{ij}$ matritsani hisoblash dasturi. Bunda $i = \overline{1,4}$, $j = \overline{1,4}$.

Standart komponentalar palitrasidan **Button1**, **Additional** komponentalar palitrasida **StringGrid1**, **StringGrid2**, **StringGrid3** tugmalarini hosil qilinadi va **Button1** tugmasining dasturlash maydoniga quyidagi:

```

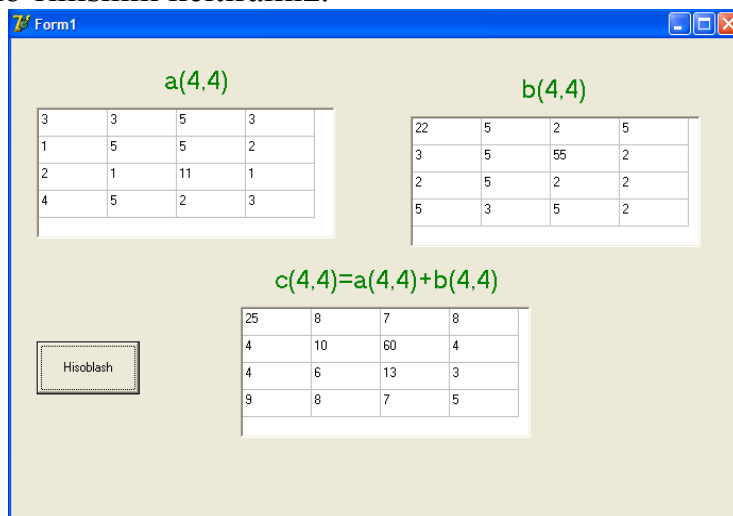
var
i,k:integer;
begin
for i:=1 to 4 do
begin
for k:=1 to 4 do
begin
stringgrid3.Cells[i-1,k-1]:=floattostr(strtoint(stringgrid1.cells[i-1,k-1])+strtoint(stringgrid2.cells[i-1,k-1]));
end;
end;
end;

```

end;

dastur kodini kiritamiz.

Oynani umumiy ko‘rinishini keltiramiz:



3.36-rasm.

7-misol. Ikkinchi tartibli determinantni hisoblash.

Standart komponentalar palitrasidan **Button1**, **Additional** komponentalar palitrasida **StringGrid1** tugmalari hosil qilinadi va **Button1** tugmasining dasturlash maydoniga quyidagi:

```
const n=2;
```

```
var
```

```
a:array[1..2,1..2] of integer;
```

```
i,k:integer;
```

```
s,s1,y:real;
```

```
begin
```

```
s:=1;
```

```
s1:=1;
```

```
for i:=1 to 2 do
```

```
for k:=1 to 2 do
```

```
a[i,k]:=strtoint(stringGrid1.cells[i-1,k-1]);
```

```
for i:=1 to 2 do
```

```
for k:=1 to 2 do
```

```
begin
```

```
if i=k then
```

```
s:=s*a[i,k];
```

```
if i=n+1-k then
```

```
s1:=s1*a[i,k];
```

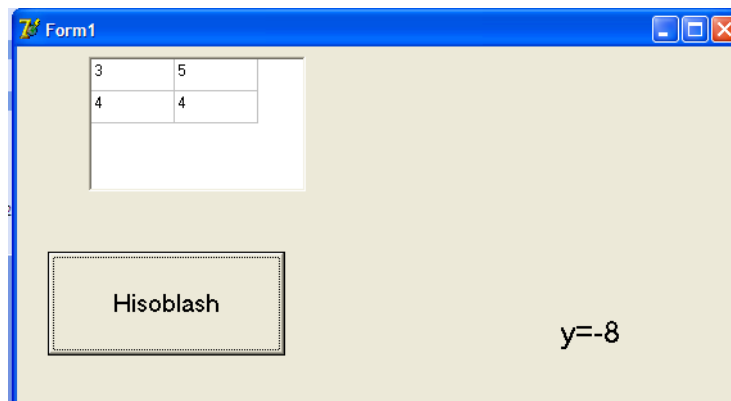
```
y:=s-s1;
```

```
end;
```

```
label1.caption:='y='+floattostr(y);
```

```
end;
```

dastur kodini kiritamiz.



3.37-rasm.

8-misol. $A(N,N)$ tartibli determinantni hisoblash dasturi. Bunda $n=2,3,4$ bo'lgan hol uchun.

Standart komponentalar palitrasidan **Button1**, **Label1**, **Label2**, **Label3** va **Edit1**, **Edit2**, **Additional** komponentalar palitrasidan **StringGrid1** tugmalarini hosil qilamiz va **Form1** oynasining dastur maydoniga quyidagi:

type

Tmass=array of Real;

Tmatrix=array of Tmass;

var

Form1: TForm1;

n:integer;

implementation

*{ \$R *.dfm }*

“Button1” tugmasining dastur maydoniga quyidagi:

begin

n:=StrToInt(Edit1.Text);

StringGrid1.ColCount:=n;

StringGrid1.RowCount:=n;

end;

“Button2” tugmasining dastur maydoniga quyidagi:

procedure Per(k,n:integer;var a:Tmatrix; var p:integer);

var z:Real;j,i:integer;

begin

z:=abs(a[k,k]);i:=k;p:=0;

for j:=k+1 to n-1 do

begin

if abs(a[j,k])>z then begin

z:=abs(a[j,k]); i:=j;

end; end;

if i>k then begin

p:=p+1;

for j:=k to n-1 do begin

z:=a[i,j];

a[i,j]:=a[k,j]; a[k,j]:=z; end; end; end;

```

function Znak(p:integer):integer;
begin
if p mod 2=0 then result:=1 else result:=-1; end;
procedure Opr(n:integer;var a:tmatrix;var det:real);
var
k,i,j,p:integer;
r:real;
begin
det:=1.0;
for k:=0 to n-1 do begin
if a[k,k]=0 then Per(k,n,a,p);
det:=znak(p)*det*a[k,k];
for j:=k+1 to n-1 do
begin
r:=a[j,k]/a[k,k];
for i:=k to n-1 do
a[j,i]:=a[j,i]-r*a[k,i];
end;
end;
end;
var k,j,i:integer;
a:Tmatrix;
det:real;
begin
n:=strtoint(edit1.Text);
SetLength(a,n,n);
for k:=0 to n-1 do
for j:=0 to n-1 do
a[k,j]:=strtofloat(StringGrid1.Cells[j,k]);
Opr(n,a,det);
Edit2.Text:=FloatToStrF(det,ffFixed,5,0);
end;
dastur kodini kiritamiz.

```

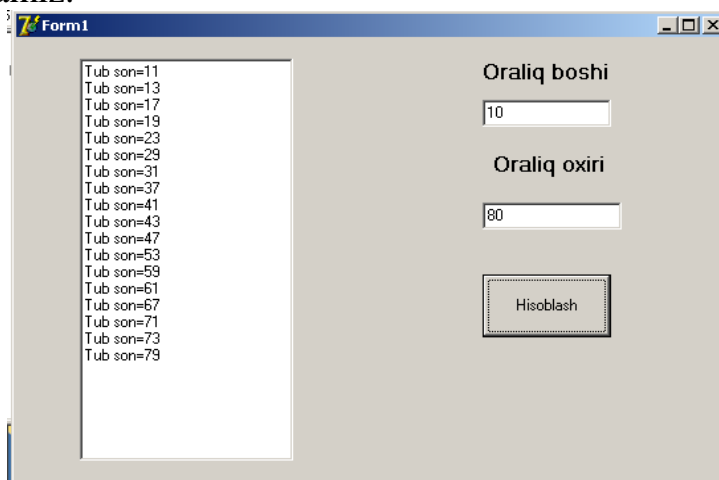
3.38-rasm.

9-Misol. Tub sonlar jadvalini chiqaruvchi dastur.

Standart komponentalar palitrasidan **Edit1**, **Edit2**, **Button1**, **Label1**, **Label2**, **Memo1** tugmalarini hosil qilinadi va **Button1** tugmasining dastur maydoniga quyidagi:

```
const n=100000;
var
  k,m,l, i:integer;
  a:array[1..n] of integer;
begin
  m:=strtoint(edit1.Text);
  l:=strtoint(edit2.Text);
  for i:= 2 to n do
    a[i]:=i;
    for i:=1 to n do
      for k:=2 to trunc(sqrt(i)) do
        if (a[i]/k-trunc(a[i]/k))=0 then a[i]:=0;
  for i:=m to l do
    if a[i]<>0 then
      Memo1.Lines.add('Tub son='+inttostr(a[i]));
end;
```

dastur kodini kiritamiz.



3.39-rasm.

Mustaqil bajarish uchun topshiriqlar

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

2. $y = \sum_{i=1}^5 \frac{ik_i}{2^i}$ 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 kichigini topish dasturini tuzing.
7. $A(10)$ vektor elementlarini eng kattasini topish dasturini tuzing.
8. $A(5)$ vektor elementlarini o'sib borish tartibida joylashtiring.
9. $A(5)$ vektor elementlarini kamayib borish tartibida joylashtiring.
10. Uchunchi tartibli kvadrat matritsaning teskarisini topish dasturini tuzing
11. X vektor berilgan. Vektor elementlarining tartibini buzmasdan, yangi massiv kiritmasdan, faqat nolga teng bo'lgan elementlarni massivning oxiriga joylashtiring.
12. A va B vektorlar berilgan. C vektorni shunday tashkil qiling, unung elementlari A da ham, B da ham mavjud bo'lsin.
13. $A(N, N)$ matritsaning p va q tartib raqamli satrlari o'rnini almashritiring.
14. $X(K, L)$ matritsaning eng katta va eng kichik elementlari o'rnini almashtiring.
15. $A(3, 4)$ matritsaning satr elementlari ko'paytmasidan B massivni hosil qiling.
16. $Z(3, 4)$ matritsaning har bir ustunidagi manfiy elementlar sonidan tashkil topgan M vektorni hosil qiling.
17. $A(M, N)$ matritsa berilgan. Har bir satrdagi eng kichik elementlar orasidan eng kattasini va u joylashgan tartib raqamini aniqlang.
18. Diagonal elementlaridan tashqari barchga elementlari nolga teng bo'lgan $C(M, M)$ matritsa tashkil eting.
19. $K(3, 4)$ matritsaning musbat elementlaridan tashkil topgan L vektorni hosil qiling.
20. $A(10, 15)$ matritsaning har bir ustunidagi musbat elementlarining sonini va yig'ingisini eslab qoluvchi va hisoblovchi dastur tuzing. Natija ikkita satr ko'rinishida chop etilsin. $a_{ij} > 0$.

3.6. Protsedura va funksiya

Dastur tuzish jarayonida uning turli joylarida ma'nosiga ko'ra bir xil mustaqil xarakterga ega bo'lgan va yechilayotgan asosiy masalaning biror qismini hal qilishni o'z bo'yniga olgan murakkab algoritmdan bir necha marotaba foydalanishga to'g'ri keladi. Masalan, matritsalarini ko'paytirish, matritsani vektorga ko'paytirish, chiziqsiz tenglamani yechish, chizikli algebraik tenglamalar tizimsini yechish, faktorial hisoblash, yig'indi hisoblash va hokazo kabi masalalarni hal qilish algoritmlari juda ham ko'p masalalarni yechishning bosh algoritmlarida qayta-qayta turli boshlang'ich ma'lumotlar bilan qatnashishi mumkin. Bunday hollarda, malakali dasturchi dastur matnini ixchamlashtirish, dasturning ishonchlilik darajasini oshirish, dasturni taxrirlash (otladka)ni tezlashtirish va dasturning umumiyligini (universalligini) ta'minlash uchun protsedura va funksiyalardan kengroq foydalanib, mukammal dastur yaratishga harakat qiladi.

Protsedura va funksiyalar mustaqil dasturli ob'ektlar hisoblanadi. Bu mustaqil dasturli ob'ektni dasturchi o'z xoxishiga va undan olinadigan natijalariga ko'ra protsedura yoki funksiya ko'rinishida aniqlashi mumkin. Odatda olinadigan natija yagona qiymatli bo'lsa funksiyadan, olinadigan natijalar soni bir nechta bo'lsa protseduradan foydalanish maqsadga muvofiqdir.

Protsedurani yozish strukturasi xuddi asosiy dastur strukturasi kabi bo'lib, faqat sarlavhalari bilangina farq qiladi xolos:

procedure <protsedura ismi>(<formal parametrlar ro'yxati>);

label <metkalar ro'yxati>;

const <o'zgaruvchlarni kiritish>;

type <yangi turlarni aniqlash>;

var <o'zgaruvchilarning turlarini e'lon qilish>;

<qism dasturgagina tegishli bo'lgan ichki protsedura va funksiyalar e'loni>;

begin

<protseduraning tana qismi>

end;

Protseduralar va funksiyalarni aniqlash asosiy dasturning **var** (o'zgaruvchilarning turlarini e'lon qilish) bo'limida bajariladi. Protseduradan dasturda foydalanish uchun uning ismi va faktik parametrlar ro'yxati yoziladi.

Shunda protsedura o'ziga belgilangan ishni bajarib, o'zining faktik parametrlari orqali asosiy dasturga o'z natijasini beradi.

Protseduraning e'loni va unga murojaat qilishni, keyinchalik ko'riladigan misollar orqali o'zlashtirib olamiz.

Parametrsiz protseduralar.

Yuqorida aytib o'tganimizdek, protsedura hisoblab bergan natijalar uning faktik parametrlari orqali asosiy dasturga uzatiladi. Lekin, ayrim hollarda protsedura parametrsiz ham bo'lishi mumkin. Bu holda asosiy dasturning barcha parametrlari protsedura parametrlari rolini bajaradi. Parametrsiz protsedurada ham protseduraning barcha bo'limlari saqlanib qoladi, faqat parametrlar ro'yxatigina qatnashmaydi.

Protseduralarni aniqlash va ulardan foydalanishni quyidagi misol ustida ko'rib chiqaylik:

1-misol: $u = \max(x+y, x*y)$, $v = \max(0.5, u)$ – berilgan x va y haqiqiy sonlardan foydalanib u va v qiymatlarni aniqlash. Bu yerda x , u - qiymatlari kiritiladigan haqiqiy turli o'zgaruvchilar.

1. Masalani yechish dasturining protseduradan foydalanmay tuzilgan holi:

var

x, y, u, v : *real*;

a, b, s : *real*;

begin

$x := \text{strtofloat}(\text{edit1.Text});$

$y := \text{strtofloat}(\text{edit2.Text});$

$a := x + y$; $b := x * y$;

if $a > b$ **then** $s := a$ **else** $s := b$;

$u := s$;

```

a:= 0.5; b:=u;
if a > b then s:= a else s:=b;
v:=s;
label1.Caption:='u='+floattostr(u)+#13+'v='+floattostr(v);
end;

```

Ahamiyat bersangiz, dasturdagi shartli operator ikki marta takrorlanib, bir xil ish bajardi.

2. Masalani yechish dasturini parametrsiz protseduradan foydalanib tuzilgan holi (endi yuqoridagi dasturda yo‘l qo‘yilgan kamchilikni protseduralar orqali tuzatishga harakat qilamiz):

```

var x, y, u, v: real; a, b, s: real;
procedure max1;
begin
if a>b then s:=a else s:=b;
end;
begin
x:=strtofloat(edit1.Text);
y:=strtofloat(edit2.Text);
a:= x + y; b:=x * y;
max1; {max1 protsedurasiga 1-marta murojaat qilinmoqda}
u:=S;
a:=0.5; b:= u;
max1; { max1 protsedurasiga 2-marta murojaat qilinmoqda }
v:=S;
label1.Caption:='u='+floattostr(u)+#13+'v='+floattostr(v);
end.

```

Asosiy dasturning operatorlar qismida ikki marta yozilgan max1 parametrsiz protsedurasiga murojaat, e‘lon qilingan protsedurani ikki marta asosiy dasturga olib kelib ishlatishni tashkil qiladi. Ahamiyat berilsa, ikkinchi dastur birinchi protsedurasiz tuzilgan dasturga ko‘ra ixchamroq va soddaroqdir. Biz kiritgan protsedura hozircha faqat ikkita haqiqiy son ichidan kattasini aniqlab berdi xolos, shuning uchun dastur matning hajmini kamaytirishdan erishilgan yutuq salmoqli bo‘lmadi. Lekin, protseduralar asosan ko‘p hajmli matndagi amallarni, vazifalarni bajarishga mo‘ljallanadi va bu holda erishilgan yutuq salmog‘i ancha yuqori bo‘ladi.

Parametrsiz protseduraning asosiy kamchiligi, uning asosiy dasturga va undagi ma‘lum parametrlarga bog‘lanib qolganligidir.

Parametrli protseduralar.

Protsedura bilan asosiy dasturni bog‘laydigan asosiy faktor bu – protsedura parametrlaridir. Parametrlar ikkita turga ajratiladi: qiymatli parametrlar (parametr-qiymat), o‘zgaruvchili parametrlar (parametr - o‘zgaruvchi).

Parametr - qiymat bu protsedurani ishlash jarayonini ta‘minlovchi parametrlar hisoblanadi, ya‘ni asosiy dastur qiymatlarini protseduraga uzatadigan parametrlardir.

Endi, yuqorida ko‘rib chiqilgan sonlarni eng kattasini topish algoritmining dasturini qiymatli parametr bilan yozilgan protseduralar orqali amalga oshiraylik:

```

var x, y, u, v: real; S: real;

```

```

procedure max2 ( a, b: real );
begin
if a>b then S:=a else S:=b;
end;
begin
x:=strtofloat(edit1.Text);
y:=strtofloat(edit2.Text);
    max2 (x + y, x * y); u:=S;    max2 (0.5 , u); v:=S;
    labell.Caption:='u='+floattostr(u)+#13+'v='+floattostr(v);
end;

```

bu yerda a, b - protseduraning qiymatli formal parametrlari.

Protseduraga murojaat qilishda formal va faktik parametrlarning turlari o‘zaro mos kelishi kerak, aks holda dastur xato tuzilgan hisoblanadi. Yuqoridagi dasturdan ko‘rinib turibdiki, a va b formal parametrlar o‘rniga natijaviy qiymatlari ma’lum ifodalar qo‘yildi. Demak, qiymatli faktik parametrlar o‘rniga, shu turli natijaga erishuvchi ifoda yozilishi mumkin. Bundan tashqari, protsedurada kiritilgan a va b parametrlari faqat protseduraning ichidagina ma’noga ega, tashqarida, misol uchun asosiy dasturda ular tushunarsiz, qiymatlari aniqlanmagan miqdorlardir. Shuning uchun, qiymatli parametrlarga protsedura natijalarini o‘zlashtirib, asosiy dasturga uzatib bo‘lmaydi.

Yuqorida tuzilgan dasturning asosiy kamchiligi, topilgan katta son doim **S** o‘zgaruvchisiga o‘zlashtirilayapti. Misolimiz shartiga ko‘ra, natijalar **u** va **v** o‘zgaruvchilariga o‘zlashtirilishi kerak. Shuning uchun, dasturda ikki marta qo‘shimcha **u:=S** va **v:=S** o‘zlashtirish operatorlari yozildi.

Bu kamchilikni tuzatish uchun protseduraga yana bir parametrni kiritamiz. Lekin, kiritilgan bu parametr protseduraga qiymat olib kirmaydi balki, protsedura natijasini asosiy dasturga olib chiqib ketadi. Bunday parametrni parametr - o‘zgaruvchi deb ataladi.

Parametr-o‘zgaruvchini parametr-qiymatdan farq qilish uchun protsedurani aniqlashdagi parametrlar ro‘yxatida o‘zgaruvchi oldidan **var** xizmatchi so‘zi yoziladi. Parametr - o‘zgaruvchidan so‘ng albatta, uning turi ko‘rsatib qo‘yiladi. Yuqorida aytganimizdek, formal parametr - qiymat o‘rniga protseduraga murojaat vaqtida shu turli ifoda yozish mumkin bo‘lsa, parametr - o‘zgaruvchi uchun bu hol mutlaqo mumkin emas.

Protsedurani mukammallashtirib borish dinamikasini his etish uchun yana, yuqorida ko‘rilgan maksimum topish misolining dasturini parametr - o‘zgaruvchi ishlatilgan holda ko‘rib chiqamiz:

```

var
    x, y, u, v: real;
procedure max3(a, b: real; var S: real);
begin
    if a>b then S:=a else S:=b;
end;
begin
    x:=strtofloat(edit1.Text);

```

```

y:=strtofloat(edit2.Text);
max3( x + y, x * y, u); {x+y va x*y ifodalarining kattasi u o'zgaruvchisiga
o'zlashtirilmoqda}
max3( 0.5, u, v); {0.5 va u ifodalarining kattasi V o'zgaruvchisiga o'zlashtirilmoqda}
label1.Caption:='u='+floattostr(u)+#13+'v='+floattostr(v);
end;

```

Shunday qilib, bitta dasturni protseduraning uch xil varianti uchun tuzib chiqib, natijada ixcham va sodda dasturga ega bo'ldik.

Protseduralarni aniqlashda shu paytgacha oddiy turli parametrlardan foydalanib keldik. Lekin, biz shuni yaxshi bilamizki, Paskal tilida hosilaviy turlar ham mavjud. Parametr - o'zgaruvchiga hosilaviy, yangi turlar berish xuddi oddiy skalyar tur berish kabi amalga oshirilaveradi. Ammo, parametr - qiymatlarda yangi turlar masalasiga batafsilroq yondashish kerak.

Biz yuqorida eslatib o'tdikki, faktik parametr formal parametr - qiymatga mos turli ixtiyoriy ifoda bo'lishi mumkin. Lekin, Paskal tilida ixtiyoriy turli qiymatlar uchun shu turdagi natija beruvchi hech qanday amal ko'zda tutilmagan. Shuning uchun, bu turlar uchun faktik parametrlar faqat shu turga mos o'zgaruvchilar bo'lishi mumkin xolos. Bunday hol, xususiy holda massivlar uchun ham o'rinlidir.

Faraz qilaylik, dasturda o'zgaruvchilar quyidagicha e'lon qilingan bo'lsin:

```

const n=20;
type vector = array [1..N] of real;
var u, v: real;
x, y: vector;

```

Bu yerda $\mathbf{u}=\max\{x_i\}$, $\mathbf{v}=\max\{y_i\}$ larni aniqlash talab qilinayotgan bo'lsin. Vektorning eng katta hadini topishni albatta protsedura ko'rinishida tashkil qilamiz:

```

procedure max1 (A:vector; var S: real);
var i: integer;
begin
  S:=A[1];
  for i:=2 to n do
    if A[i] > S; then S:= A[i];
  end;

```

Bu protseduraga asosiy dasturda murojaat $\mathbf{max1(x,u)}$; $\mathbf{max1(y,v)}$; ko'rinishida amalga oshiriladi.

Protseduradagi A vektorini parametr - qiymat sifatida yozib qo'yganimiz uchun, protseduraga qilinayotgan har bir murojaatda A vektorga mos ravishda X va Y vektorlar ko'chirib yoziladi va so'ngra protsedura o'z ishini bajaradi. Biz bilamizki, bir tarafdin, massivlarning ustida ko'chirish amalini bajarishga ancha vaqt ketadi, ikkinchi tarafdin, har safar yangidan protseduraga qilingan murojaatda A vektor uchun xotiradan qo'shimcha joy ajratiladi. Shuning uchun, protseduraning sarlavhasida quyidagicha almashtirish qilsak, yuqoridagi ikki kamchilikni bartaraf qilgan bo'lamiz:

```

procedure max1 (var A: vector; var S: real);

```

Endi protsedurani e'lon qilish, undan foydalanib dastur yaratish malakasini hosil qilganimizdan so'ng, uni e'lon qilishning sintaksis qoidalarini ko'rib chiqaylik.

Protsedurani aniqlash (e'lon qilish) quyidagicha amalga oshiriladi:

<protsedurani aniqlash>::=<protsedura sarlavhasi>;<blok>

Bu yerda <blok> tushunchasi to'liqligicha <dastur tanasi> tushunchasi bilan bir xil sintaksis qoida asosida aniqlangani uchun, bu tushunchaga ortiq qaytib o'tirmaymiz.

Endi esa <protsedura sarlavxasi>ga ta'rif beramiz:

<protsedura sarlavhasi>::=Procedure <protsedura ismi> Procedure <protsedura ismi>(<formal parametrlar ro'yxati>)

Protsedura ismi dasturchi tomonidan tanlanadigan oddiy identifikator hisoblanadi.

Formal parametrlar ro'yxati quyidagicha aniqlanadi:

<formal parametrlar ro'yxati>::=<formal parametrlar seksiyasi> {; <formal parametrlar seksiyasi>}

Formal parametrlar seksiyasi deganda protsedura parametrlarining parametr-qiyamat va parametr-o'zgaruvchilardan iborat bo'lishligi tushuniladi:

<formal parametrlar seksiyasi>::={<ism> {, <ism>}: <ism turi> | var <ism>{,<ism>}:<ism turi>

bu yerda <ism> - formal parametrlar sifatida ishlatiladigan identifikator.

Endi yuqoridagi aniqlashlarga tushuntirishlar berib o'tsak.

Yuqoridagi Bekus-Naur formulalaridan ko'rinib turibdiki, formal parametrlar ro'yxati (agar u mavjud bo'lsa) bitta yoki bir nechta o'zaro nuqta-vergul (;) belgisi bilan ajratilgan seksiyalardan tashkil topgan. Har bir seksiyada esa o'z navbatida, bitta yoki bir nechta o'zaro nuqta-vergul bilan ajratilgan formal parametrlar qatnashishi mumkin. Protseduradagi formal parametrlar sonini, dasturchining o'zi protsedurani aniqlash moxiyatidan kelib chiqqan holda tanlaydi.

Masalan,

Procedure P(A:Char; B:Char; Var C:Real; Var D:Real; E:Char);

bu yerda formal parametrlar ro'yxati beshta seksiyadan iborat: A,B,E – lar Char turli qiymatlar, C, D – lar Real turidagi o'zgaruvchilar. Shu bilan bir qatorda har bir seksiya faqat, bitta parametрни o'z ichiga olmoqda.

Bir xil turli, hamda ketma-ket joylashgan qiymatlarni va o'zgaruvchilarni bitta seksiyaga birlashtirib protsedura sarlavhasini quyidagicha yozish ham mumkin:

Procedure P(A,B:Char; Var C,D:Real; E:Char);

Shunday qilib, seksiya deganda bir xil turli parametr - qiymatlar yoki parametr - o'zgaruvchilarning ro'yxatini tushunish mumkin.

Ko'pchilik boshlovchi dasturchilar yo'l qo'yadigan quyidagi xatoliklardan ehtiyot bo'lmoq zarur:

Procedure P(Var X:Real; Y:Real);

bu sarlavha

Procedure P(Var X,Y:Real);

sarlavhasi bilan bir xil emas.

Aniqlangan protseduraga murojaat yoki protsedura operatoridan qanday foydalanishni aniqlashni ko'rib chiqaylik (yaratilgan protsedurani «Aktivlashtirish», ya'ni ishlatish):

<protsedura operatori>::=<protsedura ismi> | <protsedura ismi>(<faktik parametrlar ro'yxati>)

Agar protsedura aniqlanishida parametrsiz bo'lsa, unga murojaat qilish ham faqat, protsedura ismini yozish bilangina amalga oshiriladi.

Agar protsedura aniqlanishida parametrli bo'lsa, albatta protsedura-operator ham unga mos faktik parametrlar ro'yxatiga ega bo'ladi. Shu parametrlar orqali protseduraga murojaat qilinayotganida, formal parametrlar faollashtiriladi:

\langle faktik parametrlar ro'yxati $\rangle ::= \langle$ faktik parametr $\rangle \{, \langle$ faktik parametr $\rangle\}$

Protseduralarni aniqlash va ulardan foydalanishni quyidagi misol yordamida ko'rib chiqaylik:

2-misol. Ekranga 5 raqamini chiqaruvchi dastur tuzing.

Standart komponentalar palitrasidan **Button1**, **Memo1** tugmalarini hosil qilamiz va **Button1** tugmasining dastur maydoniga quyidagi:

```
const n=10;
```

```
procedure chiqarish;
```

```
var
```

```
i:integer;
```

```
begin
```

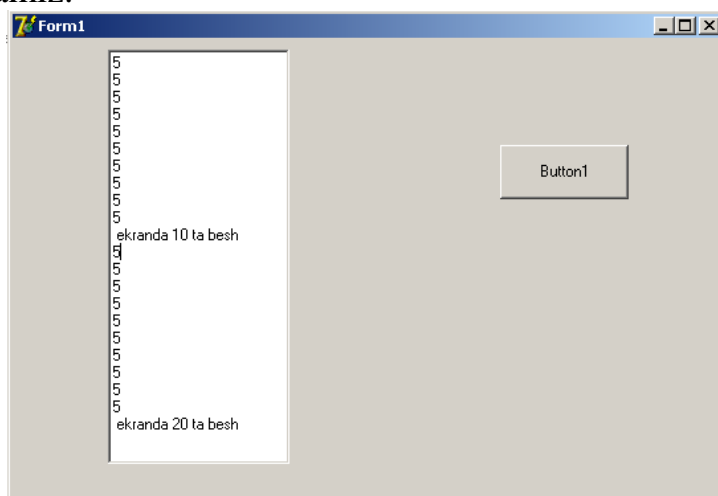
```
for i:=1 to n do Memo1.Lines.add('5'); end;
```

```
begin
```

```
chiqarish; Memo1.Lines.add(' ekranda 10 ta besh');
```

```
chiqarish; Memo1.Lines.add(' ekranda 20 ta besh'); end;
```

dastur kodini kiritamiz.



3.40-rasm.

3-misol. $n!$ ni hisoblash dasturi tuzing.

Standart komponentalar palitrasidan **Edit1**, **Button1**, **Memo1** tugmalarini hosil qilamiz va **Button1** tugmasining dastur maydoniga quyidagi:

```
var
```

```
y, n: integer;
```

```
function fact(m: integer): integer;
```

```
var
```

```
k: integer;
```

```
begin
```

```
if m=0 then fact:=1
```

```
else
```

```
fact:=fact(m-1) * m;
```

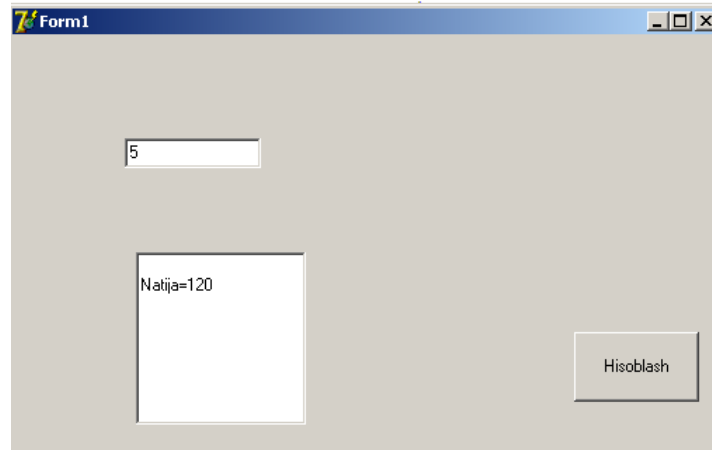


```

end;
begin
  n:=strtoint(edit1.Text);
  y:=fact(n);
  Memo1.Lines.add('Natija='+inttostr(y));
end;

```

dastur kodini kiritamiz.



3.41-rasm.

4-misol. $Y = x + 5$ funksiyani $-2 \leq x \leq 10$ oraliqda $\Delta x = 1$; qadam bilan hisoblash dasturi tuzing.

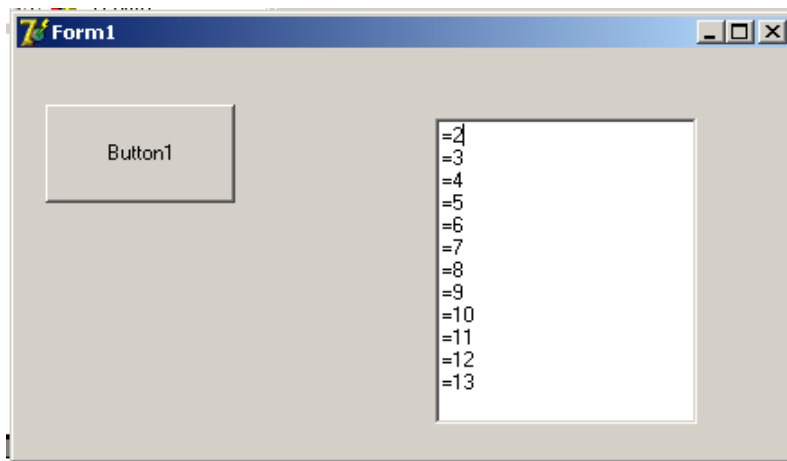
Standart komponentalar palitrasidan **Button1**, **Memo1** tugmalarini hosil qilamiz va **Button1** tugmasining dastur maydoniga quyidagi:

```

var
i:real;
procedure qismdastur(x:real);
var
y:real;
begin
y:=x+4;
Memo1.Lines.add('Natija='+floattostr(y));
end;
Begin i:=-2;
repeat
qismdastur(i);
i:=i+1;
until i=10;
end;

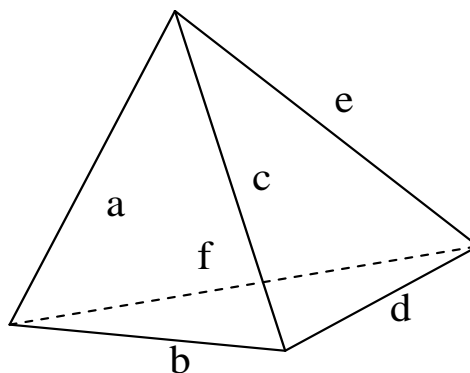
```

dastur kodini kiritamiz.



3.42-rasm.

5-misol. Quyidagi figurani yuzasini hisoblash dasturi tuzing.



3.1-chizma

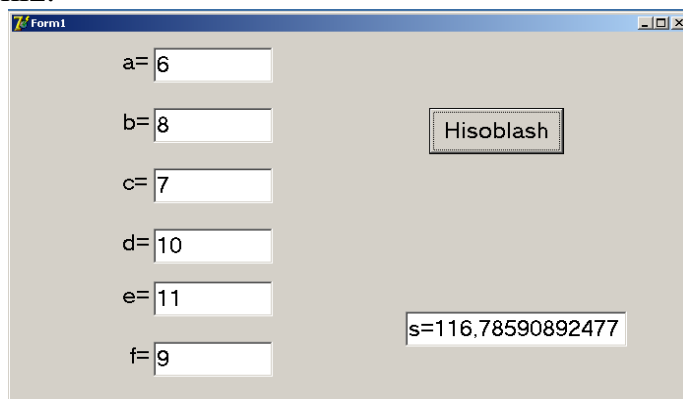
Standart komponentalar palitrasidan **Edit1, Edit2, Edit3, Edit4, Edit5, Edit6, Edit7, Button1** tugmalarini hosil qilamiz va **Button1** tugmasining dastur maydoniga quyidagi:

```

var
s1,s2,s3,s4,a,b,c,d,e,f,s:real;
procedure Sq(x,y,z:real; var s:real);
var p:real;
begin
p:=(x+y+z)/2;
s:=sqrt(p*(p-x)*(p-y)*(p-z)); end;
begin
a:=strtofloat(edit1.Text);
b:=strtofloat(edit2.Text);
c:=strtofloat(edit3.Text);
d:=strtofloat(edit4.Text);
e:=strtofloat(edit5.Text);
f:=strtofloat(edit6.Text);
Sq(a,b,c,s1); Sq(c,d,e,s2); Sq(b,e,f,s3); Sq(a,f,d,s4);
s:=s1+s2+s3+s4;
edit7.Text:='s='+floattostr(s);
end;

```

dastur kodini kiritamiz.



3.43-rasm.


Mustaqil bajarish uchun topshiriqlar

1. Ekranga 10 ta dastur soʻzini chiqaruvchi dastur tuzing.
2. 1 dan 20 gacha boʻlgan sonlarning koʻpaytmasini hisoblovchi dastur tuzing.
3. Kvadratni chapdan oʻnga harakatini koʻrsatadigan dastur tuzing.
4. Toʻgʻri toʻrtburchakni chapdan oʻnga harakatini koʻrsatadigan dastur tuzing.
5. $y = \sqrt{x+2x}$ funksiyaning qiymatini $-2 \leq x \leq 2$ oraliqda $\Delta x = 0,4$ qadam bilan, $3 \leq x \leq 5$ oraliqda $\Delta x = 0,5$; qadam bilan, $6 \leq x \leq 10$ oraliqda $\Delta x = 1$; qadam bilan hisoblang.
6. $A(5)$ vektor elementlarining oʻsib borish tartibidajoylashtiring.
7. $A(5)$ vektor elementlarini eng kichigini toppish dasturini chizing
8. $A(10)$ vektor elementlarini eng kattasini toppish dasturini chizing
9. $A(5)$ vektor elementlarini oʻsib borish tartibida joylashtiring
10. $Z(3,4)$ matritsaning har bir ustunidagi musbat elementlar sonidan tashkil topgan M vektorni hosil qiling.

IV BOB. DELPHI DASTURINING GRAFIK VA MULTIMEDIA IMKONIYATLARI

4.1. Grafik obektlar bilan ishlash


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

 - Image – 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 **Object Inspektorini** 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);



4.1-rasm.

Bu ob'ektning **Object Inspektorini** 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.

 - Shape - 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.

Delphi dasturlash tili 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 piksel koordinatalari (0,0). Koordinatalar yuqoridan pastga va chapdan o‘ngga qarab o‘sib boradi. Soha o‘lchovlarini Image komponentasining Height va Width xossalari va formaning ClientHeight va Clientwidth xossalari orqali aniqlash mumkin.

Qalam.

Qalam geometrik figuralarni chizish uchun ishlatiladi. Chiziq ko‘rinishi **Tren** ob’ektning quyidagi jadvalda ko‘rsatilgan xossalari orqali aniqlanadi:

4.1-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

4.2-jadval

Konstanta	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

4.3-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

4.4-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

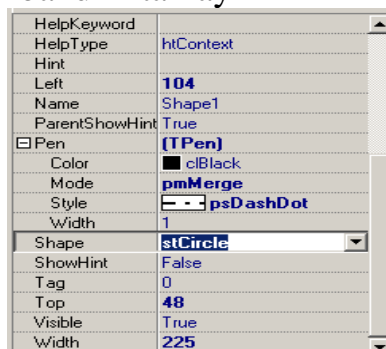
4.5-jadval

Konstanta	Soha bo'yash uslubi
bsSolid	Uzluksiz bo'yash
bsClear	Sohani bo'yalmaydi
bsHorizontal	Gorizantal shtrixlash
bsVertical	Vertikal shtirixlash
bsFDiagonal	Diogonal shtirixlash, oldinga og'ish
bsBDiagonal	Diogonal shtirixlash, orqaga og'ish
bsCross	Katakli gorizantal-vertikal shtirixlash
bsDiagCross	Katakli diogonal shtirixlash

Yuqorida aytilganlardan foydalangan holda bir nechta misollar ko'rib o'tamiz:

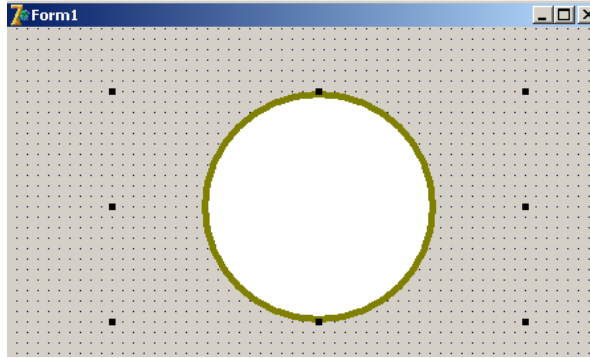
1-misol. Shape- tugmasidan foydalanib aylana hosil qiling.

1. Additional komponentalar palitrasidan Shape tugmachasini tanlaymiz.
2. **Object Inspektorining** Pen bandini tanlaymiz



4.2-rasm.

3. **Pen** qatorining **stCircle** bandini tanlaymiz– (aylana chizish buyrug‘ini tanlaymiz).



4.3-rasm.

4-Misol. Shrift o‘lchamlari va ranglardan foydalanuvchi dastur.

Bu masalani yechishimiz uchun quyidagi bosqichlarni amalga oshiramiz:

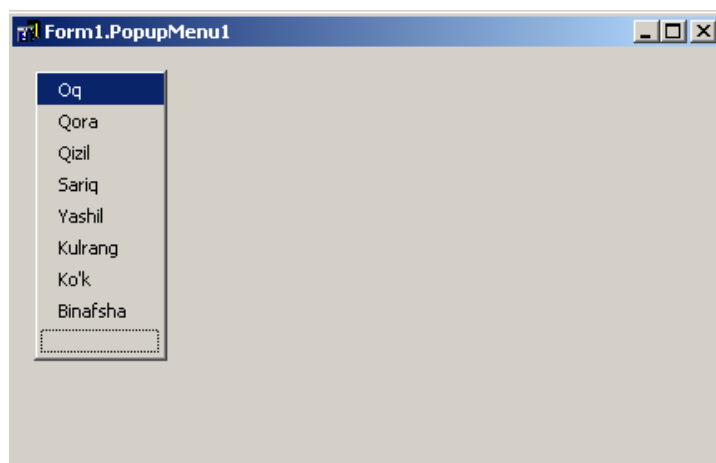
1. Standart komponentalar palitrasidan “Memo1” tugmasini va 6 ta “Label” tugmalarini hosil qilamiz;
2. Standart komponentalar palitrasidan “PopupMenu1” tugmasini va 16 ta “RadioButton1” tugmasini hamda 1 ta “GroupBox” tugmasini hosil qilamiz;
3. “GroupBox” maydoniga 7 ta “Button” tugmasini hosil qilamiz;
4. Win32 komponentalar palitrasidan “XPManifest1” tugmachalarini hosil qilamiz;
5. Standart komponentalar palitrasidan 6 ta “Button” tugmasini hosil qilamiz.

Yuqorida aytilgan topshiriqlarni quyidagi (4.4-rasm)da ko‘rsatilganidek joylashtiramiz.



4.4-rasm.

6. “PopupMenu” tugmasining dasturlash maydoniga quyidagi:



4.5-rasm.

7. Har bir yozuvlarning dasturlash maydoniga quyidagi:

```

procedure TForm1.Oq1Click(Sender: TObject);
begin
form1.Color:=clWhite; end;
procedure TForm1.Qora1Click(Sender: TObject);
begin
form1.Color:=clblack end;
procedure TForm1.Qizil1Click(Sender: TObject);
begin
form1.Color:=clred end;
procedure TForm1.Sariq1Click(Sender: TObject);
begin
form1.Color:=clyellow; end;
procedure TForm1.Yashil1Click(Sender: TObject);
begin
form1.Color:=clgreen; end;
procedure TForm1.Pushti1Click(Sender: TObject);
begin
form1.Color:=clpurple; end;
procedure TForm1.Kulrang1Click(Sender: TObject);
begin
form1.Color:=clgray; end;
procedure TForm1.Kok1Click(Sender: TObject);
begin
form1.Color:=clblue; end;
procedure TForm1.Binafsha1Click(Sender: TObject);
begin
form1.Color:=clpurple; end;
dastur kodini kiritamiz.

```

8. “Fon rangi” bo‘limidagi “RadioButton” tugmasining dasturlash maydoniga quyidagi:

```

procedure TForm1.RadioButton1Click(Sender: TObject);
begin

```



```

memo1.Color:=clWhite; end;
procedure TForm1.RadioButton2Click(Sender: TObject);
begin
memo1.Color:=clblack; end;
procedure TForm1.RadioButton3Click(Sender: TObject);
begin
memo1.Color:=clred; end;
procedure TForm1.RadioButton4Click(Sender: TObject);
begin
memo1.Color:=clblue; end;
procedure TForm1.RadioButton5Click(Sender: TObject);
begin
memo1.Color:=clgreen; end;
procedure TForm1.RadioButton6Click(Sender: TObject);
begin
memo1.Color:=clyellow; end;
procedure TForm1.RadioButton7Click(Sender: TObject);
begin
memo1.Color:=clpurple; end;
procedure TForm1.RadioButton8Click(Sender: TObject);
begin
memo1.Color:=clgray; end;
dastur kodini kiritamiz.

```

9. “Shrift rangi” bo‘limidagi “RadioButton” tugmasining dasturlash maydoniga quyidagi:

```

procedure TForm1.RadioButton9Click(Sender: TObject);
begin
memo1.Font.Color:=clWhite; end;
procedure TForm1.RadioButton10Click(Sender: TObject);
begin
memo1.Font.Color:=clblack; end;
procedure TForm1.RadioButton11Click(Sender: TObject);
begin
memo1.Font.Color:=clred; end;
procedure TForm1.RadioButton12Click(Sender: TObject);
begin
memo1.Font.Color:=clblue; end;
procedure TForm1.RadioButton13Click(Sender: TObject);
begin
memo1.Font.Color:=clgreen; end;
procedure TForm1.RadioButton14Click(Sender: TObject);
begin
memo1.Font.Color:=clyellow; end;
procedure TForm1.RadioButton15Click(Sender: TObject);
begin

```

```
memo1.Font.Color:=clpurple; end;  
procedure TForm1.RadioButton16Click(Sender: TObject);  
begin  
memo1.Font.Color:=clgray; end;  
dastur kodini kiritamiz.
```

10. “Shrift o‘lchami” bo‘limidagi “RadioButton” tugmasining dasturlash maydoniga quyidagi:

```
procedure TForm1.RadioButton17Click(Sender: TObject);  
begin  
memo1.Font.Size:=8; end;  
procedure TForm1.RadioButton18Click(Sender: TObject);  
begin  
memo1.Font.Size:=10; end;  
procedure TForm1.RadioButton19Click(Sender: TObject);  
begin  
memo1.Font.Size:=12; end;  
procedure TForm1.RadioButton20Click(Sender: TObject);  
begin  
memo1.Font.Size:=14; end;  
procedure TForm1.RadioButton21Click(Sender: TObject);  
begin  
memo1.Font.Size:=16; end;  
procedure TForm1.RadioButton22Click(Sender: TObject);  
begin  
memo1.Font.Size:=18; end;  
procedure TForm1.RadioButton23Click(Sender: TObject);  
begin  
memo1.Font.Size:=20; end;  
dastur kodini kiritamiz.
```

11. “Shrift turi” bo‘limidagi “Button” tugmasining dasturlash maydoniga quyidagi:

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
memo1.Font.Style:=memo1.Font.Style+[fsbold];  
button2.Visible:=true;  
button1.Visible:=false;  
label3.Visible:=true; end;  
procedure TForm1.Button2Click(Sender: TObject);  
begin  
memo1.Font.Style:=memo1.Font.Style-[fsbold];  
button1.Visible:=true;  
button2.Visible:=false;  
label3.Visible:=false; end;  
procedure TForm1.Button3Click(Sender: TObject);  
begin  
memo1.Font.Style:=memo1.Font.Style+[fsitalic];
```

```

button4.Visible:=true;
button3.Visible:=false;
label4.Visible:=true; end;
procedure TForm1.Button4Click(Sender: TObject);
begin
memo1.Font.Style:=memo1.Font.Style-[fsitalic];
button3.Visible:=true;
button4.Visible:=false;
label4.Visible:=false; end;
procedure TForm1.Button5Click(Sender: TObject);
begin
memo1.Font.Style:=memo1.Font.Style+[fsunderline];
button6.Visible:=true;
button5.Visible:=false;
label5.Visible:=true; end;
procedure TForm1.Button6Click(Sender: TObject);
begin
memo1.Font.Style:=memo1.Font.Style-[fsunderline];
button5.Visible:=true;
button6.Visible:=false;
label5.Visible:=false; end;
dastur kodini kiritamiz.

```



4.6-rasm.

5-misol. Geometrik shakllar hosil qilish

Chiziq chizishda forma oynasida chiziqning joylashishi va koordinatalar to‘rini ko‘raylik.

Standart komponentalar palitrasidan “Button1” tugmasini hosil qilamiz va “Button1” tugmasining dastur maydoniga quyidagi:

```

procedure TForm1.Button1Click(Sender: TObject);
var

```

```

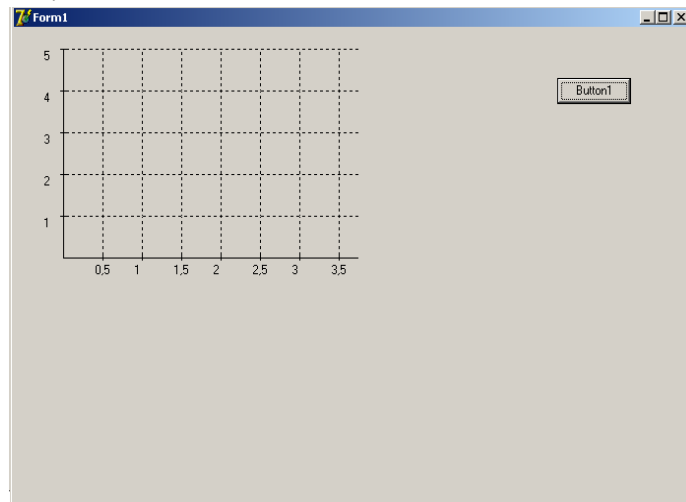
x0,y0:integer; // koordinata o'qlarining boshlang'ich koordinatalari
dx,dy:integer; // qadam koordinata to'rida (pikselda)
h,w:integer; // koordinata to'rini kiritish maydoning bo'yi va eni
x,y:integer;
lx,ly:real; // belgi (raqamlashda) x va y to'r chiziqlari
dlx,dly:real; // belgida qadam (raqamlashda) x va y to'r chiziqlari
cross:integer; // raqamlanmagan to'r chiziqlarini hisoblagich
dcross:integer; // raqamlangan va raqamlanmagan chiziqlar soni
begin
x0:=50; y0:=220; // o'qlarning boshlang'ich nuqtalari (50,220)
dx:=40; dy:=40; // koordinat to'rida shar 40 pikseldan
dcross:=1; // panjara chiziqlarini belgilash X: 1 — har biri;
// 2 — bitta orqali;
// 3 — ikkita orqali;
dlx:=0.5; // x o'qining qadam belgisi
dly:=1.0; // y o'qining qadam belgisi, belgilashda bajariladi: 1, 2, 3 va h.
h:=200; w:=300;
with form1.Canvas do begin
cross:=dcross;
MoveTo(x0,y0); LineTo(x0,y0-h); // o'q X
MoveTo(x0,y0); LineTo(x0+w, y0); // o'q Y
x:=x0+dx;
lx:=dlx;
repeat
MoveTo(x,y0-3);LineTo(x,y0+3);
cross:=cross-1;
if cross = 0 then // raqamlash
begin
TextOut(x-8,y0+5,FloatToStr(lx));
cross:=dcross ;
end;
Pen.Style:=psDot;
MoveTo(x,y0-3);LineTo(x,y0-h); // to'r chizig'i
Pen.Style:=psSolid;
lx:=lx+dlx;
x:=x+dx;
until (x>x0+w);
// Y o'qini belgilash
y:=y0-dy;
ly:=dly;
repeat
MoveTo(x0-3,y);LineTo(x0+3,y);
TextOut(x0-20,y,FloatToStr(ly)); // raqamlash
Pen.Style:=psDot;
MoveTo(x0+3,y); LineTo(x0+w,y); // to'r chizig'i

```

```

Pen.Style:=psSolid;
y:=y-dy;
ly:=ly+dly; until (y<y0-h);
end; end;
dastur kodini kiritamiz.

```



4.7-rasm.

Kiritilga dasturning murakkab joyi bu dasturning forma oynasini (forma koordinatalariga) to‘r va qadamlar bilan belgilashdir. Shu bilan birga dastur x va y o‘qlarning joylashishi, ularning son bilan belgilanishini va to‘rda nafaqat bitta balki bir nechta chiziq chizish imkoniyatiga ega.

6-Misol. Tutashgan chiziq chizish dasturi.

O‘zaro tutashgan kesmalardan iborat shaklni chizish uchun polyline usulidan foydalaniladi. Bu usul parametri TPoint tipli massivdan iborat.

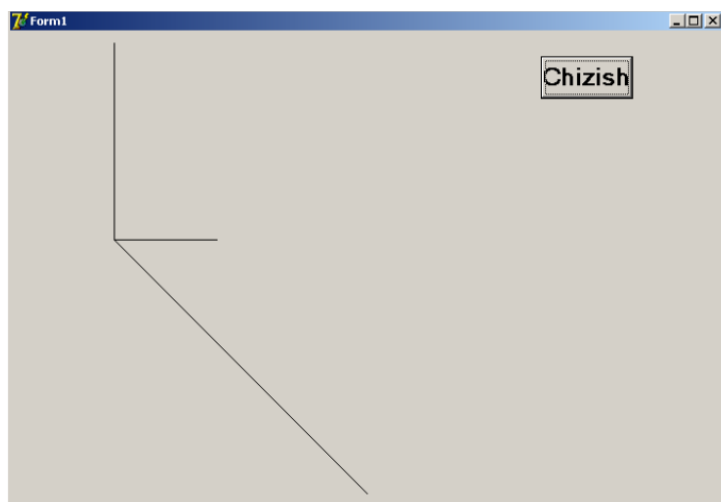
Tutash chiziqni forma oynasiga chiqarish uchun “Button1” tugmasidan foydalanamiz.

“Button1” tugmasining dasturlash maydoniga quyidagi:

```

var
gr: array[1..50] of TPoint;
x0,y0: integer;
dx,dy: integer;
i: integer; begin
x0 := 100; y0 := 200;
dx :=5; dy := 5;
for i:=1 to 50 do begin
gr[i].x := x0 + (i-1)*dx;
gr[i].y := y0 + (i-1)*dy;
end;
with form1.Canvas do begin
MoveTo(x0,y0); LineTo(x0,10);
MoveTo(x0,y0); LineTo(200,y0);
Polyline(gr);
end; end;
dastur kodini kiritamiz.

```



4.8-rasm.

Polyline usuli yordamida yopiq ko'pburchak chizish uchun massivning birinchi va oxirgi elementi bir nuqtaning koordinatalaridan iborat bo'lishi kerak.

7-misol. Ko'pburchak chizing.

Polygon usuli ko'pburchak chizishga mo'ljallangan bo'lib, parametri TPoint tipidagi massivdir. Massivning har bir elementi (x,u) maydonlari ko'pburchak uchi koordinatalaridan iborat bo'lgan yozuvdir.

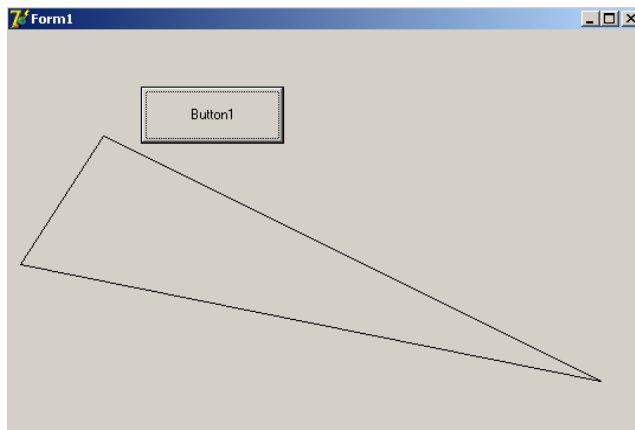
Yuqoridagilarni tasdiqlash uchun biz uchburchak chizishni ko'rib chiqamiz:

Uchburchakni forma oynasiga chiqarish uchun **Button1** tugmasidan foydalanamiz.

Button1 tugmasining dasturlash maydoniga quyidagi:

```
var
pol: array[1..3] of TPoint;
begin
pol[1].x := 10;
pol[1].y := 100;
pol[2].x := 10;
pol[2].y := 10;
pol[3].x := 70;
pol[3].y := 100;
Form1.Canvas.Polygon(pol); end;
```

dastur kodini kiritamiz.



4.9-rasm.

8- misol. $y = x$ funksiyaning grafigini hosil qilish.

Bu grafikning forma oynasida hosil qilish uchun, **Button1** tugmasini hosil qilamiz va dasturlash maydoniga quyidagi:

```
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;
```

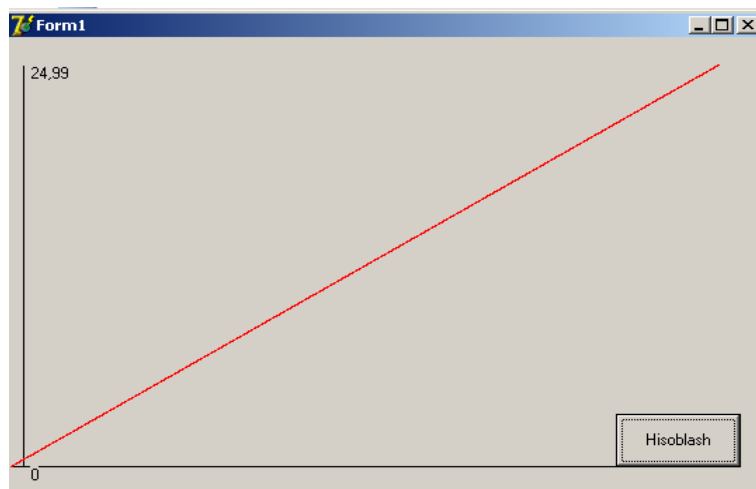
```
dx:=0.01;
```

```
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;
dastur kodini kiritamiz.

```



4.10-rasm.

9-misol. $y = \sin^2 x$ funksiya grafigini chizish.

Dastur kodi va oynaning umumiy ko‘rinishini keltiramiz:


```

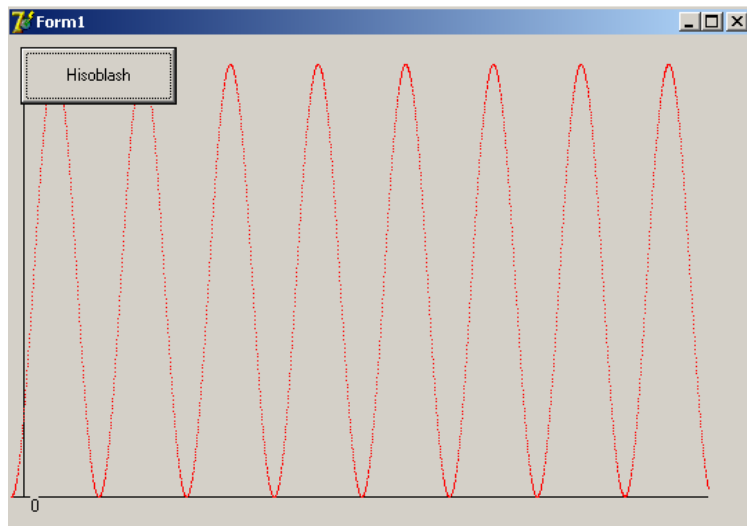
unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm1 = class(TForm)
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;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

```



4.11-rasm.

Mustaqil bajarish uchun topshiriqlar

1. Forma oynasini har xil ranglarga bo'yovchi dastur tuzing.
2. Memo tugmasining ichini har xil ranglarga bo'yovchi dastur tuzing.
3. Memo tugmasidagi yozuvlarning o'lchamlarini o'zgartiruvchi dastur tuzing.
4. Shrift rangini o'zgartiruvchi dastur tuzing
5. Ellips tasvirini hosil qiling.
6. To'rtburchak hosil qiling.
7. Kvadrat hosil qiling.
8. To'g'ri chiziq hosil qiling.
9. Olti burchak hosil qiling.
10. Quyidagi funksiyalarning grafiklarini hosil qiling:
 1. $y = 2x - 5$
 2. $y = \sqrt{x}$
 3. $f = 2 \sin x \cos(2x - 1)e^{\frac{3x}{7}}$

4.2. Multimedia dasturlarini yaratish vositalari

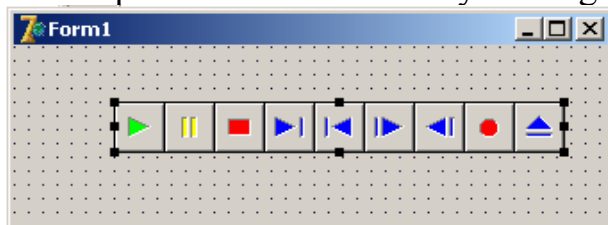
Multimedia–kompyuterda ishlatiladigan harakat (animatsiya), ovozlari, videotasvirlarning hamma ko‘rinishi bilan bog‘liq tushunchadir. Delphida “MediaPlayer” komponenti mavjud bo‘lib, u multimedia dasturlarini tuzishda hamma imkoniyatlarni yaratib beradi. Ushbu komponentdan foydalanishning ikki xil usuli mavjud:

- ixtiyoriy dasturchiga multimedia dasturini tuzishga imkon yaratilgan;
- komponentda multimedia fayllari bilan bajariladigan hamma asosiy amallar kiritilgan, lekin sizga quyi funksiyalar kerak bo‘ladigan bo‘lsa, Delphi dasturlash tili imkoniyatlaridan foydalanishingiz mumkin.

Ushbu mavzuvda komponenta ishlaganda uning ichki funksiyalari qanday ishlatilishini ko‘rib o‘tamiz.

MediaPlayer komponenti

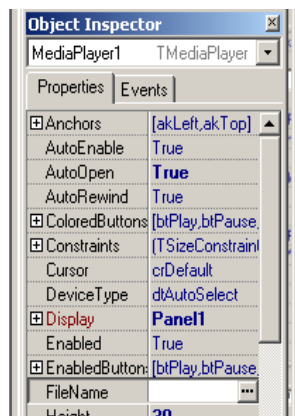
System komponentalar palitrasidan “MediaPlayer1” tugmasini hosil qilamiz.



4.12-rasm.

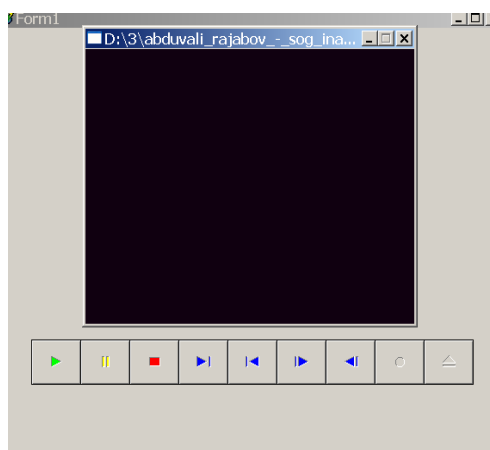
“MediaPlayer1” tugmasida “ishga tushirish”, “o‘tkazish”, “yozish” va boshqa tugmalari bor.

Tugma hosil qilingandan so‘ng, siz **Object Inspektor**ida **FileName** xususiyatini ko‘rasiz.



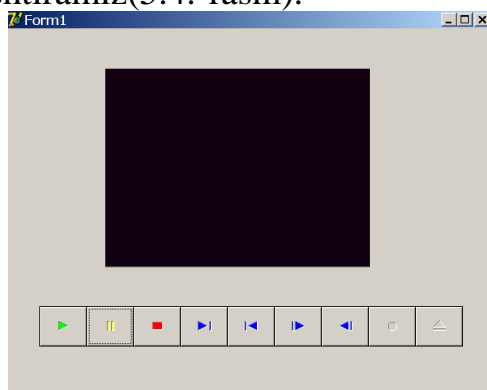
4.13-rasm.

Shu yerda sichqoncha tugmasi tanlanadi va ro‘yxatdan “.mpg” kengaytmani faylni belgilanadi. So‘ngra **AutoOpen** xususiyatini **True** qiymatga o‘rnatish kerak. Bu qadamlar bajarilgach, dastur yuklanishga tayyor bo‘ladi. Dasturni yuklab ishga tushirish (Play) tugmasini tanlanadi va siz tanlagan videoklipingizni ko‘rishingiz yoki musiqani tinglashingiz mumkin(4.14-rasm).



4.14-rasm.

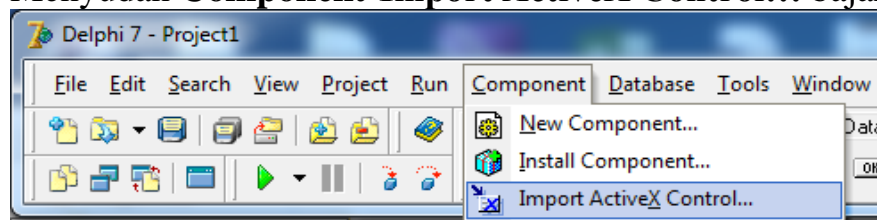
MediaPlayer komponentining yana bir muhim xususiyati –**Display**. Dastlabki misolimizda bu xususiyat bo‘sh qoldirilgan va videoklip aloxida oynada ko‘ringan edi. Lekin, videoklip namoyishida ekran sifatida masalan, **Panel**dan foydalanish mumkin. **Standart** komponentalar palitrasidan **Panel1** tugmasini hosil qilamiz va **Caption** xususiyatidagi matnni o‘chiramiz. So‘ngra **MediaPlayer1** uchun **Display** xususiyatidagi ro‘yxatdan **Panel1**ni tanlang. Shundan so‘ng dasturni yuklab, ishga tushirish tugmasini faollashtiramiz(5.4.-rasm).



4.15-rasm.

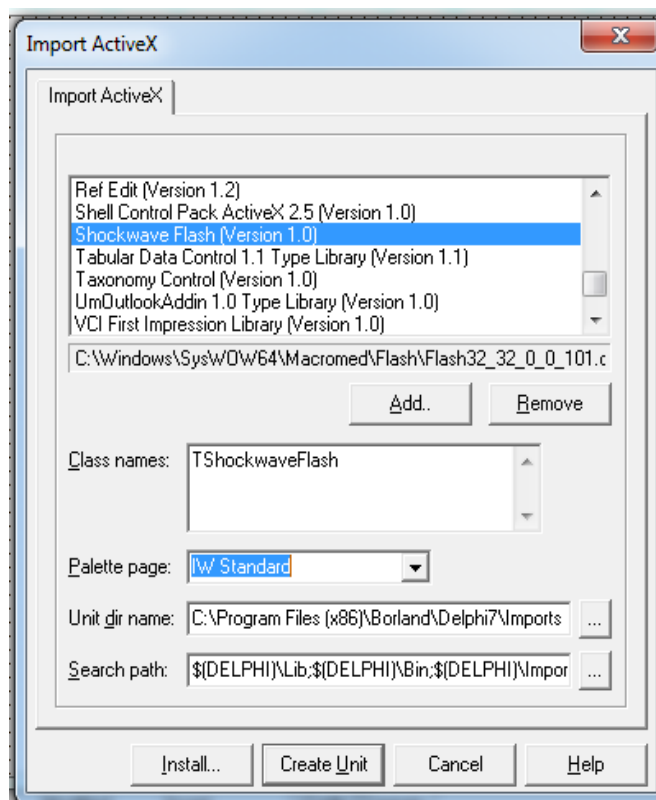
Delphining yanam bir imkoniyati shundan iboratki, **.swf** fayl kengaytmali animatsiyalarni ham joylashtirish imkoni mavjud. Buning uchun Komponentalar palitrasiga **ShockwaveFlash** (ShockwaveFlashObjects_TLB) komponentini o‘rnatish lozim bo‘ladi. Buning uchun quyidagi amallar ketma –ketligi bajariladi.

Menyudan **Component-Import ActiveX Control...** bajariladi:



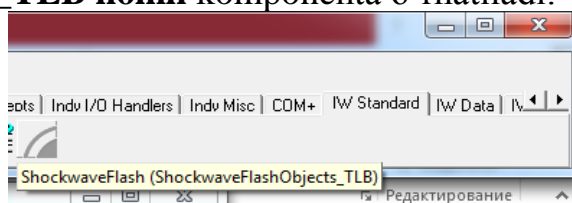
4.16-rasm.

Natijada, quyidagi oyna hosil bo‘ladi:



4.17-rasm.

Ushbu oynadan **Shockwave Flash (Version 1.0)** bandi tanlanib, **Palette Page** qatoridan joylashtirish lozim boʻlgan komponenta tanlanadi. Masalan: **IW Standard** komponentasi belgilanib, **Install..** tugmasi bosilib, navbatdagi oynaga oʻtib, **OK** tugmasi bosiladi. Natijada, **IW Standard** komponentasida **ShockwaveFlashObjects_TLB** nomli komponenta oʻrnatiladi:



4.18-rasm.

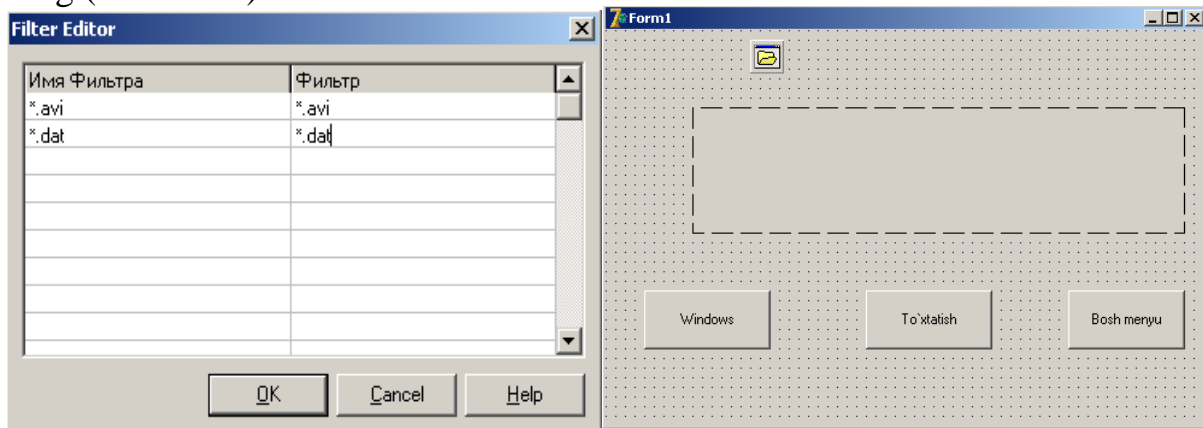
Mazku komponentadan foydalanib, **.swf** fayl kengaytmali animatsiya effektlarining joylashtirish imkoniyati tugʻiladi. Buning uchun esa **ShockwaveFlashObjects_TLB** komponentasi tanlani, forma oynasiga joylashtirilib, **object Inspector** xossasidan **Movie** qatoriga **.swf** fayl kengaytmali animatsiya effekti turgan joy yoziladi. Masalan: *C:\Users\hasee\Desktop\3333\Burchak bissiktrisasi.swf*

1-masala. Delphi dasturlash tilida ovozsiz animatsiyalar bilan ishlovchi ilova yaratish.

Bu masalani yechishimiz uchun quyidagi bosqichlarni amalga oshiramiz:

1. “Win32” komponentalar palitrasidan “Animate1” tugmasini hosil qilamiz.
2. **Object Inspektor**ning “Visible” xossasiga “false”, “Repetitions” xossasiga “0”, “Active” xossasiga “false”, “AutoSize” xossasiga “false”, “Center” xossasiga “true” buyruqlarini kiriting.
3. Standart komponentalar palitrasidan “Button1”, “Button2”, “Button3” tugmalarini hosil qilamiz.

4. “Button1”, tugmasining “Name” xossasiga “Vwind”, “Caption” xossasiga “Windows” jumllarini kiriting.
5. “Button2”, tugmasining “Name” xossasiga “BStop”, “Caption” xossasiga “To‘xtatish” jumllarini kiriting.
6. “Button3”, tugmasining “Name” xossasiga “BFile”, “Caption” xossasiga “Bosh menyu” jumllarini kiriting.
7. Dialog komponentalar palitrasidan “OpenDialog1” tugmasini hosil qilamiz.
8. “OpenDialog1” tugmasining “Filter” xossasiga “*.avi”, “*.dat” buyrug‘larini kiriting (4.19-rasm).



4.19-rasm.

9. Form oynasining dasturlash maydoniga quyidagi dasturni kiritamiz:

```
var i:word;
    Form1: TForm1;
implementation
{$R *.dfm}
```

10. “Button1” tugmasining dasturlash maydoniga quyidagi dasturni kiritamiz:

```
begin
Animate1.Visible := true;
i := 1;
Animate1.CommonAVI := aviFindFolder;
Animate1.Active := true;
end;
```

11. “Button2” tugmasining dasturlash maydoniga quyidagi dasturni kiritamiz:

```
begin
Animate1.Stop; end;
```

12. “Button3” tugmasining dasturlash maydoniga quyidagi dasturni kiritamiz:

```
begin
if OpenFileDialog1.Execute then with Animate1 do
begin
i := 9;
FileName := OpenFileDialog1.FileName;
Visible := true;
Active := true;
end;
```

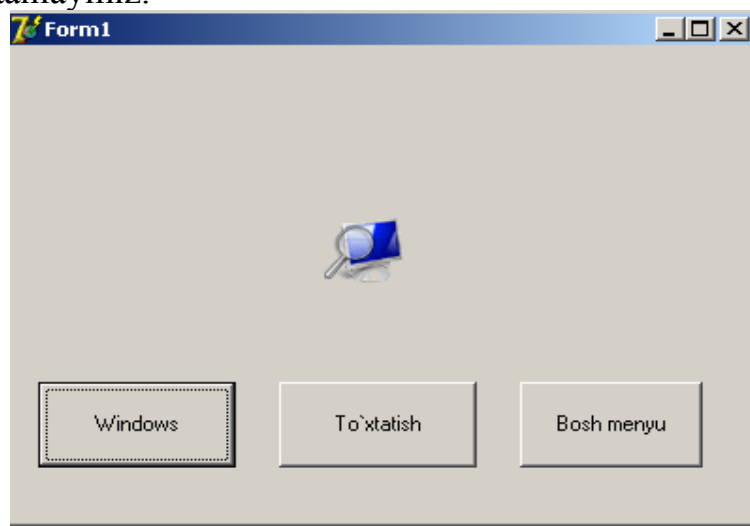
13. “Animate1” tugmasining dasturlash maydoniga quyidagi dasturni kiritamiz:

```

begin
Inc(i);
with Animatel do begin
case i of
2: CommonAVI := aviFindFile;
3: CommonAVI := aviFindComputer;
4: CommonAVI := aviCopyFiles;
5: CommonAVI := aviCopyFile;
6: CommonAVI := aviRecycleFile;
7: CommonAVI := aviEmptyRecycle;
8: CommonAVI := aviDeleteFile;
end;
if i < 9 then Active := true
else Visible := false;
end; end;

```

14. “F9” tugmasini tanlaymiz.



4.20-rasm.

Yuqoridagi misollardan ko‘rinib turibdiki, Delphining dasturlash tiliga videoroliklarni, tovushli ma’lumotlarni, animatsiya effektlarini joylashtirish, ko‘rish, eshitish va ular uchun vizvual loyihalar yaratishda foydalanish mumkin.

Mustaqil bajarish uchun topshiriqlar

1. Delphining forma oynasiga ovozli ma’lumotlarni va videoroliklarni joylashtiring.
2. Macromedia Flash dasturida birona animatsiyali rolik tayyorlab, uni **.swf** fayl kengaytma bilan saqlang va unin Delphining forma oynasiga joylashtiring.
3. iSpring dasturida test tuzing va uni **.swf** fayl kengaytma bilan saqlab, Delphining forma oynasiga joylashtiring.

V BOB. DELPHI VIZUAL DASTURLASH VOSITASIDA MA'LUMOTLAR BAZASINI YARATISH TEXNOLOGIYALARI

Ma'lumotlar bazasi (MB) – bu fayl ko‘rinishidagi turli xildagi axborotlar jamlanmasidir. Ma'lumotlar bazasi uchun dastur tuzish axborotlarni qayta ishlashni ta'minlaydi. Bunday dasturlar ishga yuklanganida foydalanuvchi undan o'zini qiziqtirgan ma'lumotlarni olishi, yangi yozuvlarni qo'shib keraksizini o'chirish imkoniyatiga ega bo'ladi.

Ma'lumotlarni saqlash, uzatish va qayta ishlash uchun MBni yaratish hamda undan keng foydalanish bugungi kunda dolzarb bo'lib qolmokda. Ma'lumotlar bazasini yaratish uchun foydalanuvchi axborotlarni turli belgilar yordamida tartiblashga va ixtiyoriy belgilar birikmasi bilan tanlanmani tez olishga intiladi. Buni faqat ma'lumotlar tizimalashtirilgan holda bajarish mumkin. Ma'lumotlar bazasining zamonaviy texnologiyasida ma'lumotlar bazasini yaratish, uni dolzarb holatda yuritishni va foydalanuvchilarga axborot olishini ta'minlovchi maxsus dasturiy vosita, ya'ni ma'lumotlar bazasini boshqarish tizimi yordami bilan markazlashtirilgan holda amalga oshirishni nazarda tutadi.

Ma'lumki ma'lumotlarni kiritish va ularni qayta ishlash jarayoni katta hajmdagi ish bo'lib, ko'p mehnat va vaqt talab qiladi. MB bilan ishlashda undagi ma'lumotlarning aniq bir tuzilmaga ega bo'lishi, foydalanuvchiga ma'lumotlarni kiritish va qayta ishlash jarayonida undagi ma'lumotlarni tartiblashtirish hamda kerakli ma'lumotlarni izlash va tez ajratib olish kabi qulayliklarni tug'diradi. Odatda MB ma'lum bir ob'ekt sohasini ifodalaydi va uning ma'lumotlarni o'z ichiga oladi, ularni saqlaydi va foydalanuvchiga ma'lumotlarni qayta ishlash uchun foydalanish imkonini yaratib beradi.

Bugungi kunda moliya, ishlab chiqarish, savdo-sotiq, tibbiyot va boshqa sohada ish yuritishda ma'lumotlar bazasiz tasavvur qilib bo'lmaydi. Shu bois mazku bobda Delphi muhitida MB yaratish texnologiyalarini ko'rib chiqamiz.

5.1. Delphi muhitida DataAccess va ADO komponentalar panelidan foydalanish

Delphi dasturlash tilida ma'lumotlar bazasini o'zaro bog'liqlikda tashkil qilish va uni boshqarish masalalari bo'yicha yuqori imkoniyatlarga egadir. Ma'lumotlar bazasida bir nechta ilovalar bilan birgalikda ishlash mumkin. Ko'pgina ma'lumotlar bazasi **BDE** va **ADO** da **ODBC** boshqaruvsiz ishlovchi maxsus drayverlarga egadir. Ushbu paragrafda o'zaro bog'liq ma'lumotlar bazasini **ADO** (Active Data Objects) komponenti asosida tuzamiz.

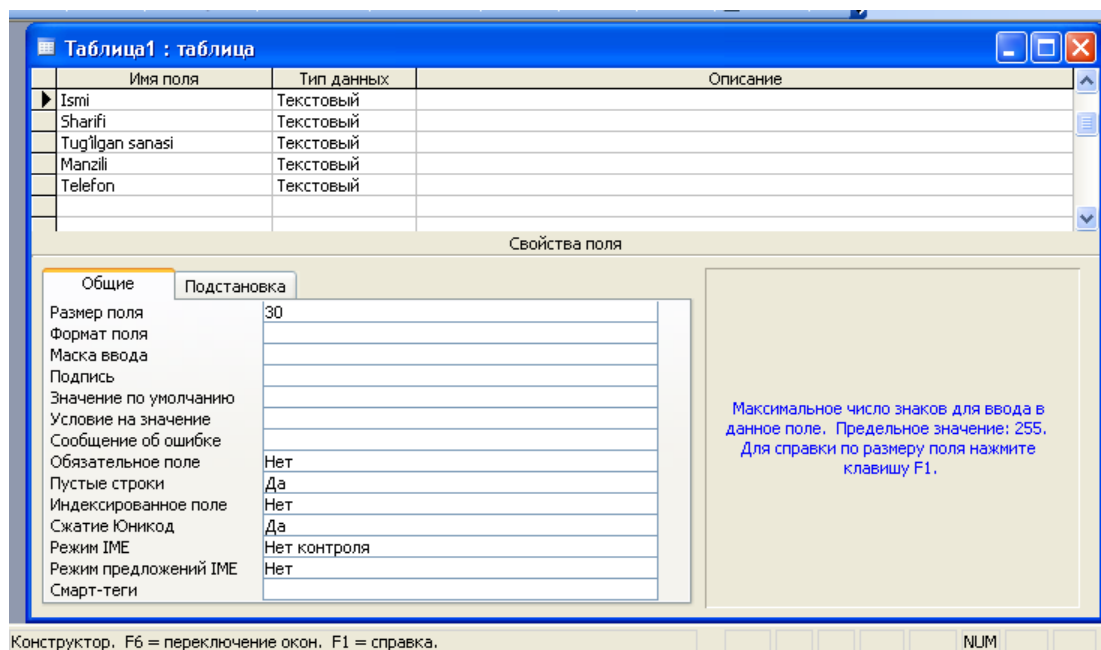
Ishni bajarish bosqichlari: **MS-Access** dasturi yordamida ma'lumotlar bazasini tuzish; Dasturli ma'lumotlar bazasi bilan bog'liqligini tashkil qilish.

MS-Access dasturida jadval hosil qilish

MS-Access dasturi ishga tushuriladi. **Пуск-Программы-Microsoft Access**. Hosil bo'lgan oynaning o'ng tomonidan **Созданий база данных-Новая база данных** buyrug'i tanlanib, faylni biror papkaga "db1.mdb" nom bilan saqlanadi.

Hosil bo‘lgan ketma-ketlikdan so‘ng, **Создание таблицы в режиме конструктора** rejimi tanlanadi va quyidagi ko‘rinishda jadval hosil qilinadi (6.1-rasmga qarang):

Maydon (polya) nomi	Ma'lumotlar tipi	Formati
Familyasi	Текстовый	30 simvolov
Ismi	Текстовый	30 simvolov
Sharifi	Текстовый	30 simvolov
Tug'ilgan sanasi	Текстовый	30 simvolov
Manzili	Текстовый	30 simvolov
Telefon	Текстовый	30 simvolov



5.1-rasm.

Jadvalni “Infor1” nom bilan saqlab, quyidag ko‘rinishda to‘ldiriladi (5.2-rasmga qarang):

	Код	Familyasi	Ismi	Sharifi	Tug'ilgan sanas	Telefoni
▶	1	TOSHEVA	XOLIDA	Sultonova	11.01.1994	+998944398292
	2	MAHMUDOV	SULTONBEK	YUSUFJONOVICH	18.01.1992	+998933724422
	3	NAGIMOVA	FERUZA	RASHID QIZI	10.09.1992	+998942299292
	4	USMONOVA	SHOIRA	TOHIR QIZI	05.02.1994	+998933165258

5.2-rasm.

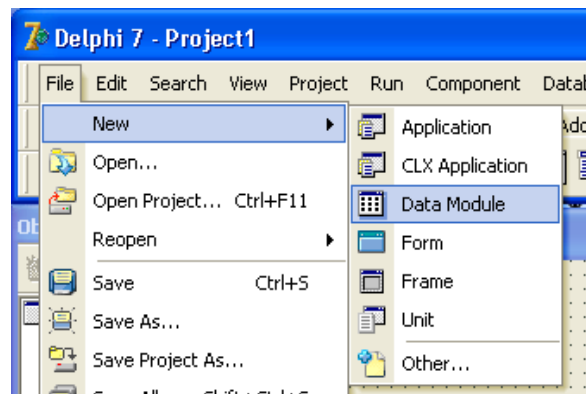
Jadval to‘ldirilgandan so‘ng, **MS-Access** dasturi yopiladi.

Delphi dasturlash tili muhiti bilan ishlash

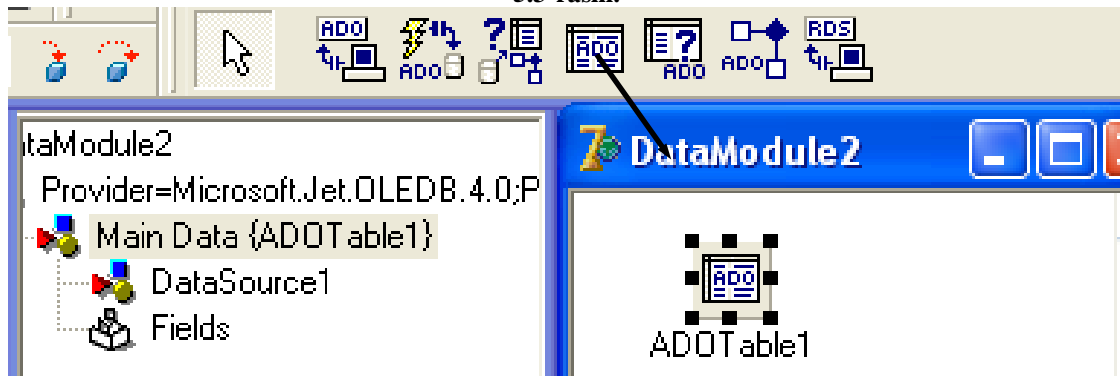
Masalaning birinchi bosqichi

Delphi dasturi ishga tushiriladi. Пуск→Программы→Borland Delphi7→Delphi7. Dastur yuklangandan so‘ng, **MS-Access**da “db1.mdb” nom bilan yaratilgan fayl kompyuter xotirasiga saqlanadi.

Ma'lumotlar ustida effektiv ravishda ishlash uchun maxsus ma'lumotlar moduli **Data Module** tuziladi. Buning uchun **File→New→Data Module** modul oynasining **Components** qismiga **ADO** komponentlar panelidagi **ADOTable** tugmasi joylashtiriladi (5.3, 5.4-rasmga qarang):

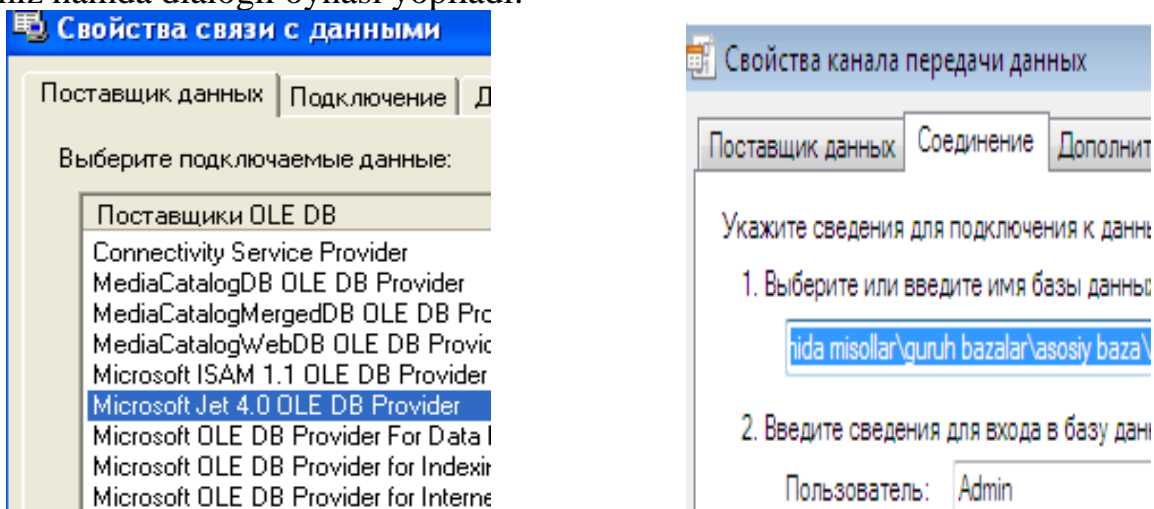


5.3-rasm.



5.4-rasm.

ADOTable tugmasining **Connection String** xususiyatiga bog‘lash qatoriga **Connection String** dialogi chaqiriladi va **Build...** tugmasi tanlanadi. So‘ngra **Microsoft.Jet.OLEDB.4.0** (MS-Access ma’lumotlar bazasiga kirishga ruxsat beruvchi yo‘l) ma’lumotlar tipi tanlanadi (5.5-rasmga qarang.) **Connection** qismining birinchi qatoriga “db1.mdb” (siz tuzgan ma’lumotlar bazasi fayli nomi) fayli nomi yoziladi. **Blank password** qismiga ✓ belgisi qo‘yiladi va bog‘lanishni tekshirib ko‘ramiz hamda dialogli oynasi yopiladi.

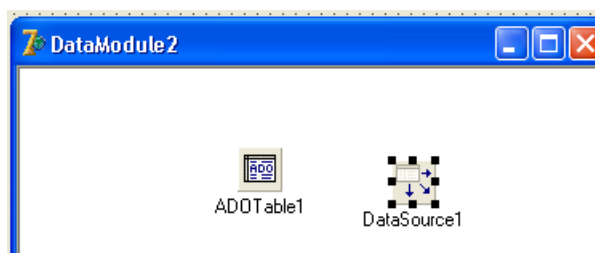


5.5-rasm.

ADOTable1 tugmasi uchun quyidagi xususiyatlar o‘rnatiladi:

- *Active-True;*
- *TableName- Infor1.*

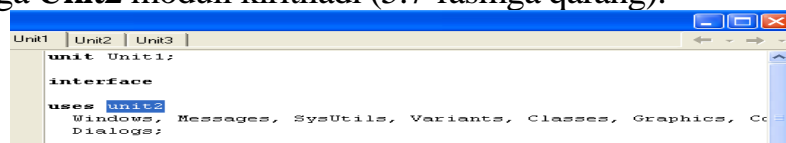
Hosil bo‘lgan ketma-ketlikdan so‘ng, **DataModule2** ga **DataAccess** komponentlar panelidagi **DataSource** tugmasi joylashtiriladi (5.6-rasmga qarang).



5.6-rasm.

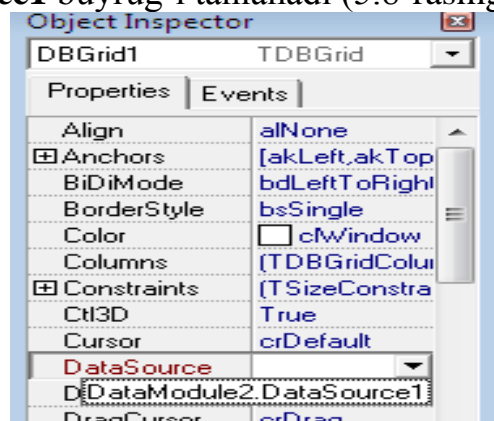
Bu tugma uchun **DataSet** xususiyatiga **ADOTable1** buyrug‘i joylashtiriladi. **DataSource** tugmasi ma’lumotlar bazasi bilan undagi elementlarni aks ettirish orasida uzatuvchilik funksiyasini bajaradi.

Unit1 modulini dasturlash maydoniga **F12** tugmasi orqali o‘tiladi va uning **Uses** qismiga **Unit2** moduli kiritiladi (5.7-rasmga qarang).



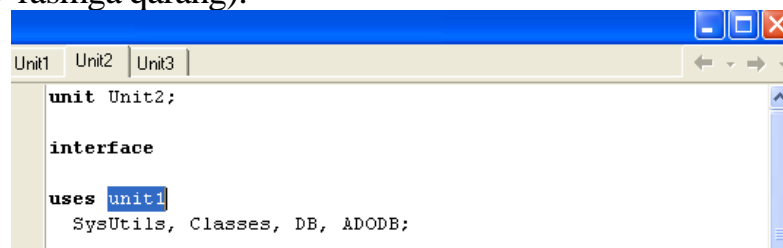
5.7-rasm.

Yuqoridagi amallar bajarilgandan so‘ng, **F12** tugmasi yordamida **Form1** oynasiga qaytiladi. **Data Controls** komponentlar panelidagi **DBGrid** tugmasi form oynasiga joylashtiriladi va **Object inspector** dan **DataSource** bandidan **DataModule2.DataSource1** buyrug‘i tanlanadi (5.8-rasmga qarang).



5.8-rasm.

Bajarilgan amallar ketma-ketligidan so‘ng, **F12** tugma yordamida **Unit1** modulini dasturlash maydoniga o‘tiladi hamda **Uses** qismiga kiritgan **Unit2** moduli o‘chiriladi va **Unit2** modulini dasturlash maydoniga o‘tib, uning **Uses** qismiga **Unit1** moduli kiritiladi (5.9-rasmga qarang).



5.9-rasm.

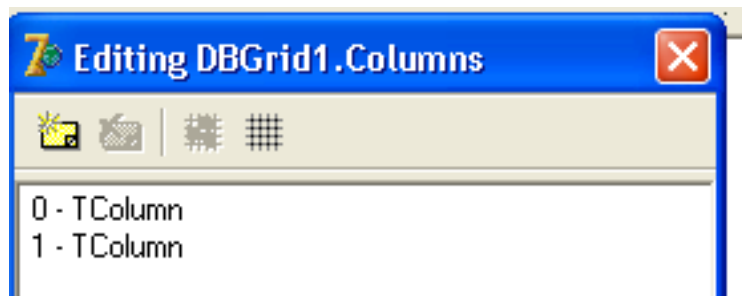
Natijada, quyidagi ko‘rinish hosil bo‘ladi (5.10-rasmga qarang):

	Код	Familyasi	Ismi	Sharifi	Tug'ilgan sanasi
▶	1	TOSHEVA	XOLIDA	Sultonova	11.01.1994
	2	MAHMUDOV	SULTONBEK	YUSUFJONOVICH	18.01.1992
	3	NAGIMOVA	FERUZA	RASHID QIZI	10.09.1992
	4	USMONOVA	SHOIRA	TOHIR QIZI	05.02.1994
	5	QOSIMOV	TEMUR	SIROJIDDINOVICH	12.12.1983

5.10-rasm.

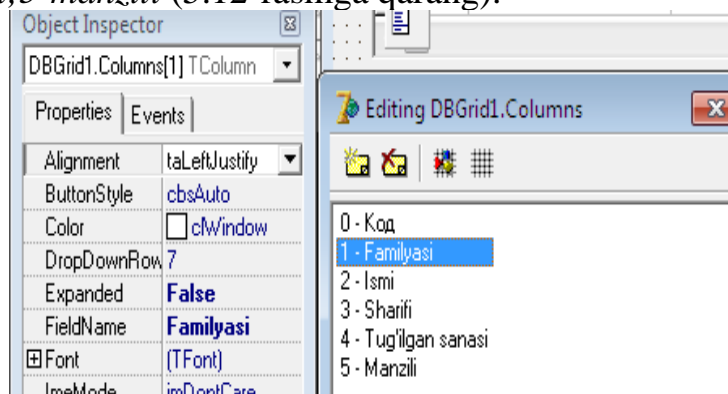
Masalaning Ikkinchi bosqichi

DBGrid1 tugmasi ustiga sichqoncha tugmasi tez-tez ikki marta bosiladi. Natijada, yangi loyiha oynasida **Column** yozuvlari paydo bo'ladi (5.11-rasmga qarang).



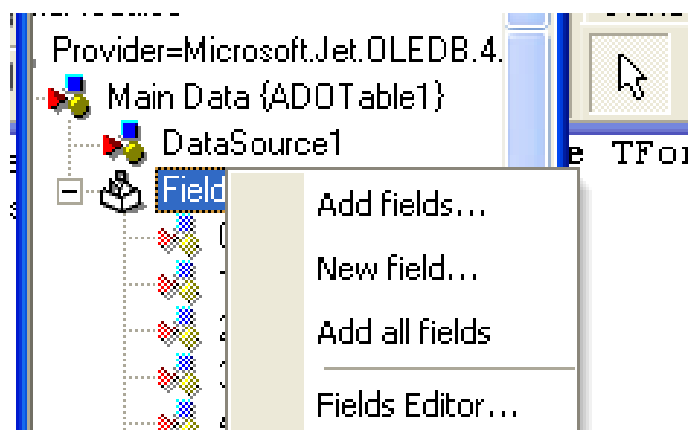
5.11-rasm.

Hosil bo'lgan oynaning **Add New** bandidan jadvalga beshta ustun qo'shiladi. So'ngra ustunlarning sarlavhalari nomini **Object inspector** ning **Title** bandini **Caption** bo'limiga quyidagi jumlar kiritiladi: *0-kod; 1-Familya; 2-Ismi; 3-Sharifi; 4-Tug'ilgan sanasi; 5-manzili* (5.12-rasmga qarang).



5.12- rasm.

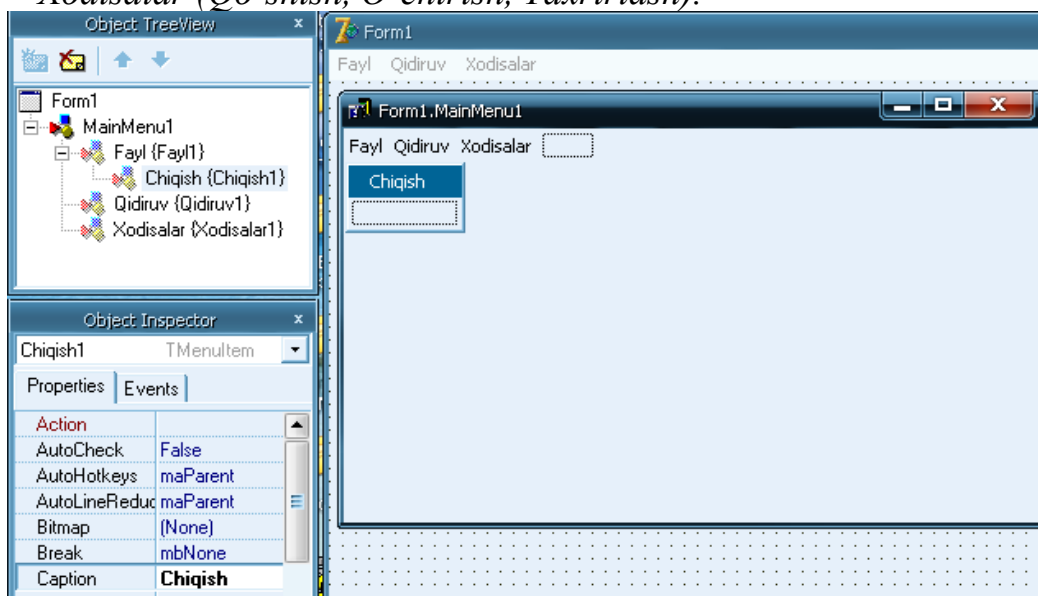
Form1 oynasiga **Data Controls** komponentlar panelidagi **TDBMemo** tugmasi joylashtiriladi va **Object inspector** ning **DataSource** bandidan **DataModule2.DataSource1** buyrug'i hamda **Data Fields** bandiga **Comment** qiymatlari o'rnatiladi. **Data Source1** ma'lumotlar modulining tarkibiga kiruvchi **Fields** bandiga sichqonchani o'ng tugmasi bosiladi va hosil bo'lgan kontekstli menyudan **Add all Fields** bandi tanlanadi (5.13-rasm qarang).



5.13 - rasm.

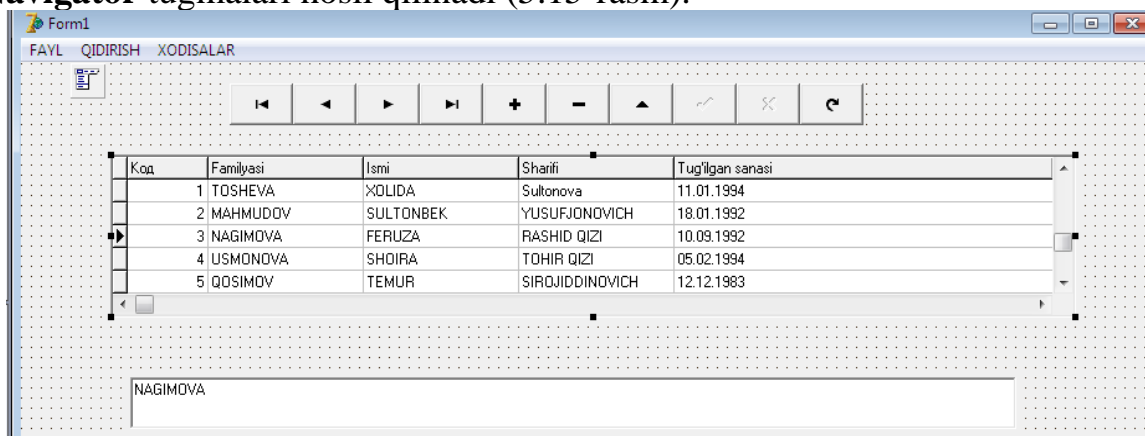
Form1 oynasiga **Standart** komponentlar panelidagi **MainMenu** tugmasi joylashtiriladi hamda **Object inspector** ning **Caption** bandiga quyidagi jumlar kiritiladi (6.14-rasmga qarang):

- *Fayl (chiqish);*
- *Qidirish (Sana bo'yicha, Telefon bo'yicha, Filterdan ozod qilish);*
- *Xodisalar (Qo'shish, O'chirish, Taxrirlash).*



5.14-rasm.

Form1 oynasiga **Data Controls** komponentlar panelidan **DBEdit** va **DBNavigator** tugmalari hosil qilinadi (5.15-rasm).



5.15-rasm.

Masalaning uchunchi bosqichi

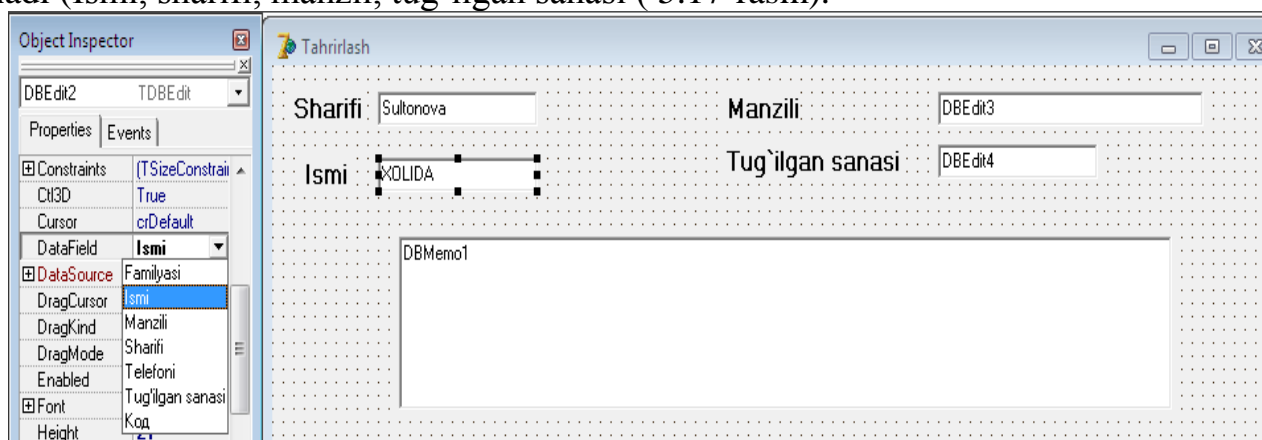
Yuqorida bajarilgan ketma-ketlikdan so‘ng, **Form3** oynasi hosil qilinadi.

Standart komponentalar palitrasidan 4 ta **Label** tugma, **Data Controls** komponentalar palitrasidan 4 ta **DBEdit**, 1 ta **DBMemo** va **Additional** komponentalar palitrasidan 2 ta **BitBtn** tugmalari hosil qilinadi hamda har bir **Label** tugmasining **Caption** bandiga mos ravishda “Ismi, sharifi, manzili va tug‘ilgan sanasi” jumalari kiritiladi (5.16-rasmga qarang).



5.16-rasm.

Tahrirlash oynasining **Uses** qismiga **Unit2** (ma'lumotlar moduli) moduli yoziladi va **Form3** oynasiga qaytib, oldin hosil qilingan 4 ta **DBEdit** tugmalarining **Object inspector** xossasini **DataSource** bandiga **DataModule2.DataSource1** tanlanadi va **DataFiled** xususiyatiga ma'lumotlar bazasidagi kerakli maydonlar (ustunlarni) hosil qilinadi (Ismi, sharifi, manzil, tug'ilgan sanasi) (5.17-rasm).



5.17-rasm.

Hosil bo‘lgan ketma-ketlikdan so‘ng, **Form1** oynasiga o‘tiladi va oldin hosil qilingan menyularning dasturlash maydonlariga mos ravishda dasturlar kiritiladi (5.18-rasmga qarang):



5.18-rasm.

1. Qo‘shish bandiga quyidagi dastur kodi kiritiladi:
procedure TForm1.QO‘SHISH1Click(Sender: TObject);

```

begin
{jadvalga yozuvni qo'shish}
DataModule2.Adotable1.Insert;
{Yangi yozuvga sana va vaqt qiymatlarini o'rnatish}
DataModule2.Adotable1.FieldName('Ismi').Value:= text;
DataModule2.Adotable1.FieldName('Sharifi').Value:= text;
DataModule2.Adotable1.FieldName('Manzili').Value:= text;
Form3.ShowModal;
end;

```

2. O'chirish bandiga quyidagi dastur kodi kiritiladi:

```

procedure TForm1.N8Click(Sender: TObject);
{Xodisalar / O'chirish}
begin
{Yozuvni o'chirish}
DataModule2.ADOTable1.Delete;
end;

```

3. Tahrirlash bandiga quyidagi dastur kodi kiritiladi:

```

procedure TForm1.N9Click(Sender: TObject);
{Xodisalar / Taxrirlash}
begin
if Form3.ShowModal=mrOK then
{ O'zgartirishlarni belgilab qo'shish }
DataModule2.ADOTable1.Post
else
DataModule2.ADOTable1.Cancel;
end;

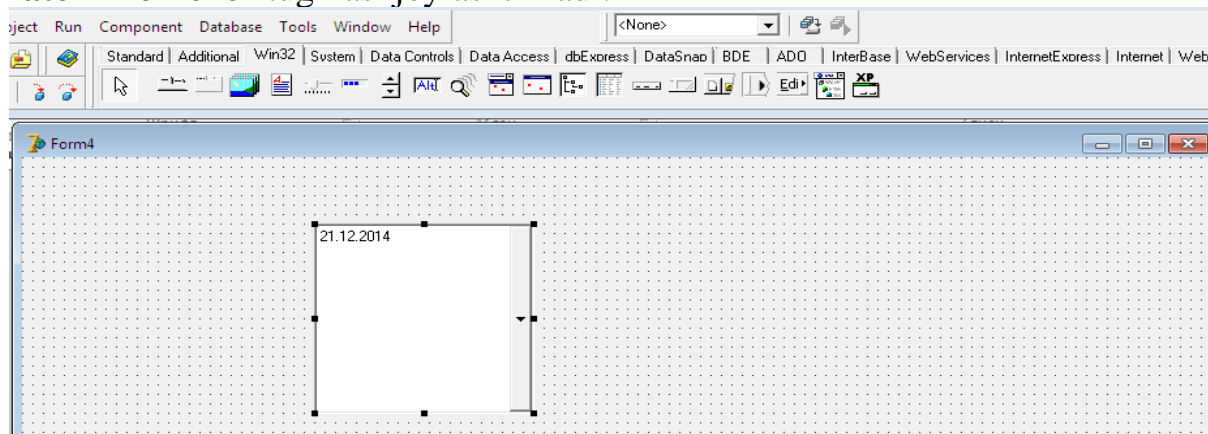
```

Masalaning to'rtinchi bosqichi

Biz bu bosqichda kalendaridan foydalanish imkoniyatini beruvchi quyidagi tugmalardan foydalanishni ko'rib chiqamiz: **TDateTimePicker**; **TMonthCalendar**.

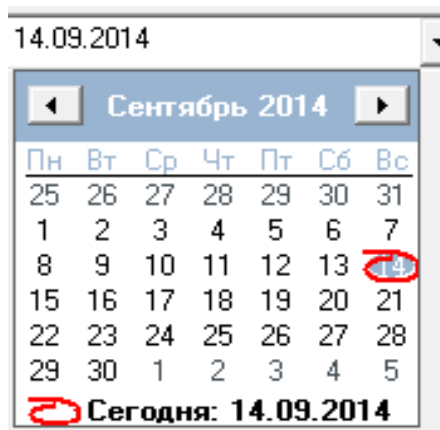
TDateTimePicker Komponentasidan Foydalanish

Forma4 oynasi hosil qilinadi va **Win32** komponentalar palitrasidan **TDateTimePicker** tugmasi joylashtiriladi.



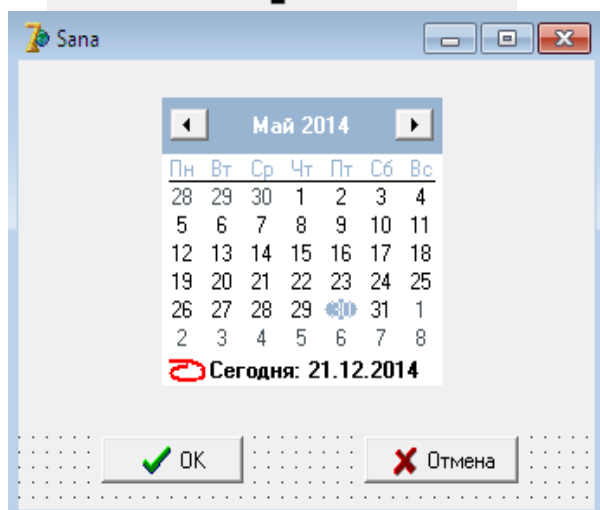
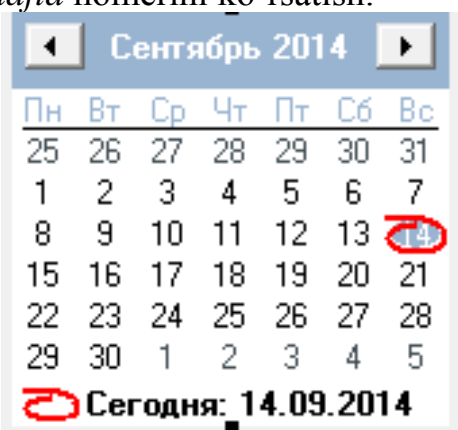
5.19-rasm.

TDateTimePicker tugmani joylashtirib, ishga tushirilganda quyidagi ko'rinishga keladi:



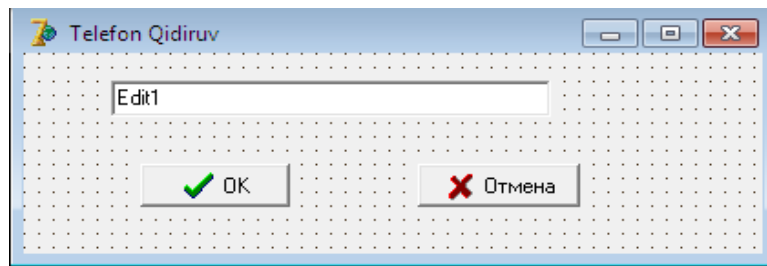
5.20-rasm.

Win32 komponentalar palitrasidan **MonthCalendar1** tugmasi joylashtiriladi. Ushbu tugmani joylashtirgandan so‘ng, forma oynasida doimiy kalendar ko‘rsatishga imkon beradi. Uning quyidagi asosiy xossalari mavjud: *FirstDayOfWeek* - birinchi sifatida ko‘rsatilgan hafta kuni; *Date* - tanlangan sana; *MaxDate* - maksimal sana; *MinDate* - minimal sana; *MultiSelect* - oy sonlari diapazonini tanlash imkoniyati; *ShowToday* - joriy sanani ko‘rsatish; *ShowTodayCircle* - joriy sana doirasini ko‘rsatish; *WeekNumbers* - hafta nomerini ko‘rsatish.



5.21-rasm.

Form1 oynasidagi “Qidirish” menyusi bandlarini ishga tushirish uchun 5.22-rasmda ko‘rsatilganidek yana ikkita forma oynasi hosil qilinadi.



5.22-rasm.

Unit1 modulining **Uses** qismiga **Unit4** va **Unit5** so'zlari yoziladi. "Qidirish" menyusi bandining prosteduralariga quyidagi o'zgartirishlar kiritiladi:

1. **Sana** bandiga quyidagi dastur kodi:

```

procedure TForm1.N5Click(Sender: TObject);
begin
  {Sanasi bo'yicha filterlash}
  if Form4.ShowModal=mrOK then
    begin
      with DataModule2.ADOTable1 do
        begin
          Filter := 'Sana=' +
            DateToStr(Form4.MonthCalendar1.Date);
          Filtered := True;
        end;
      end;
    end;
end;

```

2. **TELEFON** bo'yicha filterlash bandiga quyidagi dastur kodi:

```

procedure TForm1.N6Click(Sender: TObject);
begin
  if Form5.ShowModal=mrOK then
    begin
      with DataModule2.ADOTable1 do
        begin
          Filter := 'Telefoni = '+Form5.Edit1.Text;
          Filtered := True;
        end;
      end;
    end;
end;

```

3. Filtni bekor qilish bandiga quyidagi dastur kodi:

```

procedure TForm1.N10Click(Sender: TObject);
{Filtni olib tashlash}
begin
  DataModule2.ADOTable1.Filter := '';
  DataModule2.ADOTable1.Filtered := False;
end;

```

Yaratilgan loyiha oynsi saqlanib, natijani **F9** tugmasi orqali tekshirib ko'riladi.

O'zaro bog'langan jadvallar bilan ishlash

Bu bo'limda o'zaro bog'langan jadvallarga ega bo'lgan ma'lumotlar bazasi bilan ishlash amallari ko'rib chiqiladi. Ushbu tajriba ishida ma'lumotlar bazasi bilan bog'lanish **ADO** komponenti orqali amalga oshiriladi. Buning uchun yuqorida ko'rsatilgan amallar ketma-ketligini bajariladi.

Yuqoridagi masalani “Kundalik” ma’lumotlar bazasining “db1.mdb” faylini “Guruh baza2” papkasiga nusxasini ko‘chirib, **MS-Access** dasturi ishga tushushiriladi. “Guruh baza2” papkasidagi “db1.mdb” nomli fayl ochiladi va ma’lumotlar bazasiga yangi “Fan” nomli jadval tuziladi. U quyidagi ko‘rinishlarga ega bo‘ladi: *Kod – счетчик (xisoblagich); Familiya- числовой (sonli); Informatika- числовой (sonli); Tarix- числовой (sonli); Rus tili- числовой (sonli); Chet tili- числовой (sonli).*

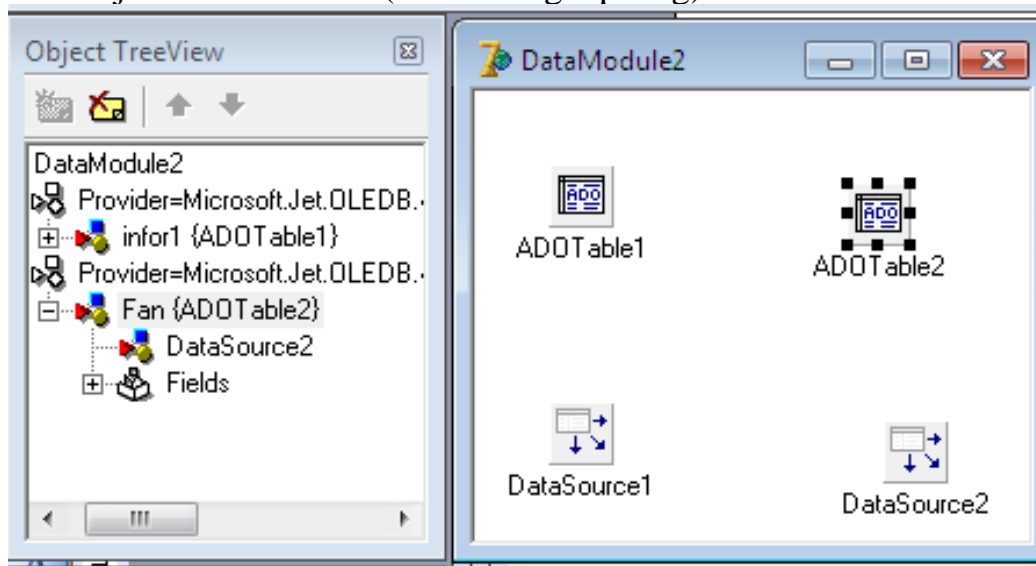
Yuqoridagi ketma-ketlik bajarilgandan so‘ng, jadval quyidagi ko‘rinishda to‘ldiriladi (6.23-rasnga qarang):

	Код	Familyasi	Infor	CHET TILI	O'Z TARIX	RUS TILI
+	1	TOSHEVA	76	77	78	71
+	2	MAHMUDOV	71	73	72	76
+	3	NAGIMOVA	71	71	78	74
▶+	4	USMONOVA	91	86	87	92
+	5	QOSIMOV	72	77	77	76

5.23 - rasm.

Hosil bo‘lgan jadvaldan so‘ng, “Infor1” jadvaliga o‘zgartirish kiritiladi, ya’ni “Kod” maydonini (ustunini) (Формат- числовой, число десятичных знаков-0). Bu maydonga 1÷4 gacha bo‘lgan sonlar kiritiladi. Demak, biz ikkita o‘zaro, ya’ni “Fan” kodi bo‘yicha bog‘langan jadvallarga ega bo‘ldik.

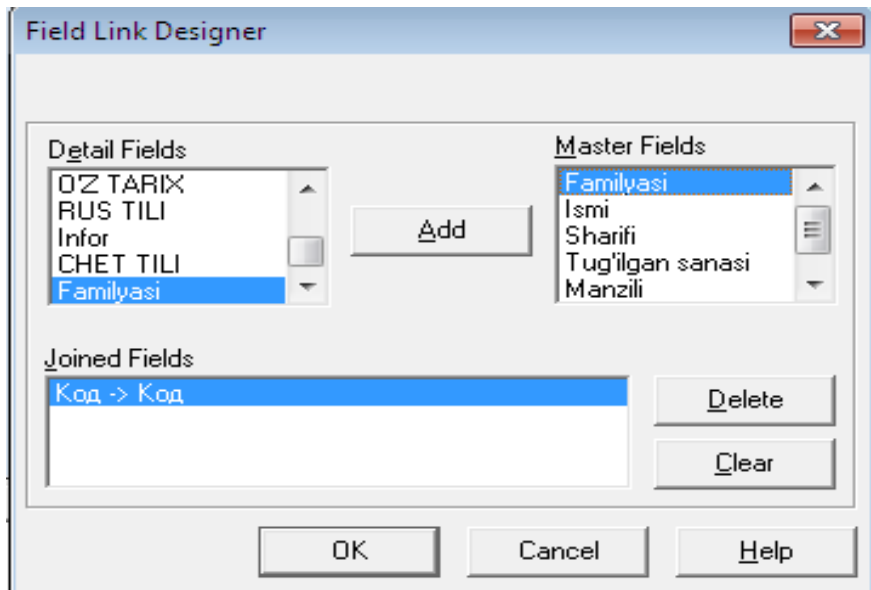
“Guruh baza” papkasidagi fayllardan “Guruh baza2” papkasiga nusxa ko‘chiriladi va **Delphi 7** dasturi ishga tushuriladi hamda oldin saqlanga “Lab4” loyiha ochiladi va uni “Lab5” nomi bilan saqlanadi. Loyihani ishga tushuriladi va uning ishlashi tekshirib ko‘riladi. Yangi nom bilan saqlangan loyihaning **DataModule2** formasiga o‘tiladi va unga **ADOTable2**, **DataSource2** tugmalar joylashtiriladi. **ADOTable2** tugmasi uchun bog‘lanish qatorini **Connection string** bo‘limiga “db1.mdb” ma’lumotlar bazasi bilan bog‘lanish sozlanadi va **TableName** qatoriga “Fan” jadvali tanlanadi (5.24-rasmga qarang).



5.24 - rasm.

Oldin hosil qilingan ikkita “Fan” va “Infor1” jadvallarni bir-biri bilan bog‘lanish hosil qilinadi. Buning uchun **MasterSource** xususiyatiga **DataSource1** so‘zi

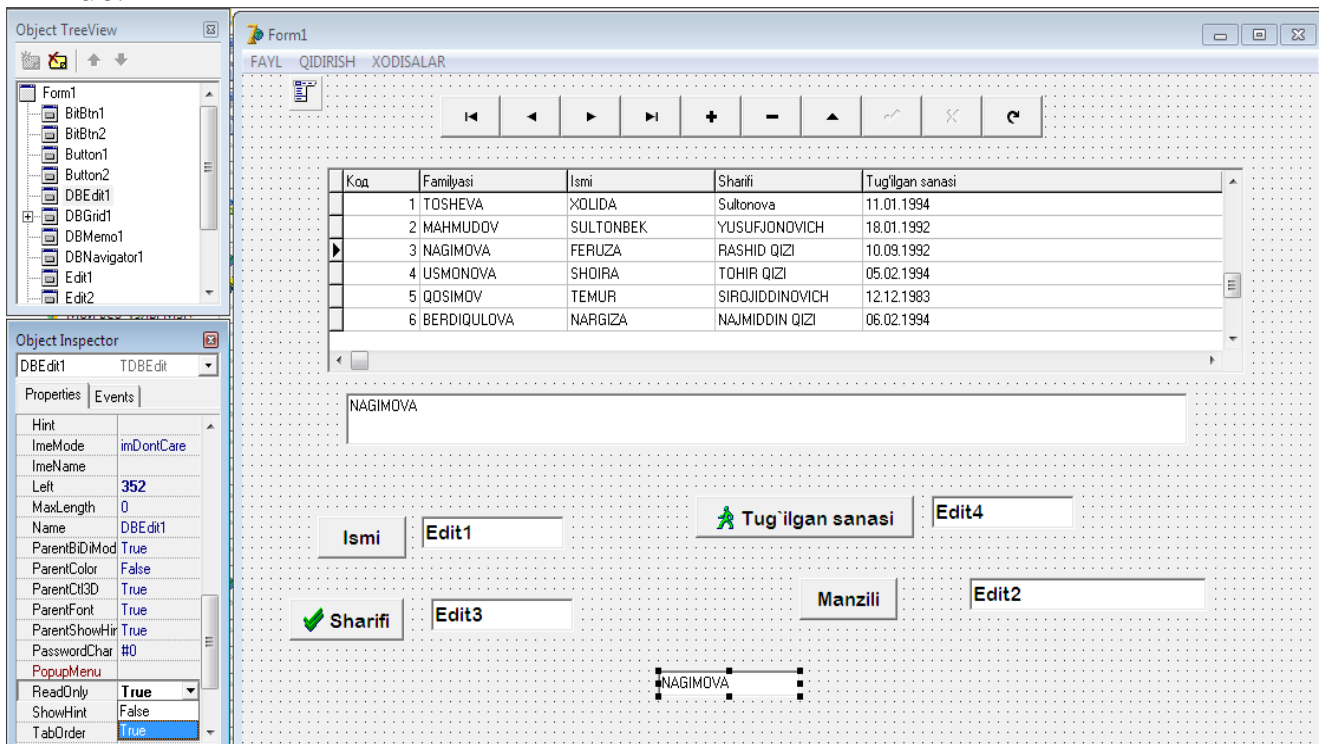
tanlanadi va **MasterFields** jadvalini 5.25-rasmda ko‘rsatilganidek bog‘lanish hosil qilinadi.



5.25 – rasm.

DataSource2 bo‘limiga **ADOTable2** ma‘lumotlari o‘rnatiladi (**ADOTable2** jadvalini faollashtiramiz).

Form1 formasiga **DBEdit** tugmasi joylashtiriladi va ma‘lumotlar manbaiga **DataModule2.DataSource2** ni o‘rnatiladi. **DataField** maydoniga (qatoriga) “Familiya” maydoni o‘rnatiladi. Ushbu qatorni taxrirlash cheklanadi, ya’ni **ReadOnly = True**.



5.26-rasm.

Loyha ishga tushiriladi. Hodisalarni ko‘rish oynasida kerakli qatorda ma‘lumotlar paydo bo‘lishi lozim. 5.27- rasmda ko‘rsatilgan namunadagidek yangi **Form6** formasi yaratiladi. Bu forma ma‘lumotnomani taxrirlash uchun xizmat qiladi. Unga **TDBGrid** va **TDBNavigator** tugmalari joylashtiriladi.

Familyasi	Infor	CHET TILI	O'Z TARIX	RUS TILI	PSI-YA
TOSHEVA	76	77	78	71	72
▶ MAHMUDOV	71	73	72	76	77

5.27- rasm.

“Fan” jadvalini taxrirlash uchun formani faollashtirish va yopish prosteduralariga 5.28-, 5.29-rasmlarda ko‘rsatilgan o‘zgartirishlar kiritiladi.

Buning uchun **Form4** loyiha oynasiga o‘tib, **Object Inspektor** oynasidan **Events** buyrug‘i tanlanadi. **OnClose** bandiga sichqoncha tugmasini ikki marta tez-tez bosiladi va **FormClose** oynasi hosil bo‘ladi. Bu oynaga quyidagi dastur kiritiladi (5.28-rasmga qarang):

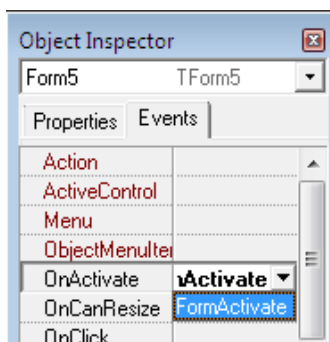
```

procedure TForm4.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  DataModule2.ADOTable2.MasterFields:='Kod';
end;

```

5.28-rasm.

Hosil bo‘lgan ketma-ketlikdan so‘ng, **Form5** oynasiga o‘tiladi va **Object Inspektor** ning **Events** buyrug‘i tanlanadi. **OnActivate** bandiga sichqoncha tugmasini ikki marta tez-tez bosiladi, natijada, **FormActivate** oynasi hosil bo‘ladi va bu oynaga quyidagi dastur kiritiladi (5.29-rasmga qarang):



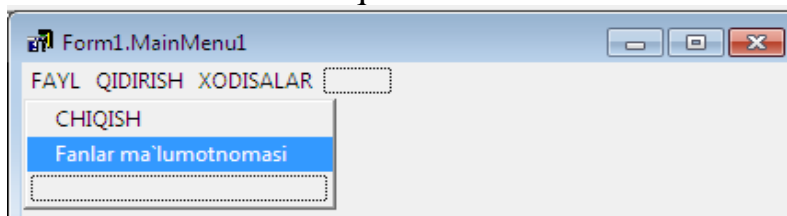
```

procedure TForm5.FormActivate(Sender: TObject);
begin
  DataModule2.ADOTable2.MasterFields:='';
  DataModule2.ADOTable2.Active:=true;
  DataModule2.ADOTable2.First;
end;

```

5.29-rasm.

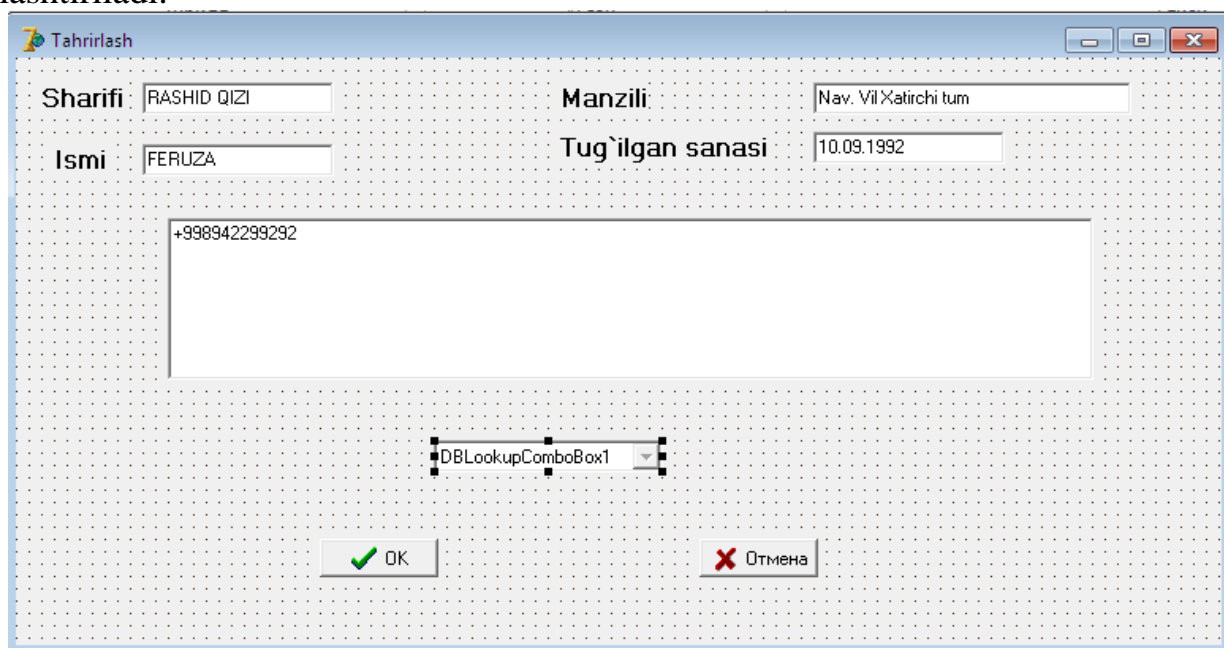
Birinci formadagi “Fayl” menyusiga “Fanlar ma’lumotnomasi” bandiga qo‘shiladi va unga **Form6** formasini chaqirish amali kiritiladi.



5.30-rasm.

Loyiha saqlanib, uni ishlashi ko‘riladi.

“Kiritish”, “Taxrirlash” oynasi (**Form3**) taxrirlanadi, ya’ni ma’lumotnomadan kerakli familiyani tanlash imkoniyati yaratiladi. Bu imkoniyatni beruvchi maydon **LookUp** deb ataladi. **Form3** oynasiga **TDBLookupComboBox** tugmasi joylashtiriladi.



5.31-rasm.

TDBLookupCombobox tugmasi uchun quyidagi xususiyatlar qo‘yiladi:
DataSource - *DataModule2.DataSource1*;
DataField – *Kod*;

ListSource DataModule2.DataSource2;

ListField – Familiya;

KeyField – Kod.

Natijada, **Form3** formasidagi ro'yxatdan familiyani tanlash natijasida, "Kontragent" kodini o'zgartirish imkoniyati mavjud bo'ladi. Buni quyidagicha tasvirlash mumkin: 1-jadvaldagi kod o'zgarsa, 2-jadvaldagi boshqa maydon ma'lumoti ko'rsatiladi. Barcha forma oynalariga rang beriladi. Buning uchun form oynasining dasturlash maydoniga quyidagi dastur kodi kiritiladi:

procedure TForm3.FormCreate(Sender: TObject);

begin

form3.color:=cllime; end;

Natijada, oynaning umumiy ko'rinishi quyidagicha bo'ladi (6.32-rasmga qarang):

5.32-rasm.

Hosil bo'lgan loyiha onalaridan **Form1** oynasiga "Ismi", "Sharifi", "manzili" va "tug'ilgan yili" bo'yicha qidiruvchi dastur kodi kiritiladi. Buning uchun **Form1** oynasida 4 ta **Edit** va 4 ta **BitBtn** tugmalari hosil qilinadi hamda **BitBtn** tugmalarining dasturlash maydonlariga quyidagi dastur kodi kiritiladi:

procedure TForm1.Button1Click(Sender: TObject);

begin if not DataModule2.ADOtable1.Locate('Ismi',edit1.Text,[]) then

showmessage('Bunday yozuv yo'q'); end;

procedure TForm1.Button2Click(Sender: TObject);

begin if not DataModule2.ADOtable1.Locate('Manzili',edit2.Text,[]) then

showmessage('Bunday yozuv yo'q'); end;

procedure TForm1.BitBtn1Click(Sender: TObject);

begin if not DataModule2.ADOtable1.Locate('Sharifi',edit3.Text,[]) then

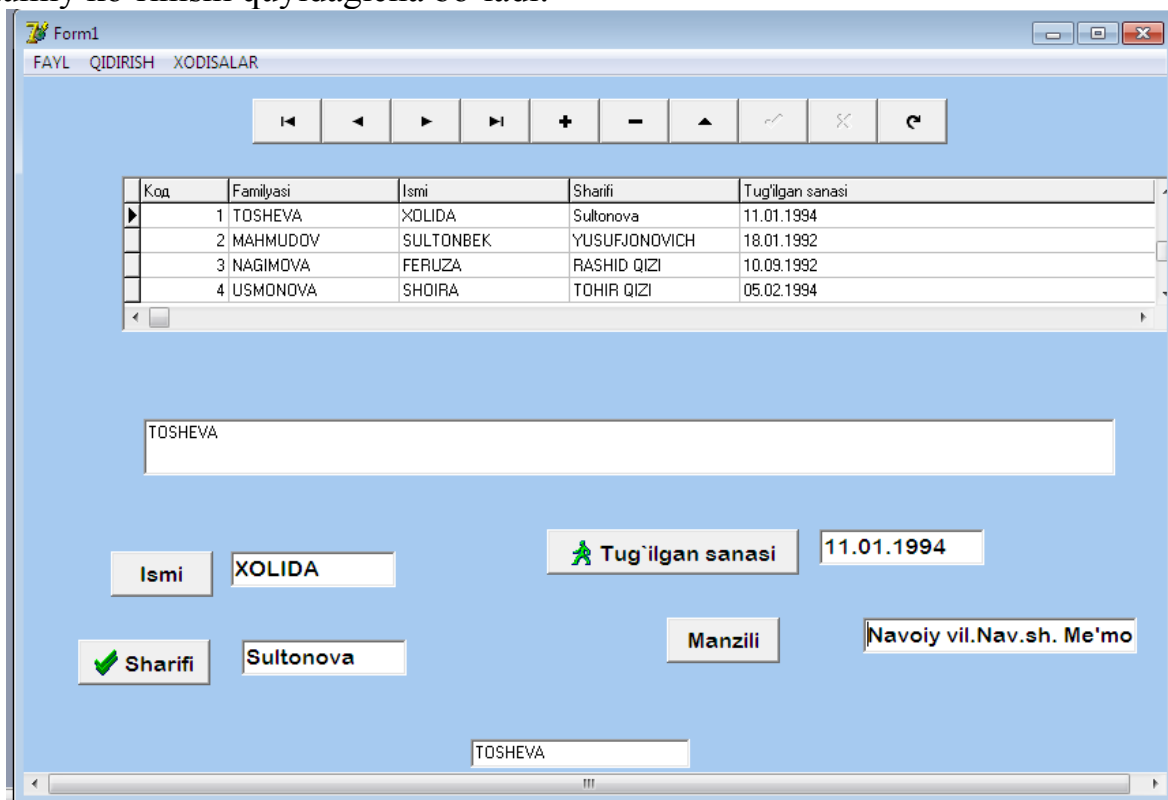
showmessage('Bunday yozuv yo'q'); end;

procedure TForm1.BitBtn2Click(Sender: TObject);

begin if not DataModule2.ADOtable1.Locate('Tug'ilgan sanasi',edit4.Text,[]) then

showmessage('Bunday yozuv yo'q'); end;

Loyiha oynasini saqlanadi va u ishga tushuriladi (5.33-rasm). Natijada, oynaning umumiy ko‘rinishi quyidagicha bo‘ladi:



5.33-rasm.

5.2. Delphi muhitida ADO komponentalar panelidan ADO Connection va ADOQuery komponentalaridan foydalanish

MS-Access dasturi ishga tushuriladi: Пуск→Все программы→Microsoft Access 2003. Dastur yuklangandan so‘ng, quyidagi amallar bajariladi: Созданий база данных-Новая база данных; ma’lumotlar bazasi faylini “SQL” papkasida “db1.mdb” nom bilan saqlanadi; Создание таблицы в режиме конструктора rejimi tanlanadi.

Yuqoridagi ketma-ketlikdan so‘ng, jadval quyidagicha to‘ldiriladi:

Maydon (polya) nomi	Ma’lumotlar tipi	Formati
Familyasi	Текстовый	30 simvolov
Ismi	Текстовый	30 simvolov
Tug’ilgan sanasi	Текстовый	30 simvolov
Shahar	Числовой	
Telefon	Текстовый	30 simvolov
E-mail	Текстовый	30 simvolov
Sana	Дата/время	
Uy telefoni	Логический	

Key1	Familiya	Ism	Telefon	E-mail	Shahar	Sana	Uyali telefon
1	TOSHEVA	XOLIDA	+998944398292	Xolida@inbox.uz	32	11.01.1994	<input checked="" type="checkbox"/>
2	MAHMUDOV	SULTONBEK	+998933724422	Sulton@inbox.uz	29	18.01.1992	<input type="checkbox"/>
3	NAGIMOVA	FERUZA	+998942299292	Feruz9292@inbox.uz	28	10.09.1992	<input type="checkbox"/>
4	USMONOVA	SHOIRA	+998933165258	Shoira5258@inbox.uz	32	05.02.1994	<input type="checkbox"/>

5.34-rasm.

Ushbu jadvalni “Telefon” nomi bilan saqlab, navbatdagi jadvalni hosil qilib, (6.35-rasm) da ko‘rsatilgan tartibda to‘ldirib, “Shahar” nomi bilan saqlanadi.

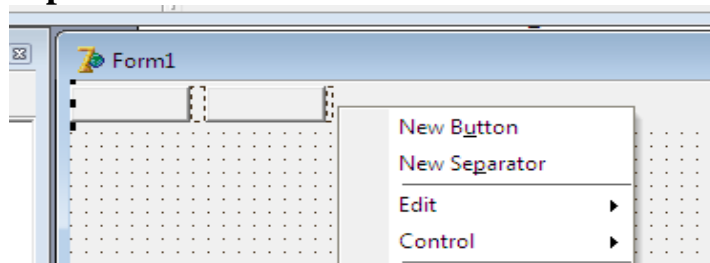
Key1	Shaharlararo qongiroq
28	Navoiy
29	Toshkent
30	Buxoro
31	Namangan
32	Samarqand
33	Xorazm
34	Farg‘ona
35	Andijon
*	(Счетчик)

5.35-rasm.

Yuqoridagi bajarilgan ketma-ketlikdan so‘ng, **Delphi** dasturi ishga tushurilib, yangi loyiha oynasi yuklanadi hamda “SQL” papkasiga “Telefon” nomi bilan saqlanadi va uskunalar panelidan **ToolBar** va **ControlBar** tugmalari forma oynasiga joylashtiriladi.

ControlBar tugmasining qulayligi shundaki, unga uskunalar paneli joylashtirilsa, panel ichida siljitish mumkin bo‘ladi. Ikkala tugma uchun asosiy xossalar guruxi **EdgeBorders** bo‘lib, bu guruhdagi **ebTop** xossasi qiymatiga *false* o‘rnatilsa tugma chetidagi hoshiya yo‘qoladi.

ToolBar tugmasi ustida sichqoncha o‘ng tugmasi tanlansa, tugma yaratadi. Har bir yaratilgan tugma, yangi komponenta hisoblanadi. **New Separator** buyrug‘i tugmalarni ajratuvchi oraliq tugmalar tahrirlash menyusi chiqadi. Bu menyuda **New Button** buyrug‘i, yangi tugmani o‘chirish uchun ajratib klaviaturadagi **Delete** tugmasini **Object Inspektor** xossalari va hodisalarini ko‘rish mumkin (5.36-rasm).



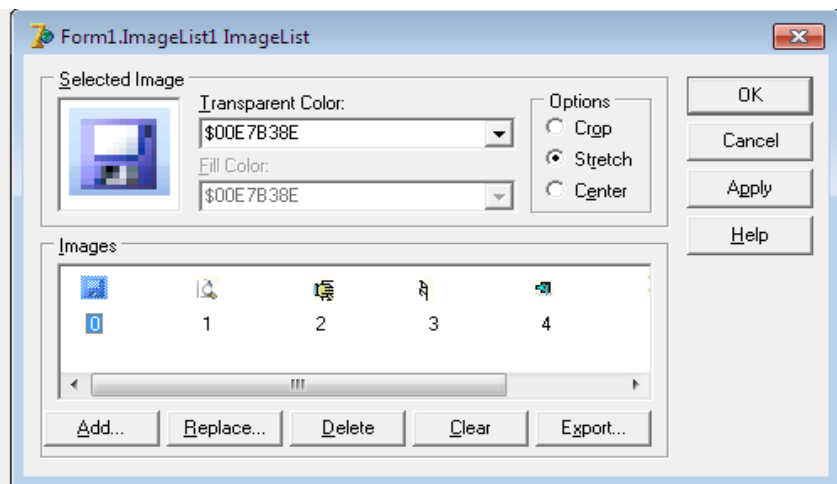
5.36-rasm.

Win32 komponentalar palitrasidan **ImageList** tugmasi tanlanadi. Bu tugma rasmlarni qulay shaklda saqlash uchun ishlatiladi. Buning uchun forma loyiha oynasiga **ImageList** tugmasi o‘rnatiladi va rasmlar joylashtiriladi.

Asosiy xossalari: **Height**-rasmlar kengligi; **Width**-rasmlar balandligi.

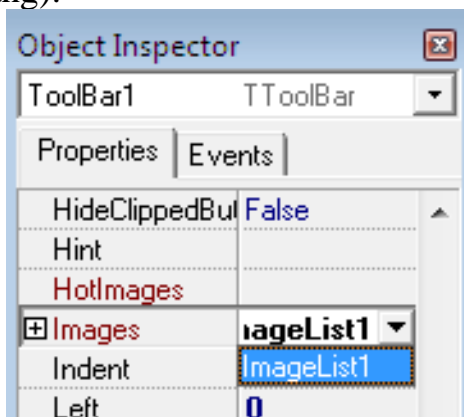
Agar uskunalar panelining **ShowCaptions** xossasiga **true** qiymat berilsa tugmalarda rasmdan tashqari matnlarni ham aks ettirish mumkin bo‘ladi. Agar **Flat** xossasiga **true** qiymati berilsa tugmalar chiroyliroq ko‘rinishga ega bo‘ladi.

Tugmalarga rasm o‘rnatish uchun formaga **ImageList** tugmasini o‘rnatiladi (6.37-rasm).



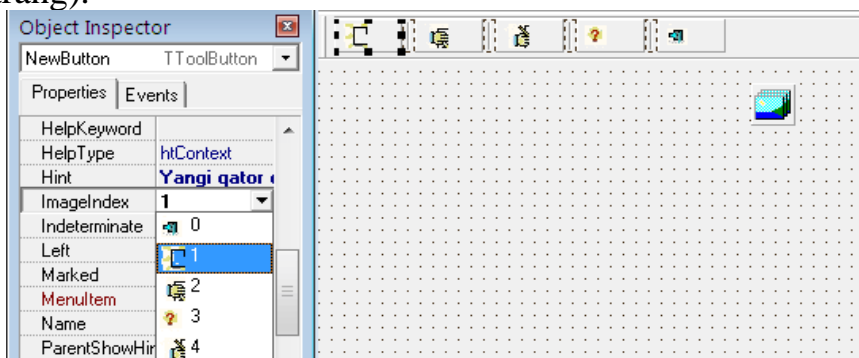
6.37-rasm.

ToolBar1 tugmasining **Object Inspektordan Images** bandidan **ImageList1** tanlanadi (6.39rasmga qarang).



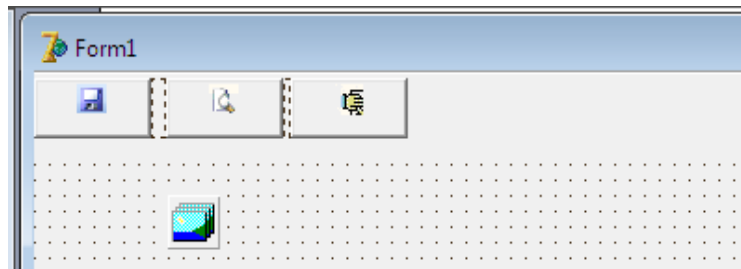
5.38-rasm.

Tugmadagi rasmini o‘zgartirish uchun **ImageIndex** bandidan foydalaniladi (5.39-rasmga qarang).



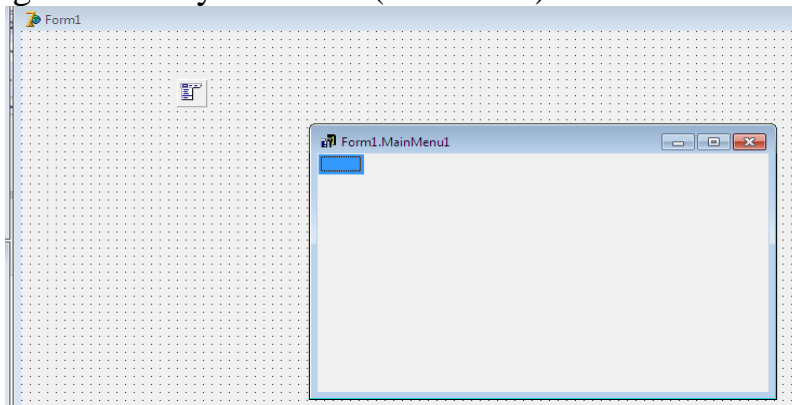
5.39-rasm.

Rasmdagi matni chap tomonga joylash uchun **List** xossasiga **true** qiymat berish lozim (5.40-rasm).



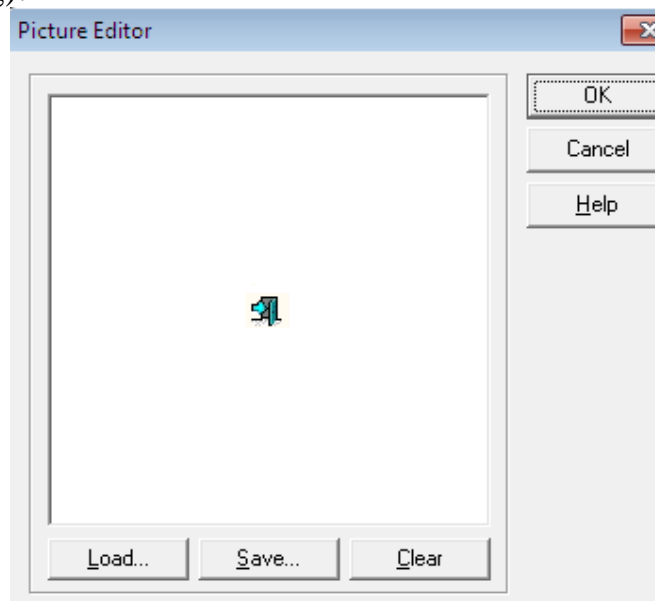
5.41-rasm.

Yuqoridagi ketma-ketlikdan so‘ng, form oynalari bilan o‘zaro bog‘lanish hosil qilinadi. Buning uchun esa **Standart** komponentalar palitrasining **MainMenu** tugmasidan foydalaniladi (5.42-rasm).



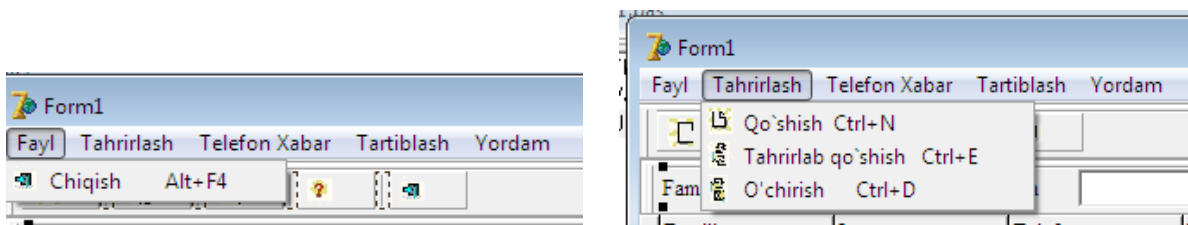
5.42-rasm.

MainMenu tugmasi belgilanib, **Object Inspektor** ning **Caption** bandiga “Fayl” so‘zini kiritib, menyu hosil qilinadi va uning ustuniga “chiqish” band kiritiladi. Fayl menyusining satr yo‘nalishi bo‘yicha ham yorliqlar yaratiladi. Masalan: “Tahrirlash”, “Telefon xabar”, “Tartiblash”, “Yordam” kabi menyularni kiritib chiqiladi. Fayl menyusining “chiqish” bandi tanlanib, **Object Inspektor** ning **Bitmap** bandiga sichqoncha tugmasi tez-tez ikki marta bosiladi. Natijada, quyidagi oyna hosil bo‘ladi (5.43-rasmga qarang):

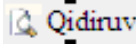


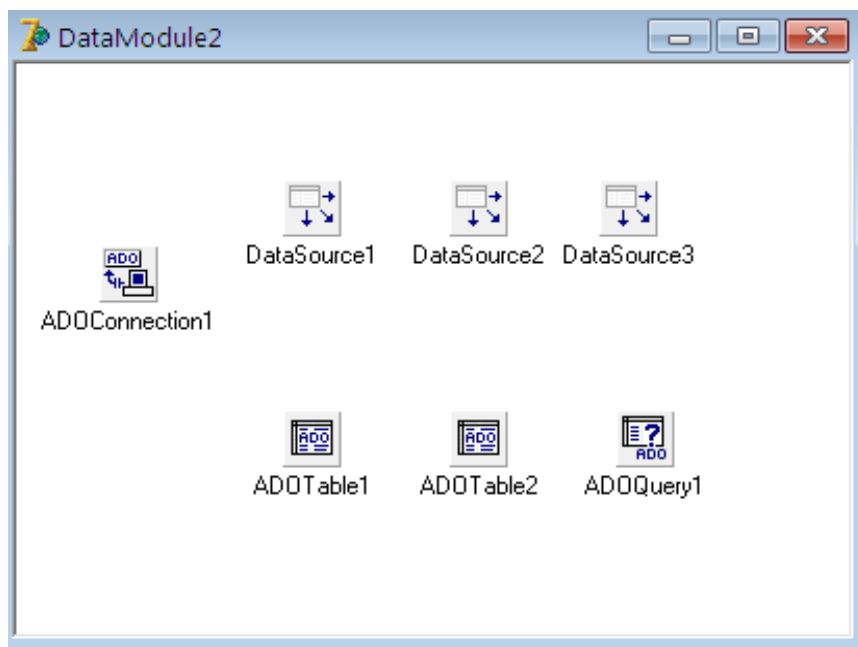
5.43-rasm.

Load buyrug‘i orqali lozim bo‘lgan papkadan rasm tanlanadi va **OK** tugmasi bosiladi. Ushbu ketma-ketlik orqali boshqa bandlarga ham sur‘atlar joylashtirib chiqiladi (5.44-rasm).



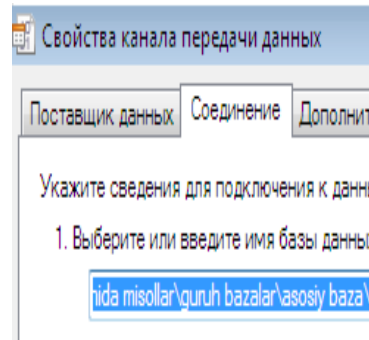
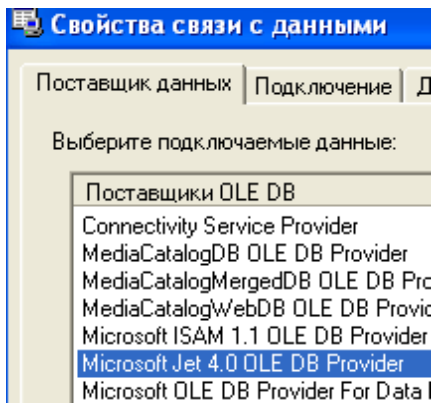
5.44-rasm.

Uskunalar panelidan **ToolBar2** tugmasi tanlanadi va form oynasiga joylashtiriladi. **ToolBar2** tugmasining ichiga 2 ta **Label**, 2 ta **Edit** va bitta **Findbutton** tugmalari joylashtiriladi. **Label1** tugmasining **Caption** bandiga “Familiya”, **Label2** ga esa “Telefon” soʻzlari kiritiladi. **Findbutton** ning **Caption** bandiga “Qidiruv” soʻzini va **ImageIndex** bandiga rasm raqamini tanlab kiritiladi  hamda **File**→**New**→**Data Module2** ketma-ketligi tanlanadi. Ushbu oynaning **Components** qismiga **ADO** komponentlar palitrasidan **ADOConnection1** tugmasi joylashtiriladi. Soʻngra **DataAccess** komponentlar palitrasidan 3 ta **DataSource**, **ADO** komponentlar palitrasidan esa 2 ta **ADOTable** va bitta **ADOQuery** tugmalari joylashtiriladi (5.45-rasm).



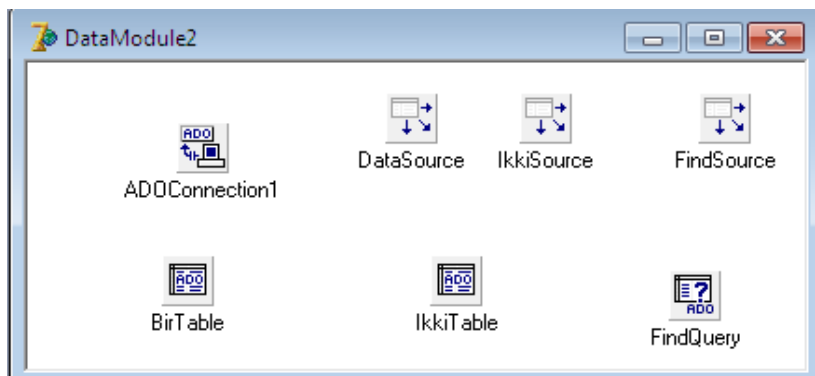
5.45-rasm.

ADO Connection komponentining **Connection String** xususiyatiga bogʻlash qatorining (Connection String) dialogi chaqiriladi va **Build...** tugmasi bosilib, soʻngra **Microsoft.Jet.OLEDB.4.0** (MS-Access maʼlumotlar bazasiga kirishga ruxsat beruvchi drayver) maʼlumotlar tipi tanlanadi. **Connection** qismining birinchi qatoriga “SQL” papkasidan “db1.mdb” (oʻzimiz tuzgan maʼlumotlar bazasi fayli nomi) fayli nomi yoziladi. **Blank password** (пустой пароль) qismiga “✓” belgisi qoʻyiladi va bogʻlanish tekshirib koʻrilib, dialogli oyna yopiladi (5.46-rasm).




5.46-rasm.

ADOTable1 tugmasining **Caption** bandiga **BirTable** nomi kiritiladi va quyidagi xususiyatlar oʻrnatiladi: **Active-True**, **TableName-Telefon1**. **ADOTable2** tugmasining **Caption** bandiga **IkkiTable** nomi kiritiladi va quyidagi xususiyatlar oʻrnatiladi: **Connection-ADODConnection1**, **Active-True**, **TableName-Telefon2**. **ADOQuery** tugmasining **Caption** bandiga **FindQuery** nomi kiritiladi (5.47-rasmga qarang).

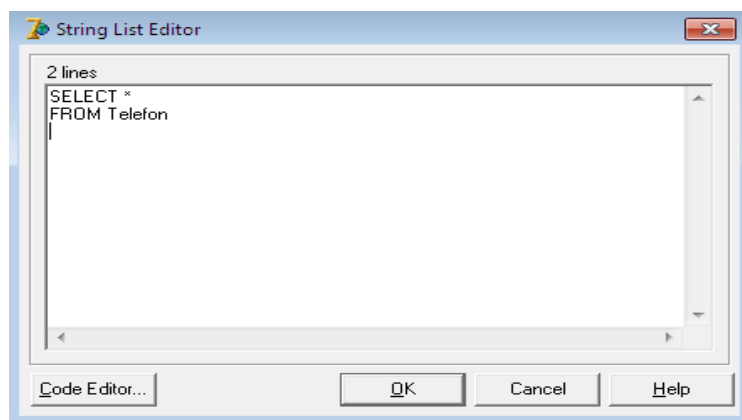
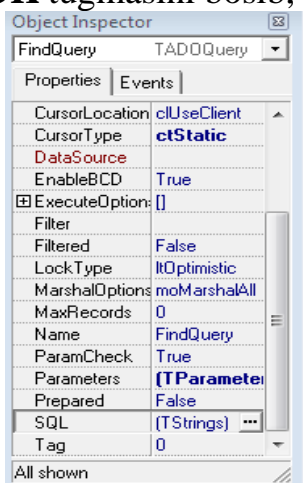


5.47-rasm.

Connection-ADODConnection, **ADOQuery**  tugmasi uchun quyidagi xususiyatlar oʻrnatiladi: **SQL** bandiga sichqoncha tugmasi 2 marta bosiladi va oyna hosil boʻlgandan soʻng quyidagi soʻzlar kiritiladi (5.48-rasm):

*SELECT **
FROM Telefon

va **OK** tugmasini bosib, **Active-True** ga oʻtkaziladi.



5.48-rasm.

Ma'lumotlarni aks ettirish uchun ma'lumotlar moduli oynasiga (DataModule2) **Data Access** komponentlar palitrasidan **Data Source** (ma'lumotlar manbai) tugmasi joylashtiriladi. Bu tugma uchun **DataSe** xususiyatiga **ADOTable1** qiymati qo'yiladi va **Data Source** tugmasi, ma'lumotlar bazasi bilan, undagi elementlarni aks ettirish orasida uzatuvchilik funksiyasini bajaradi. **Unit1** modulini taxrirlash oynasiga o'tilib, uning **Uses** qismiga **Unit2** so'zi yoziladi. Bu ma'lumotlar modulidan **Form1** loyiha oynasidan foydalanish imkoniyatini beradi (5.49-rasm). Ammo **DataModule2.DataSource1** tanlaganidan so'ng, **Unit1** modulini taxrirlash oynasiga o'tkazib, uning **Uses** qismiga kiritilgan **Unit2** so'zi o'chiriladi hamda **Unit2** modulini taxrirlash oynasiga o'tib, uning **Uses** qismiga **Unit** so'zi kiritiladi. Agar ushbu amallar bajarilmasa yaratilgan dastur ishlamaydi.



5.49-rasm.

Form1 loyiha oynasiga **Data Controls** komponentlar panelidan **DBGrid** tugmasi joylashtiriladi **Data Source** xususiyatiga **DataModule2.DataSource**, **AlignCustom**, **BorderStyle- bsNone** qiymatlari o'rnatiladi. Natijada, loyiha oynasi quyidagi ko'rinishga ega bo'ladi (5.50-rasm):

Familiya	Ism	Telefon	E-mail	Sana	Shahar	Uyali telefo
TOSHEVA	XOLIDA	+998944398292		11 Январь 1994 г.	Samarqand	Ha
MAHMUDOV	SULTONBEK	+998933724422		18 Январь 1992 г.	Toshkent	Yo'q
NAGIMOVA	FERUZA	+998942299292		10 Сентябрь 1992 г.	Navoiy	Yo'q
USMONOVA	SHOIRA	+998933165258	Shoira5258@inbox.uz	5 Февраль 1994 г.	Samarqand	Yo'q

5.50-rasm.

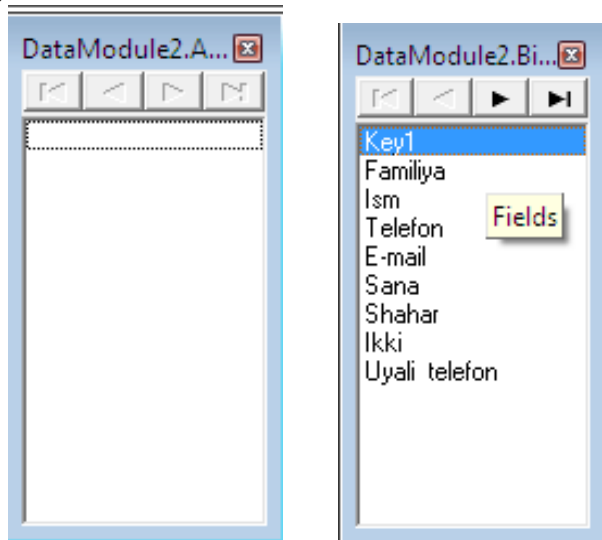


BirTable

Yuqoridagi ketma-ketlikdan so'ng, **DataModule2** oynasiga qaytib, **BirTable** tugmasining ustiga sichqoncha tugmasi ikki marta bosiladi. Natijada, bo'sh oyna hosil bo'ladi. Bu oynaga ma'lumotlar kiritiladi (Masalan, key1, familiya, ism, telefon, shahar va h.k.). **Object Inspektor** oynasining **FieldName** bandiga "Key1" so'zi kiritiladi va **ReadOnly-True**, **Visible-True** qiymatlari o'rnatiladi. So'ngra, "familiya", "ismi", "telefon" va "e-mail" so'zlari **FieldName** bandiga kiritiladi (uning xossalari o'zgartirilmaydi). "Shahar" so'zini **FieldName** bandiga **Visible-True** qiymati o'rnatiladi.

"Uyali telefon" so'zi kiritilganidan so'ng, **DisplayValues** bandiga "Ha; Yo'q" so'zlari kiritiladi. "Sana" so'zi **FieldName** bandiga kiritiladi va **DisplayFormat** bandiga "dddddd" 6 ta "d" harfi kiritiladi. **EditMask** bandiga sichqoncha tugmasini ikki marta bosish orqali bo'sh oyna hosil bo'ladi va bu oynaning **Input Mask** bandiga quyidagi raqam va belgilar kiritiladi: 99/99/9999;1;_

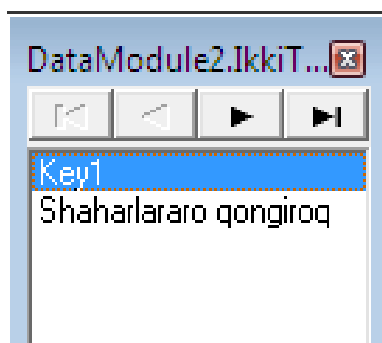
“Ikki” soʻzini **FieldName** bandiga, **DisplayLabel** bandiga esa “shahar” soʻzi kiritiladi. **FieldKind-fkLookup**, **KeyFields-shahar**, **LookupDataSet-IkkiTable**, **Lookup KeyFields-Key1**, **LookupResultField-shaharlararo qongiroq** qiymatlari oʻrnatiladi (5.51-rasm).



5.51-rasm.



- IkkiTable tugmasining ustiga sichqoncha tugmasini ikki marta bosish orqali yangi oyna hosil boʻladi. **Object Inspektor** oynasining **FieldName** bandiga “Key1” soʻzi kiritiladi va **ReadOnly- True**, **Visible-True** qiymatlari oʻrnatiladi. “Shaharlararo qongiroq” soʻzi **FieldName** bandiga kiritiladi (unga oʻzgartirishlar kiritilmaydi) (5.52-rasm).



5.52-rasm.



-FindQuery va **BirTable** tugmalari yangi **Form** loyiha oynasiga joylashtiriladi va **Object Inspektor**ning quyidagi bandlari oʻzgartiriladi:

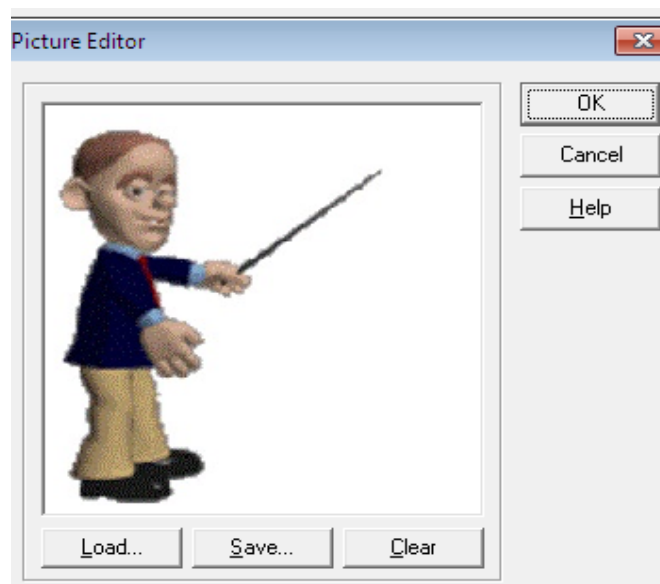
Name- EditForm;

Horz ScrollBar- TcontrolScrollBar;

Vert ScrollBar- TcontrolScrollBar;

OldCreateOrder-False;

EditForm oynasining **Standart** komponentalar palitrasidan **Panel** tugmasi tanlanadi va ushbu oynaga joylashtiriladi. **Object Inspektor** oynasidan **ParentFont** ni **False** ga oʻzgartiriladi. **Panel** tugmasining ustiga **Image** tugmasi joylashtiriladi. **Object Inspektor** oynasining **Picture** bandiga sichqoncha tugmasini ikki marta bosiladi va **Load** buyrugʻi orqali kerakli boʻlgan papkadan rasm tanlanib, **OK** tugmasi bosiladi (5.53-rasm).



5.53-rasm.

Panel tugmasining ustiga beshta **DBEdit**, oltita **Label**, bitta **DBLookupComboBox1**, bitta **DBCheckBox1** va ikkita **BitBtn** tugmalari oʻrnatiladi.

Label tugmalarining **Caption** bandiga “familiya”, ismi”, “shahar”, “tugʻilgan sana”, “telefon” va “e-mail” soʻzlari kiritiladi. **DBEdit** tugmasining **Object Inspektor** oynasining **DataSource-DataModule2.DataSource**, **DataField-“Familiya”** bandlarini har biri ushbu tartibda oʻzgartirib chiqiladi. **DBCheckBox1** tugmasining **Object Inspektor** oynasining quyidagi bandlari oʻzgartiriladi:

DataSource-DataModule2.DataSource,
DataField –Uyali Telefon.

DBLookupComboBox1 tugmasining quyidagi bandlari oʻzgartiriladi:

DataSource-DataModule2.DataSource,
DataField –Shahar,
KeyField- Key1,
ListField- Shaharlararo qongiroq,
ListSource- DataModule2.IkkiSource .

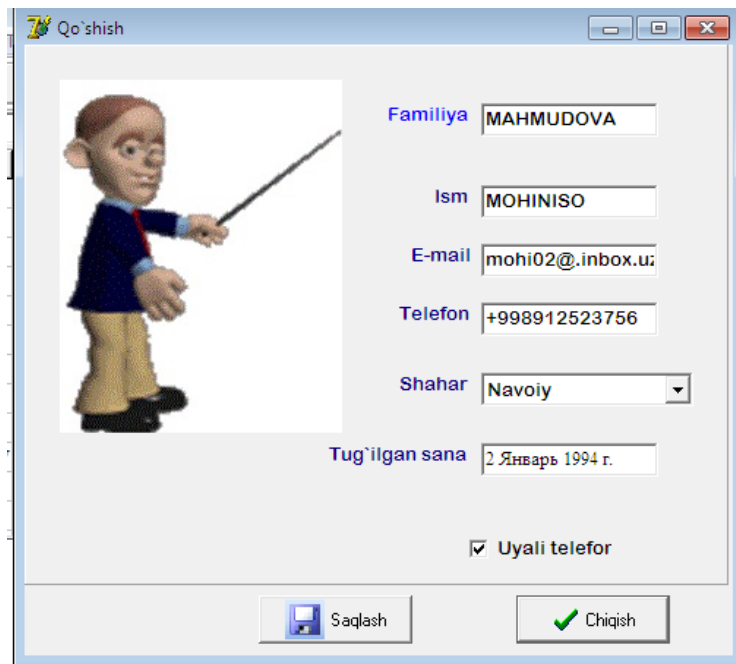
BitBtn tugmalaridan birining **Caption** bandiga “Saqlash”, ikkinchisiga esa “Chiqish” soʻzlarini kiritiladi. “Saqlash” tugmasini **Object Inspektor** oynasining **Glyph** bandiga sichqoncha tugmasini ikki marta bosiladi va **Load** buyrugʻi orqali kerakli boʻlgan papkadan rasm tanlanib, **OK** tugmasi bosiladi (6.55-rasm). “Saqlash” tugmasining dasturlash maydoniga quyidagi dastur kodi kiritiladi:

```
Procedure TEditForm.SaveBtnClick(Sender: TObject);
  begin
  if DataModule2.BirTable.Modified then
  DataModule2.BirTable.Post;
  end;
```

“Chiqish” tugmachasining dasturlash maydoniga quyidagi dastur kodi kiritiladi:

```
procedure TEditForm.CloseBtnClick(Sender: TObject);
```

```
begin
close; end;
```



5.54-rasm.

Yuqoridagi ketma-ketlikdan so‘ng, yangi **Form** loyiha oynasi hosil qilinadi. Bu **Form** oynasiga “IkkinchiForm” deb nom beriladi. Ushbu loyiha oynasiga bitta **DBGrid**, to‘rtta **BitBtn** tugmalari joylashtiriladi.

DBGrid tugmasining **Object Inspektor** oynasidan quyidagi o‘zgartirishlar kiritiladi:

Align- allLeft;

DataSource -DataModule2.IkkiSource;

BorderStyle- bsNone.

BitBtn tugmalarining **Caption** bandiga mos ravishda “Saqlash”, “Chiqish”, “O‘chirish” va “Qo‘shish” so‘zlari kiritiladi. Har bir tugmani **Object Inspektor** oynasining **Glyph** bandiga sichqoncha tugmasi ikki marta bosiladi va **Load** buyrug‘i orqali kerakli bo‘lgan papkadan rasmlar tanlanib, **OK** tugmasi bosiladi. “IkkinchiForm” oynasining umumiy ko‘rinishi 5.55-rasmda keltirilgan).

“Saqlash” tugmasining dasturlash maydoniga quyidagi dastur kodi kiritiladi:

Procedure TEditForm.SaveBtnClick(Sender: TObject);

begin

if DataModule2.BirTable.Modified then

DataModule2.BirTable.Post; end;

“Chiqish” tugmasining dasturlash maydoniga quyidagi dastur kodi kiritiladi:

procedure TEditForm.CloseBtnClick(Sender: TObject);

begin close;end;

“Qo‘shish” tugmasining dasturlash maydoniga quyidagi dastur kodi kiritiladi:

procedure TIkkiBirForm.BitBtn3Click(Sender: TObject);

begin

DataModule2.IkkiTable.Insert;

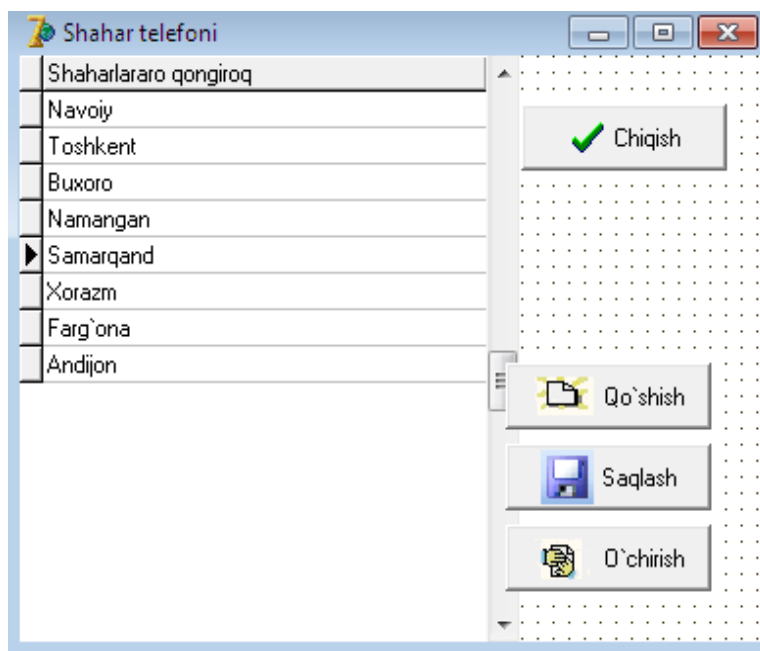
DBGrid1.SetFocus; end;

“O‘chirish” tugmasining dasturlash maydoniga quyidagi dastur kodi kiritiladi:

procedure TIkkiBirForm.DeleteBtnClick(Sender: TObject);

begin DataModule2.IkkiTable.Delete;

end;



5.55-rasm.

Yangi **Form** loyiha oynasi yaratiladi va unga “FindResultForm” deb yoki istalgan biror nom yoziladi (Maslan, “Qidiruv natijasi”). **Form** oynasiga bitta **DBGrid** va bitta **BitBtn** tugmalari joylashtiriladi.

DBGrid tugmasining **Object Inspektor** dan quyidagi o‘zgartirishlar kiritiladi:

Align- alClient;

DataSource - DataModule2.FindSource;

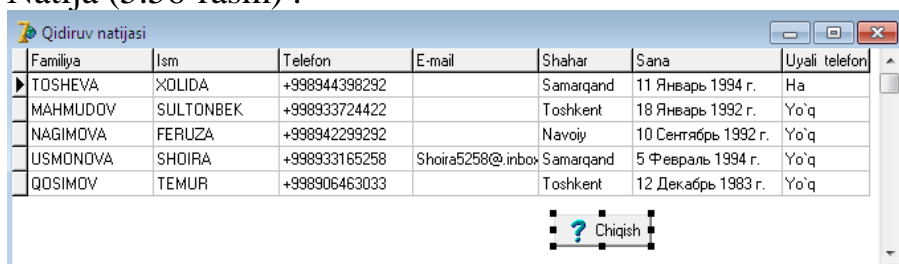
BorderStyle- bsNone.

BitBtn tugmasining **Caption** bandiga “Chiqish” so‘zi kiritiladi. **BitBtn** tugmasining dasturlash maydoniga quyidagi dastur kodi kiritiladi:

procedure TFindResultForm.BitBtn1Click(Sender: TObject);

begin close; end;

Natija (5.56-rasm) :



5.56-rasm.

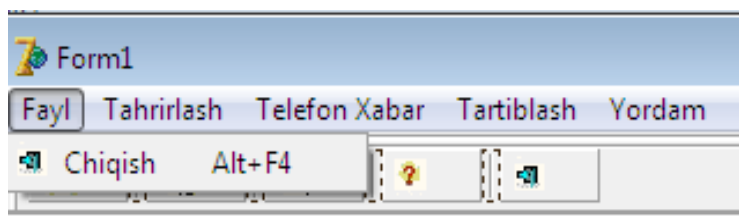
Yuqoridagi ketma-ketlikdan so‘ng, yangi **Form** oynasi yaratiladi. Unga “AboutForm” deb nom beriladi. **Form** oynasiga bitta **Panel**, bitta **Image**, ikkita **Label**, bitta **BitBtn** tugmalari joylashtiriladi. **Image** tugmasi yordamida biror sur‘at joylashtiriladi. **Label1** ga dastur tuzuvchi nomini va **Label2** ga sanasi yoziladi. **BitBtn** tugmasining **Kind** bandidan **OK** piktogrammasi tanlanadi. Natijada, oynaning umumiy ko‘rinishi quyidagicha bo‘ladi (5.57-rasm):



5.57-rasm.

Yuqoridagi ketma-ketlikdan so‘ng, asosiy oyna ya’ni **Form1** loyiha oynasiga o‘tib, “Chiqish Alt+F4” tugmasini dasturlash maydoniga quyidagi dastur kodi kiritiladi:

```
procedure TEditForm.CloseBtnClick(Sender: TObject);
begin
close;end;
```



5.58-rasm.

“Qo‘shish Ctrl+C” tugmasini dasturlash maydoniga quyidagi dastur kodi kiritiladi:

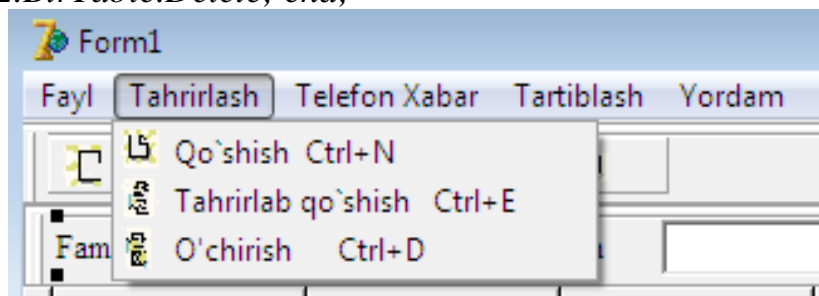
```
procedure TForm1.Qoshish1Click(Sender: TObject);
begin DataModule2.BirTable.Insert;
EditForm.ShowModal; end;
```

“Tahrirlab qo‘shish Ctrl+E” tugmasini dasturlash maydoniga quyidagi dastur kodi kiritiladi:

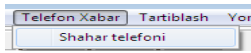
```
procedure TForm1.EditButtonClick(Sender: TObject);
begin EditForm.ShowModal; end;
```

“O‘chirish Ctrl+D” tugmasini dasturlash maydoniga quyidagi dastur kodi kiritiladi:

```
procedure TForm1.OchirishCtrlD1Click(Sender: TObject);
begin
DataModule2.BirTable.Delete; end;
```

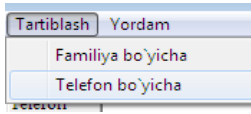


5.59-rasm.



“Telefon xabar” menyusining “Shahar telefoni” bo‘limiga quyidagi dastur kodi kiritiladi:

```
procedure TForm1.Shahartelefoni1Click(Sender: TObject);  
begin IkkiBirForm.ShowModal; end;
```

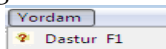


“Tartiblash” menyusining “Familiya bo‘yicha” bo‘limiga quyidagi dastur kodi kiritiladi:

```
procedure TForm1.Familiya1Click(Sender: TObject);  
begin DataModule2.BirTable.IndexFieldNames:= 'Familiya'; end;
```

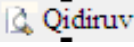
“Telefon bo‘yicha” bo‘limining dasturlash maydoniga quyidagi dastur kodi kiritiladi:

```
procedure TForm1.elfonboyicha1Click(Sender: TObject);  
begin DataModule2.BirTable.IndexFieldNames:= 'Telefon'; end;
```

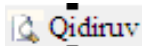


“Yordam” menyusining “Dastur F1” bo‘limiga quyidagi dastur kodi kiritiladi:

```
procedure TForm1.Dastur1Click(Sender: TObject);  
begin AboutForm.ShowModal; end;
```

Birinchi qidiruv  tugmasini dasturlash maydoniga esa quyidagi dastur kodi kiritiladi:

```
procedure TForm1.ToolButton1Click(Sender: TObject);  
begin  
DataModule2.FindQuery.Active:=false;  
DataModule2.FindQuery.SQL.Clear;  
DataModule2.FindQuery.SQL.Add('SELECT *');  
DataModule2.FindQuery.SQL.Add('FROM Telefon');  
DataModule2.FindQuery.SQL.Add('WHERE Familiya LIKE '"+Edit1.Text+"'");  
DataModule2.FindQuery.Active:=true;  
FindResultForm.ShowModal;end;
```



Ikkinchi qidiruv tugmasining dasturlash maydoniga quyidagi dastur kodi kiritiladi:

```
procedure TForm1.FindButtonClick(Sender: TObject);  
begin DataModule2.FindQuery.Active:=false;  
DataModule2.FindQuery.SQL.Clear;  
DataModule2.FindQuery.SQL.Add('SELECT *');  
DataModule2.FindQuery.SQL.Add('FROM Telefon');  
DataModule2.FindQuery.SQL.Add('WHERE Telefon LIKE  
“'+FindTelephoneEdit.Text+'”');  
DataModule2.FindQuery.Active:=true;  
FindResultForm.ShowModal; end;
```

Natija F9 tugmasi yordamida ko‘riladi (5.60- rasm):

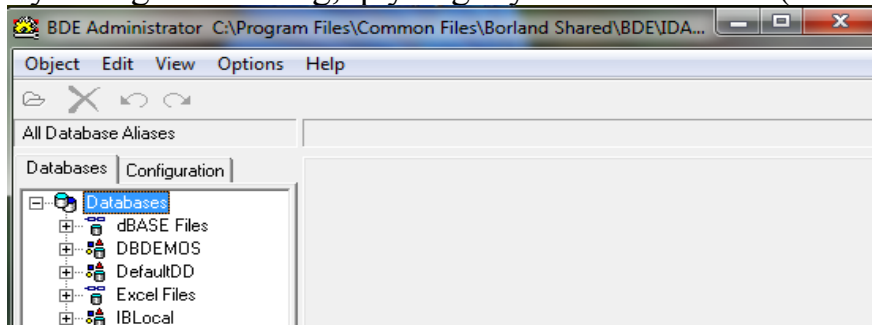
Familiya	Ism	Telefon	E-mail	Sana/vaqt	Shahar	Uyali telefon
TOSHEVA	XOLIDA	+998944398292	Xolida@ir	11 Январь 1994 г.	Samarqand	Ha
MAHMUDOV	SULTONBEK	+998933724422	Sulton@ir	18 Январь 1992 г.	Toshkent	Yo'q
NAGIMOVA	FERUZA	+998942299292	Feruz929	10 Сентябрь 1992 г.	Navoiy	Yo'q
USMONOVA	SHOIRA	+998933165258	Shoira525	5 Февраль 1994 г.	Samarqand	Yo'q
QOSIMOV	TEMUR	+998906463033		12 Декабрь 1983 г.	Toshkent	Yo'q
BERDIQULOVA	NARGIZA	+998942253658		6 Февраль 1994 г.	Navoiy	Ha
JO'LIBKOV	ASLONBEK	+998933154744		25 Декабрь 1992 г.	Namangan	Yo'q
RAHMONOV	ZAFAR	+998941892538		28 Апрель 1988 г.	Toshkent	Yo'q
ERGASHEVA	UMIDA	+998942544448	Umida444	17 Март 1987 г.	Samarqand	Yo'q
MUHAMMEDOV	SHAXBOZ	+998916465666		14 Март 1992 г.	Navoiy	Ha
RASULOVA	NIGORA	+998933160663		22 Сентябрь 1991 г.	Vuxoro	Yo'q
MAHMUDOVA	MOHINISO	+998912523756	mohi02@i	2 Январь 1994 г.	Navoiy	Ha
TO'XLIYEV	ASILBEK	+998934370931	Aska_93@	11 Февраль 1993 г.		Ha

5.60-rasm.

5.3. Delphi muhitining BDE Administrator va Database Desktop bo'limlari bilan ishlash

Delphida **BDE Administrator** bo'limidan foydalanib, ma'lumotlar bazasini yaratishimiz mumkin. **BDE Administrator** bo'limini ishga tushurish ketma-ketligi quyidagi ko'rinishda amalga oshiriladi: **Пуск-Программы-Borland Delphi-BDE Administrator**.

Dastur yuklangandan so'ng, quyidagi oyna hosil bo'ladi (5.61-rasm):



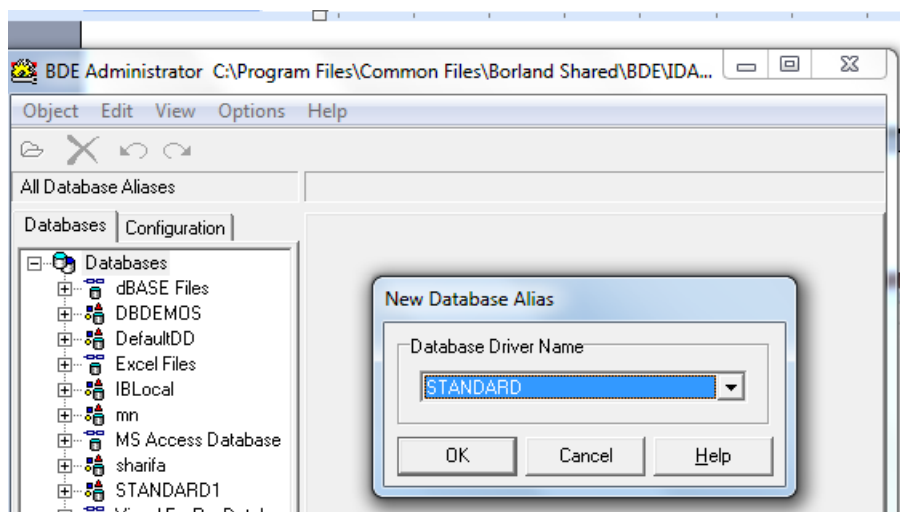
5.61-rasm.

Ushbu oynadan **Object** menyusidan **New** buyrug'i tanlanadi (5.62-rasm).



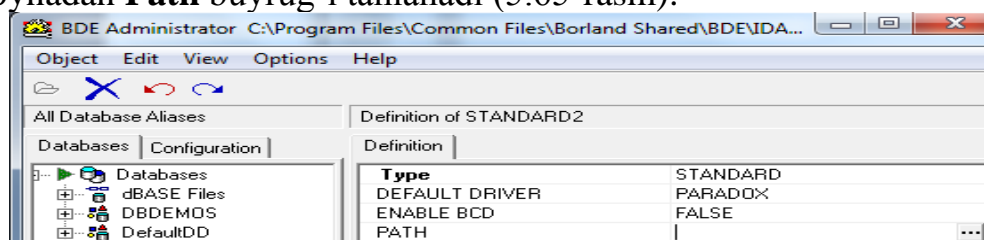
5.63-rasm.

Hosil bo'lgan oynadagi **New Database Alias** oynadan **Standard** bo'limi tanlanadi va **OK** tugmasi bosiladi (5.64-rasm):



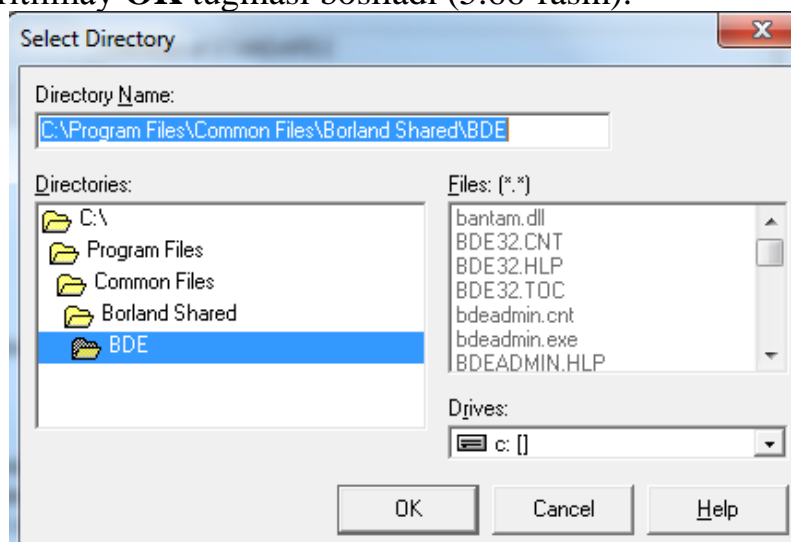
5.64-rasm.

OK tugmasi bosilgandan so'ng, chap oynadan **STANDART1** nomi o'rniga yangi nom ham qo'yishimiz mumkin, o'ng oynada **Definition** bo'limi paydo bo'ladi. Ushbu oynadan **Path** buyrug'i tanlanadi (5.65-rasm):



5.65-rasm.

Path buyrug'i tanlangandan so'ng, navbatdagi oyna ochiladi, bu oynaga o'zgartirish kiritilmay **OK** tugmasi bosiladi (5.66-rasm):



5.66-rasm.

Yuqorida bajarilgan ketma-ketlikdan so'ng, **Object>Apply** yoki **Ctrl+A** buyrug'i bajariladi. Natijada, **BDE Administrator**dan oynasi hosil bo'ladi.

Ma'lumotlar bazasi jadvalini tuzish

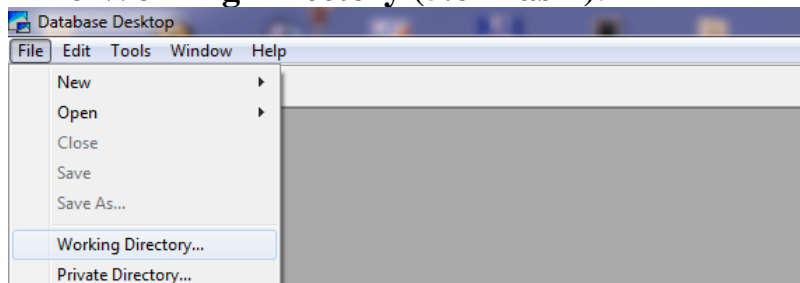
Ma'lumotlar bazasini yaratish uchun avvalombor uning modelini, ya'ni tuzilmasini ishlab chiqish zarur. Buning uchun MBda maydon tushunchasi kiritilgan bo'lib, uning tuzilmasi aniqlanadi. Maydonni aniqlash quyidagi elementlardan tashkil

topadi: *maydon nomi (Fields Name)*; *maydon turi (Type)*; *maydon o'lchami (Size)*; *maydon kaliti (Key)*.

Maydon nomi - 10 tagacha lotin harfi va sonlardan tashkil topgan bo'lishi mumkin. Masalan, **DBase MBBT** uchun maydon turlari quyidagilar bilan aniqlanadi: "A" simvolli (matnli) qiymatlarni bildirib, uning uzunligi 255 tagacha bo'lishi mumkin; "N" sonli qiymatlarni bildirib, uning diapazoni -10307 +10308; "D" sana (data) qiymatini bildirib, uning o'zgarish diapazoni 01.01.9999 dan 31.12.9999 gacha; "L" mantiqiy qiymatni bildirib, **True** va **False** qiymatlarni qabul qiladi; "M" memo (izohli) bo'lib, qatorli qiymatlarni qabul qiladi, har bir qator 255 tagacha simvol olishi mumkin. **Memo1** ma'lumotlari maxsus **Dbt** kengaytmali faylda saqlanadi.

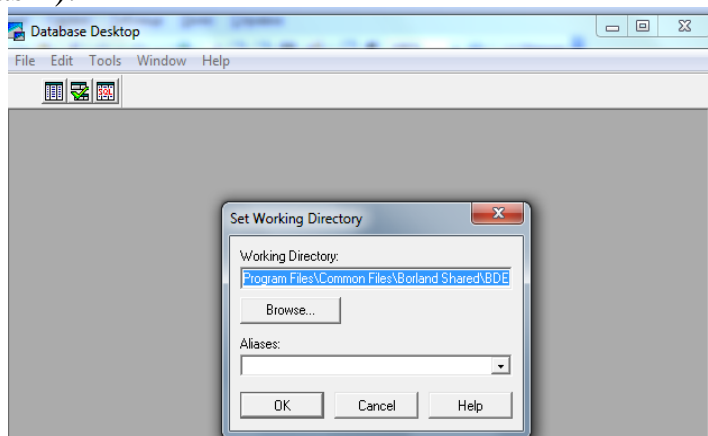
MBning jadvalini yaratish algortmi quyidagi ketma-ketlikda bajariladi:

DataBase DeskTop bo'limi ishga tushiriladi (**Пуск-Программы-Borland Delphi-Database Desktop**). **Database Desktop** oynasining bosh menyusidan quyidagi buyruq bajariladi: **File-Working Directory (5.67-rasm)**.



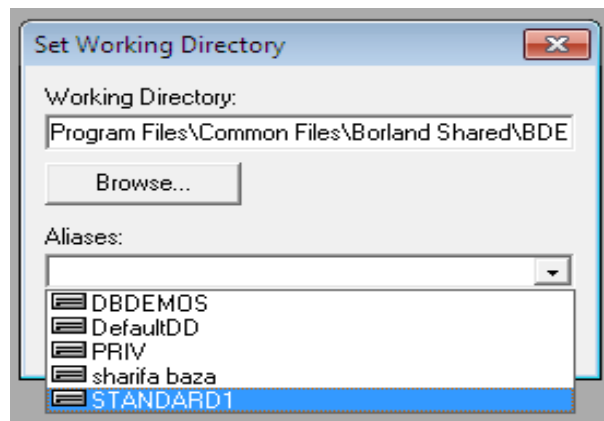
5.67-rasm.

Hosil bo'lgan **Set Working Directory** muloqot darchaining **Aliases** bo'limidan **BDE Administrator**ning chap oynasidagi **STANDART1** nomi o'rniga qo'yilgan nom yoki o'zgartirilmagan bo'lsa o'zi tanlanadi. Bu nom ma'lumotlar bazasiga **Delphi** komponenti ilovalari tamonidan mantiqiy murojaat qilishda ishlatiladi(6.68-rasm):



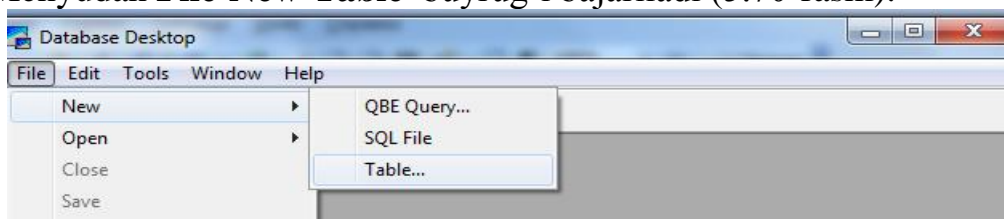
5.68-rasm.

Natijada, quyidagi oyna ochiladi, bu oynadan **STANDART1** buyrug'i tanlanadi (5.69-rasm):



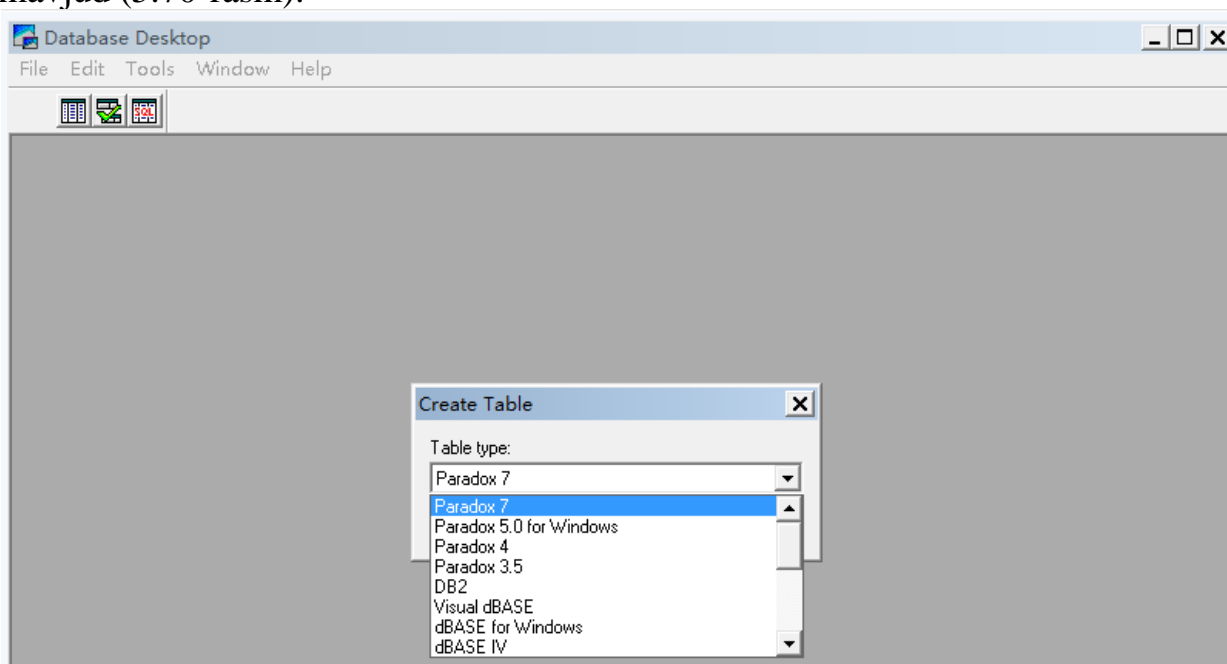
5.69-rasm.

Menyudan **File-New-Table** buyrug'i bajariladi (5.70-rasm):



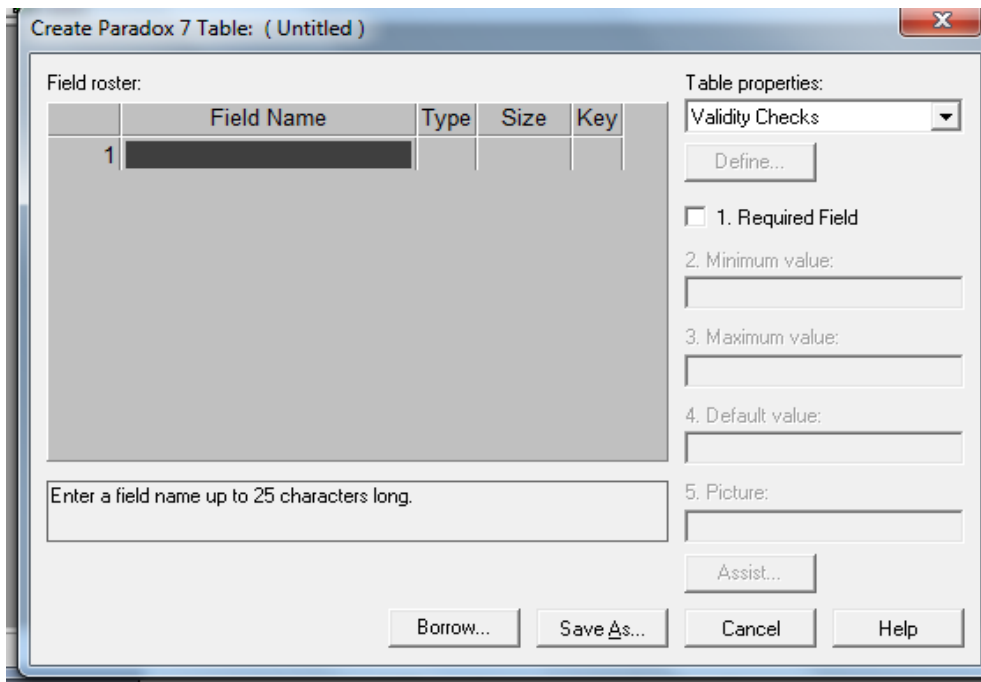
5.70-rasm.

Hosil bo'lgan **Create Table** muloqot darchasidan MB turi aniqlanadi (Masalan, **Paradox** yoki **Dbase Windows**). Bu darchada MBning bir qancha turlari mavjud (5.70-rasm):



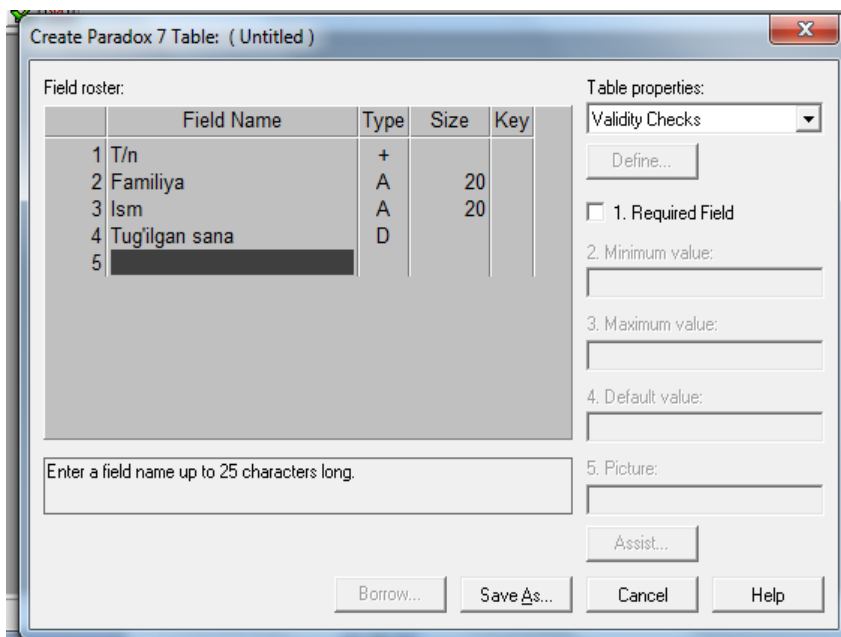
5.703-rasm.

Yuqorida aytilgan ketma-ketlikdan so'ng, quyidagi oyna ishga tushadi (5.71-rasm):



5.71-rasm.

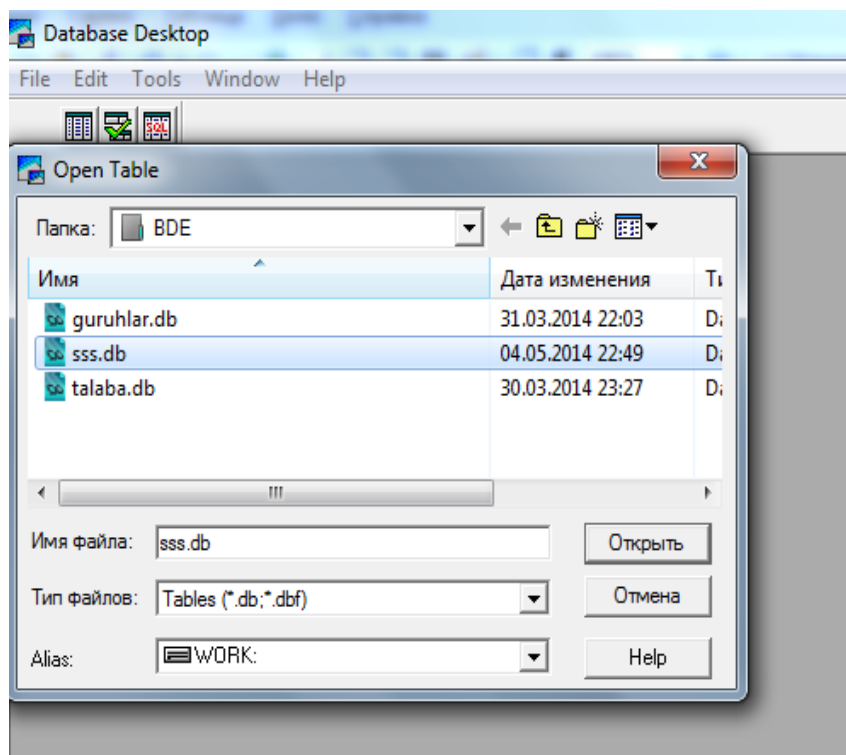
Hosil bo'lgan oyna quyidagi rasmda ko'rsatilinganidek to'ldiriladi(5.72-rasm):



5.72-rasm.

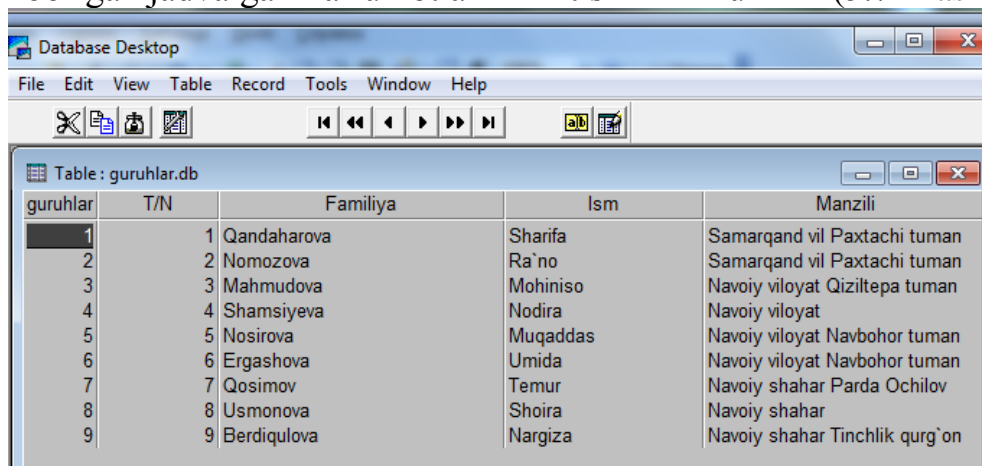
Save As buyrug'i berilib, jadval nomi kiritiladi va saqlanadi.

Agar MBga ma'lumotlar kiritish kerak bo'lsa menyudan **File-Open-Table** buyrug'i tanlanib, **Open Table** oynasidan **fayl** tanlanadi va ma'lumotlar kiritiladi (5.73-rasm):



5.73-rasm.

Hosil bo'lgan jadvalga ma'lumotlarni kiritishimiz mumkin (5.74-rasm):



5.74-rasm.

Ma'lumotlar kiritilgandan so'ng, dasturdan chiqiladi.

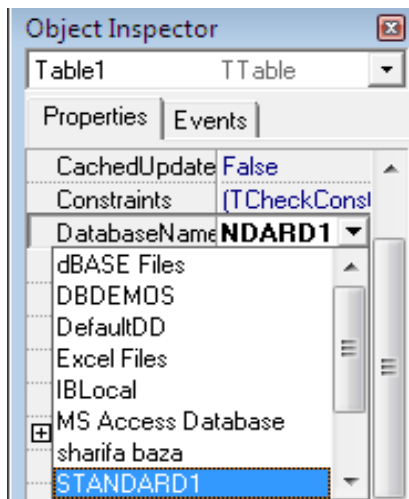
MB jadvali bilan ishlash uchun oddiy ilova yaratish algoritmi

MB jadvali bilan ishlash uchun oddiy ilova yaratish algoritmi quyidagi ketma-ketlikda bajariladi: **Delphi** dasturlash tili ishga tushirilib, **BDE** komponentalar palitrasidan **TTable** tugmasi **Forma** oynasiga joylashtiriladi (5.75-rasm):



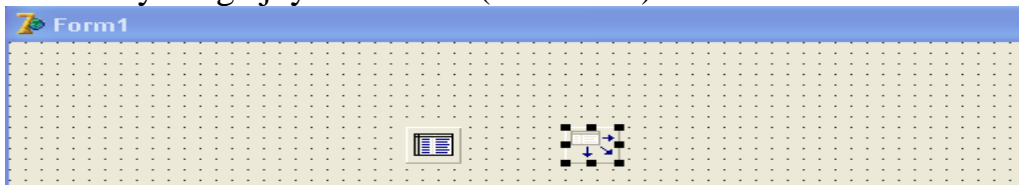
5.75-rasm.

Forma oynasidagi **TTable** tugmasi belgilanib, **Database Name** xossasida **Aliases** bo'limidan nomlangan nomi tanlanadi (5.76-rasm):



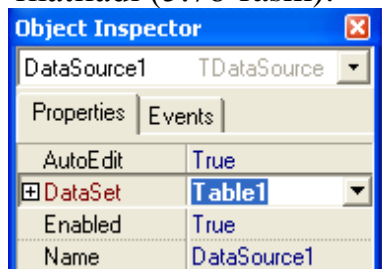
5.76-rasm.

TableName xossasidan **MB** jadvali nomi aniqlanadi va **Active** xossasiga **True** qiymati oʻrnatiladi hamda **Data Access** komponentalar palitrasidan **TDataSource** tugmasi **Forma** oynasiga joylashtiriladi (5.77-rasm):



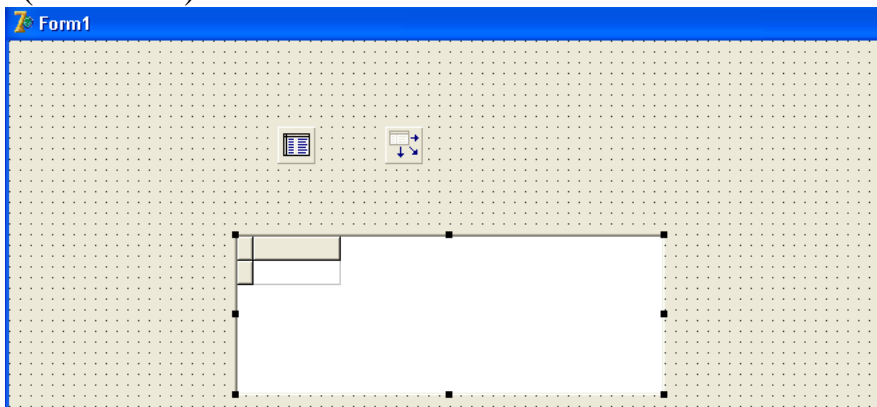
5.77-rasm.

DataSet xossasidan **Table1** oʻrnatiladi (5.78-rasm):



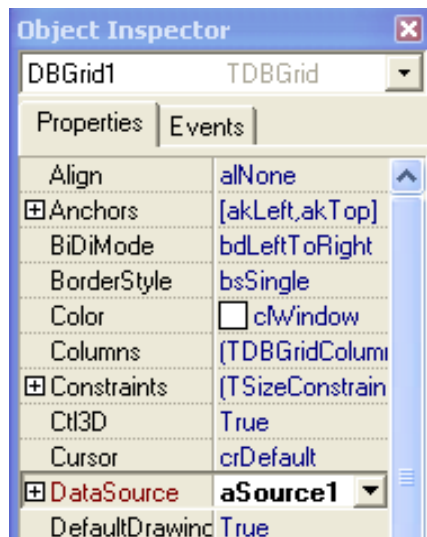
5.78-rasm.

Data Controls komponentalar palitrasidan **TDBGrid** tugmasi **Forma** oynasiga joylashtiriladi (5.79-rasm):



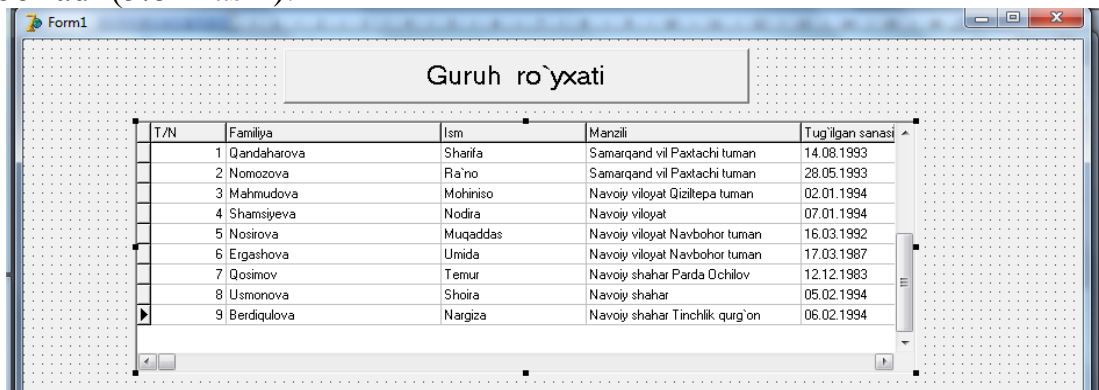
5.79-rasm.

DataSource xossasidan **DataSource1** oʻrnatiladi (5.80-rasm):



5.80-rasm.

Menyudun **File-Save Project As** buyrug‘i bajarilib. Natijada, quyidagi oyna hosil bo‘ladi (5.81-rasm):

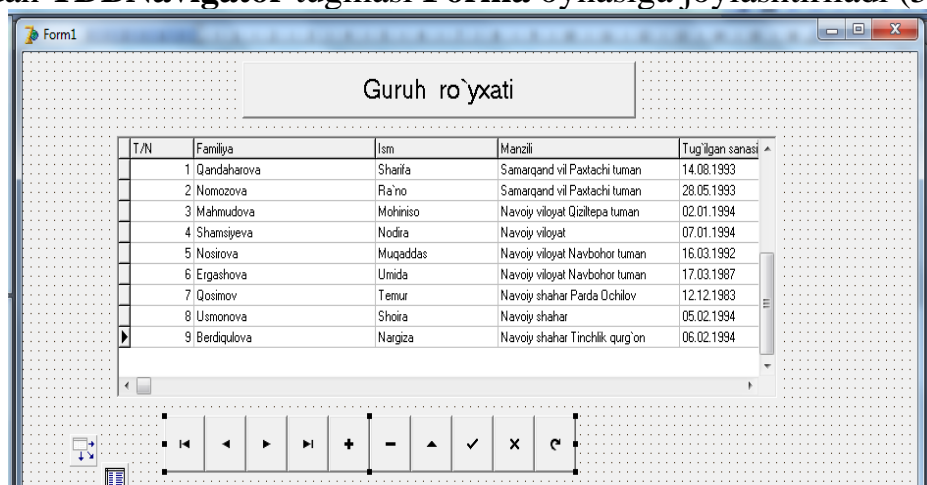


5.81-rasm.

TDBNavigator komponenti

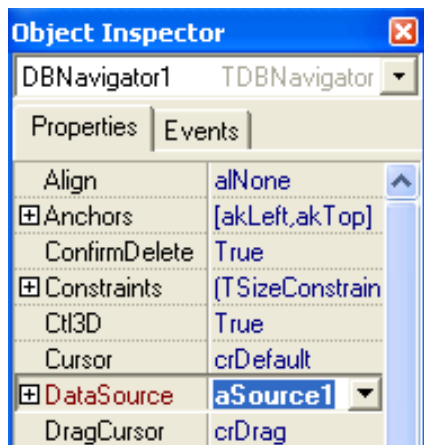
MB jadvalida ma'lumotlarni surish, o'chirish, yozuvni siljitish va taxrirlash uchun **Data Controls** komponentalar palitrasida maxsus **TDBNavigator** tugmasi mavjud. Bu tugmani formadagi MB jadvaliga quyidagi tartibda o'rnatish mumkin:

MB jadvali formasi ekranga chaqiriladi; **Data Controls** komponentalar palitrasidadan **TDBNavigator** tugmasi **Forma** oynasiga joylashtiriladi (5.86-rasm):



5.87-rasm.

TDBNavigator tugmasining **DataSource** xossasiga **DataSource1** o'rnatiladi (5.88-rasm):



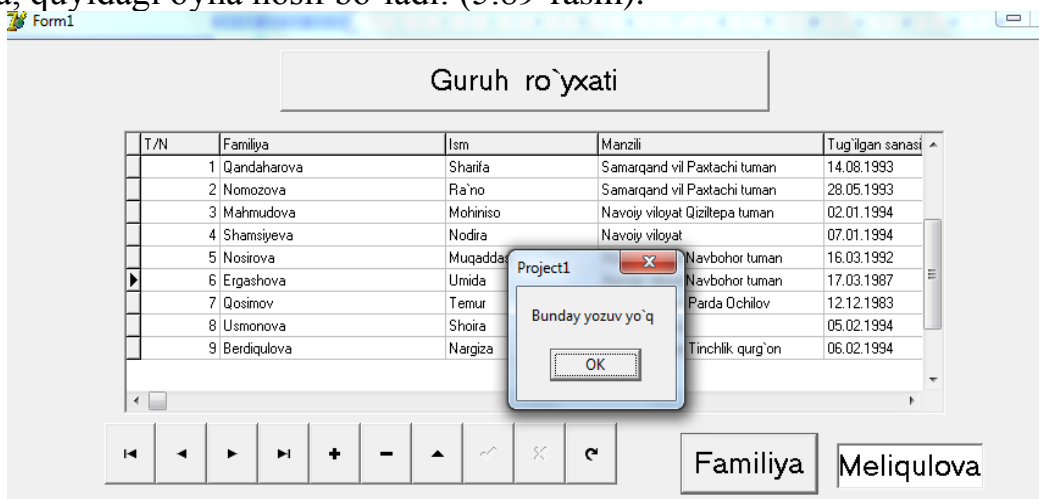
5.88-rasm.

Menyudan **File-Save Project As** buyrug‘i tanlanadi (Oldin forma keyin loyiha saqlanadi) va **F9** orqali loyiha oynasi ishga tushiriladi.

Jadvalga kiritilgan familiya, ism va boshqa ma‘lumotlar bo‘yicha izlab **TDBNavigator** tugmasining ko‘rsatkichi yordamida topiladi:

*if not table1.Locate(‘Familiya’,edit1.text,[]) then
showMessage(‘Bunday yozuv yo‘q’);*

Natijada, quyidagi oyna hosil bo‘ladi: (5.89-rasm):



5.89-rasm.

Mustaqil bajarish uchun topshiriqlar

1. Talabalarning olgan baholari haqida ma‘lumot beruvchi ma‘lumotlar bazasini yarating.
2. O‘quv jarayonini boshqaruvchi ma‘lumotlar bazasini yarating.

FOYDALANILGAN ADABIYOTLAR

1. O‘zbekiston Respublikasi Prezidentining Farmoni. O‘zbekiston Respublikasini yanada rivojlantirish bo‘yicha harakatlar strategiyasi to‘grisida. (*O‘zbekiston Respublikasi qonun yujjatlari to‘plami, 2017 y., 6-son, 70-modda.*)
2. M. Aripov, M.Muhammadiyev. Informatika, information texnologiyalar. Darslik.T.:TDYuI, 2005 y.
3. A.R. Azamatov, B.Boltayev. Algoritmash va dasturlash asoslari. O‘quv qo‘llanma.T.:”Cholpon”. 2013 y.
4. Sattorov A. Informatika va axborot texnologiyalari. Darslik. T.:”O‘qituvchi”, 2011 y.
5. Sh. A.Nazirov, M.M.Musayev, A.Ne`matov, R.V.Qobulov. Delphi tilida dasturlash asoslari. Toshkent. 2007- yil.
6. P.Karimov, S.Irisqulov, A. Isabayev. Dasturlash. Toshkent. “O‘zbekiston”. 2003-yil.
7. П. Дарахвелидзе, Э. Марков. Программирование в Delphi7. Учебник. Санкт- Петербург, “БХВ-Петербург” 2003 г.
8. M.J. Yunusova, A.B. Rahimov. Dasturlash bo‘yicha praktikum. Toshkent. “Ilm ziyo”. 2011-yil.
9. R.A.Ro‘ziyev, G‘.R.Yodgorov, O‘.M.Mirsanov “Delphi dasturlash tilini o‘rganish” (uslubiy qo‘llanma). Navoiy 2012 yil.
10. Иллюстрированный самоучител по Delphi 7 для начинающих. Lib.xost.ru. Kutubxonasi
11. Фленов Михаил. Библия для программиста в среде Delphi. 2002 г.
12. А.Я.Архангельский. Все о Delphi. Bel&Chen Co 2002 г.
13. В.Т.Безручко. Практикум по курсу информатики. М. : «Финансы и статистика», 2004 г.
14. <http://delphi-prg.ru> - dasturlash tillari. Delphi darslari sayti.
15. www.delphiplus.org – axborot texnologiyalari kunlik yangiliklari sayti.
16. <http://vt.obninsk.ru/content/mat/34> - Tizimli dasturiy ta'minot.
17. http://vit.iatp.by/software/s2_0.htm -Tizimli dasturiy ta'minot mavzusi bo'yicha elektron qo'llanma.
18. www.search.re.uz - O`zbekistonning axborotlarni izlab topish tizimi.
19. www.ictcouncil.gov.uz -Kompyuterlashtirishni rivojlantirish bo'yicha Vazirlar Maxkamasi muvofiqlashtiruvchi Kengashining sayti.
20. www.microsoft.ru
21. www.ziyonet.uz

MUNDARIJA

Soʻz boshi.....	3
I-BOB. DASTURLASH TEXNOLOGIYALARI.....	5
1.1 Dasturni qayta ishlash bosqichlari.....	5
1.2 Algoritm tushunchasi	6
1.3 Algoritmning asosiy xossalari	7
1.4 Algoritmni tasvirlash usullari	8
1.5 Chiziqli algoritmlar.....	9
1.6 Tarmoqlanuvchi algoritmlar	11
1.7 Takrorlanuvchi algoritmlar	12
1.8 Massivlarni algoritmlash	13
II- BOB. DELPHI DASTURLASH TILI.....	19
2.1 Delphi dasturlash tili tarixi.....	19
2.2 Delphi dasturlash tilida konsol ilovasida ishlash.....	21
2.3 Delphi dasturlash tili muhiti.....	22
2.4 Loyiha va modul tushunchasi.....	30
2.5 Delphida komponentalar roʻyxati.....	32
III-BOB. DELPHI TILIDA DASTURLASH.....	48
3.1 Dasturlash tili elementlari.....	48
3.2 Chiziqli jarayonlarni dasturlash.....	53
3.3 Tarmoqlanuvchi jarayonlarni dasturlash.....	61
3.4 Takrorlanuvchi jarayonlarni dasturlash.....	67
3.5 Massivlar ustida amallar.....	76
3.6 Protsedura va funksiya.....	89
IV BOB. DELPHI DASTURINING GRAFIK VA MULTIMEDIA IMKONIYATLARI.....	99
4.1 Grafik obektlar bilan ishlash.....	99
4.2 Multimedia dasturlarini yaratish vositalari.....	114
V BOB. DELPHI VIZUAL DASTURLASH VOSITASIDA MAʼLUMOTLAR BAZASINI YARATISH TEXNOLOGIYALARI..	119
5.1 Delphi muhitida DataAccess va ADO komponentalar panelidan foydalanish	119

5.2	Delphi muhitida ADO komponentalar panelidan ADO Connection va ADOQuery komponentalaridan foydalanish.....	134
5.3	Delphi muhitining BDE Administrator va Database Desktop bo‘limlari bilan ishlash.....	157

ОГЛАВЛЕНИЕ

Предисловие.....	3
I –Глава. ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ.....	5
1.1 Степени переработки программы.....	5
1.2 Понятие алгоритма.....	6
1.3 Основные свойства алгоритма.....	7
1.4 Методы образования агоритма.....	8
1.5 Линейные алгоритмы.....	9
1.6 Структурные алгоритмы	11
1.7 Повторяющиеся алгоритмы.....	12
1.8 Алгоритмизация массивов.....	13
II- Глава. ПРОГРАММНЫЙ ЯЗЫК DELPHI.....	19
2.1 История программного языка	19
2.2 Работа над приложением конзола в языке программирования Delphi	21
2.3 Среда программного языка Delphi	22
2.4 Понятии проект и модуль.....	30
2.5 Список компонентов Delphi	32
III- Глава. ПРОГРАММИРОВАНИЕ В ЯЗЫКЕ DELPHI	48
3.1 Элементы программного языка.....	48
3.2 Программирование линейных процессов	53
3.3 Программирование структурных алгоритмов.....	61
3.4 Программирование повторяющихся алгоритмов.....	67
3.5 Задачи над массивами.....	76
3.6 Процедура и функция.....	89
IV- Глава. ГРАФИЧЕСКИЕ И МУЛЬТИМЕДИЙНЫЕ ВОЗМОЖНОСТИ ПРОГРАММЫ DELPHI	99
4.1 Работа над графическими объектами.....	99
4.2 Средства разработки мультимедийного программного обеспечения.	114
V-Глава. ТЕХНОЛОГИИ СОЗДАНИЯ БАЗ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ ВИЗУАЛЬНОГО ПРОГРАММИРОВАНИЯ DELPHI	119
5.1 Использование компонентов панели DataAccess и ADO в среде Delphi	119

5.2	Использовании компонентов панели ADO, компонентов ADO Connection и ADOQuery в среде Delphi	134
5.3	Работа с разделами BDE Administrator и Database Desktop среды Delphi	157