

**O'ZBEKISTON ALOQA, AXBOROTLASHTIRISH VA
TELEKOMMUNIKATSIYA
TEXNOLOGIYALARI DAVLAT QO'MITASI
TOSHKENT AXBOROT TEXNOLOGIYALAR UNIVERSITETI
NUKUS FILIALI**

Kompyuter Injiniring fakulteti

3b Informatcion texnologiyalar yo'nalishi talabasi

Oteuliev Janibekning

C++ dasturlash tili

fanidan

Mustaqil ishi

Mavzu: Delphi dasrturlash tilida funksiya grafiklari bilan ishlovchi dastur yaratish.

Bajargan:

Oteuliev J.

Qabul qilgan:

Tleuov K.

Nukus-2014

Mundarija:

Kirish-----	4
1. Delphida kompilyatsiyasi jarayoni va dastur tuzilishi-----	5
2. Delphining grafik imkoniyatlari-----	6
3. Canvas metodida obektlar chizish-----	7
4. Delphi sonli berilganlarni grafik tasvirlash.-----	13
5. Funksiya grafiklarini chizish dasturi-----	15
6. Dastur protseduralari va natijalari tasnifi-----	17
Xulosa-----	21
Foydalanilgan adabiyotlar-----	22

Kirish

Hozirgi vaqtga kelib kompyuter olamida ko'plab dasturlash tillari paydo bo'ldi. Ular Beysik, Paskal, Ci va boshqa dasturlash tillaridir. Xo'sh ulardan qay biri yaxshiroq? Albatda bu savolga Pascal dasturlash tili deyish mumkin. Chunki u boshqa dasturlash tillaridan dasturlashni o'rganish uchun qulay xisoblanadi. Bu til 1969 yil N.Virt tomonidan yaratilgan bo'lib, keyinchalik amerikaning Borland firmasi tomonidan qayta ishlandi va uni Turbo Pascal deb nomlagan. "Turbo Pascal"ni qayta ishlash natijasida ob'ektlilik dasturlash yo'lga qo'yildi va uni Object Pascal deb atala boshlandi. Hisoblash texnikasi va texnologiyasining rivojlanishi natijasida Borland firmasi tomonidan yangi Delphi dasturlash tili yaratildi.

Delphi dasturlash tili Windows uchun mo'ljallangan bo'lib, uning birinchi versiyasi Windows 3.1 operatsion sistema qobig'ida ishlangan. Windows 95 operatsion sistema yaratilgandan so'ng, 16 razryadli Delphi 2, keyinroq 32 razryadli Delphi 3 versiyasi yaratildi. Windows 98 operatsion sistema uchun Delphi ning to'rtinchi, beshinchi va ettinchi versiyalari paydo bo'ldi. Hozirgi kunga kelib Delphi 2,3,5,6,7 2005,2006,2008,2010 va 2013 versiyalari ishlab chiqarildi. Delphi obyektlik dasturlash tilini o'zida mujassam etgan. Delphi vizual proektlar, turli holat va protseduralarni qayta ishlash va dasturlarni qayta ishlashda vatqdan yutish va boshqalarni o'z ichiga oladi.

Avvaldan ushbu dasturlash muhiti faqatgina Microsoft Windows amaliyot tizimi uchun dasturlar yaratishga mo'ljallangan, keyinchalik esa GNU/Linux hamda Kylix tizimlari uchun moslashtirildi, lekin 2002-yilgi Kylix 3 sonidan so'ng ishlab chiqarish to'xtatildi, ko'p o'tmay esa Microsoft.NET tizimini qo'llab quvvatlashi to'g'risida e'lon qilindi.

Uning quyidagi kompilyatorlari bor: Embarcadero Delphi, Free Pascal, Virtual Pascal, GNU pascal,...va boshqalar. Delphida tuzilgan mashhur programmalarga, Total commander, JBuilder, FL Studio, The Bat larni misol qilish mumkin. Hozirgi kunda bu til ancha rivojlangan bo'ldi, hattoki PHP tillari bilan ham bo'g'lanib ishlay oladi.

Delphi da dastur tuzish zamonaviy visual loyihalash texnologiyalariga asoslangan bo'lib, unda dasturlashning ob'ektga yo'naltirilgan g'oyasi mujassamlashtirilgan. Delphi 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 tilida dastur kompilyatsiyasi.

Delphi tilida dastur matni bosh va asosiy bo'limdan tashkil topadi. Bosh bo'lim dastur nomi va o'zgaruvchilar, o'zgarmlar, massivlar, belgilar(metkalar), protseduralar va funktsiyalarni

tavsiflashdan iborat bo'ladi. Asosiy bo'lim dastur tanasi deyilib, unda dasturda bajariladigan hamma operatorlar ketma-ketligi beriladi va u Begin (boshlamoq) so'zi bilan boshlanib End (tugash) so'zi bilan tugaydi. Umumiy holda dastur strukturasi quyidagi ko'rinishga ega:

Program <dastur nomi>;

Uses <Foydalanadigan bibliotekalar (modullar) ro'yhati>;

Label <Ishlatiladigan belgilar(metkalar) ro'yhati>;

Const <Ishlatiladigan o'zgarmaslarni aniqlash>;

Type <Yangi turlarni aniqlash>;

Var <o'zgaruvchilarni e'lon qilish>;

<Protsedura va funkziyalarni aniqlash>

Begin

<Bajariladigan operatorlar ketma ketligi>

End.

Forma oynasida ilovalar yaratiladi. Object Inspector oynasi obekt xossalarini taxrirlash uchun xizmat qiladi. Obekt xossalari bu - obektga berilgan xarakteristika bo'lib, uning ko'rinish, joylashishi va holatidir. Masalan, Width va Height xossalari forma o'lchamini, top va Left esa formaning ekrandagi holati, Caption - sarlovha matnini aniqlaydi.

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

Delphida dasturlash ikkita o'zaro tasir etuvchi bir-biri bilan bog'liq jarayon asosida tashkil qilinadi:

-dasturni vizual loyihalash jarayoni;

-dastur kodlarini kiritish (yozish) jarayoni.

Kodlarni yozish uchun maxsus kod oynasi mavjud bo'lib, u dastur matnini kiritish va taxrirlash uchun mo'ljallangandir. 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. Dastur loyihasini ishlashi mobaynida dasturchi kerakli dastur operatorlarini kiritib, formani loyiha bo'yicha akslantiradi. Delphida dasturlash forma oynasini tashkil etishdan boshlanadi.

Oddiy dastur ilovasini yaratish ketma-ket File=> New=> Applisition buyrug'ini berish bilan boshlanadi. Bu buyruqni berishdan oldin ikkita asosiy ishni bajarish lozim:

-papka tashkil etish;

-tizimni to'g'rilash.

Papka tuzing, masalan, My_Delhp nomli. My_Delhp papkasi ichida yana o'z dasturngizni saqlash uchun papka ochish, masalan Pgm_1.

Delphi muhitining standart nastroykasiga o'zgartirish kiritish uchun Tols=>Environment Options menyu buyrug'ini berish va muloqat darchasidan kerakli o'zgarishlarni bajarish lozim.

Delphi dasturlash muhitida ishlash jarayonida quyidagi kengaytmali fayllar ishlatiladi:

- loyiha fayli, kengaytmasi .dpr;
- paskal moduli fayli, kengaytmasi .pas;
- komponentalar joylashgan fayl, kengaytmasi .dcu;
- formalar joylashgan fayl, kengaytmasi .dfm;
- malumotlar bazasi fayli, kengaytmasi .dbf.

Tayyorlanadigan Delphi dastur uchta asosiy etapdan o'tadi:

- kompilyatsiya;
- komponovka;
- bajarish.

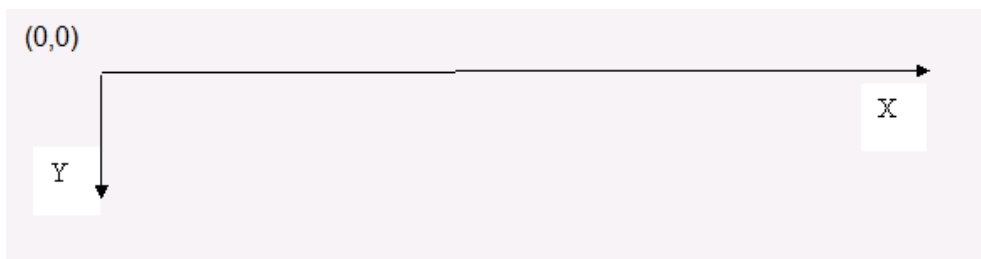
Kompilyatsiya etapida tayyorlangan dastur matni Object Pascal tiliga o'tkaziladi. Kompanovka etapida esa kerakli qo'shimcha yordamchi dasturlar va ostdasturlar unga birlashtiriladi. F9 tugmasini bosish bilan Save UnitAs dialog oynasi paydo bo'ladi va sizdan Unit.pas moduli uchun fayl nomini va joylashadigan papkani ko'rsatishingizni so'raydi. Agar joy ko'rsatilmasa Delphi avtomatik ravishda dasturngizni Bin papkasiga joylashtiradi. Yaxshisi siz bu papkani o'z ishchi papkangiz nomiga almashtiring, masalan My_Delph. Dastur kompilyatsiya qilinishi paytida Delphi sistemasi pas, dfm va dcu kengaytmali modullar tuzadi. .pas kengaytmali fayl 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 komplyator tamonidan tashkil qilinadi va yagona ishchi (bajariluvchi) .exe kengaytmali fayl tashkil qilishga baza yaratadi.

Delphida grafik imkoniyatlar

Delphi dasturchiga grafik dasturlar sxema, chertej, illyustratsiyalar yaratishga imkon beradi. Dastur grafikani ob'ekt (forma eki 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) instruksiyasi dastur oynasida turtburchak 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. Piksel holati uning gorizontali (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 i width xossalari va **formaning** ClientHeight va Clientwidth xossalari orqali aniqlash mumkin.

Canvas metodi imkoniyatlari

Qalam

Qalam geometrik figuralarni chizish uchun ishlatiladi. CHiziq ko'rinishi Tren ob'ektining quyidagi jadvalda ko'rsatilgan xossalari orqali aniqlanadi. Tren (qalam) xossalari.

Xossa	Ta'rifi
Color	CHiziq rangi
Width	CHiziq kalinligi
Style	CHiziq ko'rinishi
Mode	Akslantirish rejimi

Grafika bilan ishlashda albatta chiziladigan ranglarning qiymatlarini berish kerak bo'ladi. Buni color xossasi orqali beriladi. Quyidagi jadvalda color xossasi qiymati sifatida beriluvchi nomlangan konstantalar sanab o'tilgan.

Color xossasi qiymatlari.

Konstanta	Rang	Konstanta	Rang
clBlack	Qora	clSilver	Serebristy
clMaroon	Kashtanovy	clRed	Qizil
clGreen	YAshil	clLime	Salatnyy
clOlive	Olivkovyy	clBlue	Ko'k (zangori)

clNavy	Tim-ko‘k	clFuchsia	YArko-rozovыy
clPurple	Rozovy	clAqua	Biryuzovy
clTeal	Zeleno-goluboy	clWhite	Oq
clGray	Kul rang		

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.

Konstanta	CHiziq ko‘rinishi
psSolid	Uzluksiz chiziq
psDash	Punktir chiziq, uzun shtrixlar
psDot	Punktir chiziq, qisqa shtrixlar
psDashDot	Punktir chiziq, uzun va qisqa shtrixlar ketma ketligi
psDashDotDot	Punktir chiziq, bitta uzun va ikkita qisqa shtrixlar ketma ketligi
psClear	CHiziq aks etmaydi

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

Konstanta	CHiziq rangi
pmBlack	Qora, Pen. Color xossasi qiymatiga bogliq emas
pmWhite	Ok, Pen. Color xossasi qiymatiga bog‘liq emas
pmCopy	CHiziq rangi Pen. Color xossasi qiymatiga bog‘liq

pmNotCopy	CHiziqli rangi Pen. Color xossasi qiymatiga invers
pmNot	CHiziqli rangi sohaning mos nuqtasi rangiga invers

Muyqalam

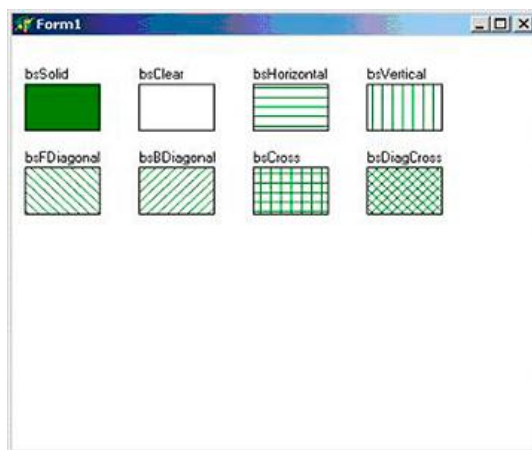
Muyqalam (Canvas.Brush) yopiq sohalarni chizish va soha ichini bo'yash uchun mo'ljallangan usullardan foydalaniladi. Muyqalam ob'ekt jadvalda ko'rsatilgan ikki xossaga ega. TBrush (muyqalam) xossalari.

Xossa	Ta'rif
Color Style	Epiq sohani bo'yash rangi 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.

Konstanta	Soha bo'yash uslubi
bsSolid	Uzluksiz bo'yash
bsClear	Soha bo'yalmaydi
bsHorizontal	Gorizontal shtrixlash
bsVertical	Vertikal shtrixlash
bsFDiagonal	Diagonal shtrixlash, oldinga og'ish
bsBDiagonal	Diagonal shtrixlash, orqaga og'ish
bsCross	Katakli gorizontal-vertikal shtrixlash
bsDiagCross	Katakli diagonal shtrixlash

Misol tariqasida sohalarni bo'yash usullari dasturini keltiramiz.



Matnni chiqarish

Grafik ob'ekt yuzasiga matn chiqarish uchun TextOut usuli qo'llaniladi. Bu usulni chaqirish nstruksiya quyidagi ko'rinishga ega:

Ob'ekt.Canvas.TextOut(x, u, Tekst)

Matn shrifti Font xossasi qiymati bilan aniqlanadi. Font xossasi TFont tipidagi ob'ekdir.

Quyidagi jadvalda TFont ob'ekti xossalari keltirilgan.

TFont ob'ekti xossalari

Xossa	Ta'rifi
Name	SHrift nomi, masalan Arial
Size	SHrift punktlarda kattaligi
Style	Simvollar chiqarish uslubi. Quyidagi konstantalar orqali beriladi: fsBold (polujirnyy), fsItalic (kursiv), fsUnderline (podcherknuty), fsStrikeOut (perecherknuty).
Color	Bu xossa bir necha uslublarni kombinatsiyasini olishga imkon beradi. Masalan: Ob'ekt. Canvas . Font := [fsBold, fs Italic] Simvollar rangi.

Matn chiqarish sohasi muyqalam joriy rangiga buyaladi. SHuning uchun matn chiqarishdan oldin Brush. Color xossasiga bsClear qiymatini yoki soha rangiga mos qiymatni berish lozim.

Misol:

```
with Form1.Canvas do begin
Font.Name := 'Tahoma';
Font.Size := 20;
Font.Style := [fsItalic, fsBold] ;
Brush.Style := bsClear;
TextOut(0, 10, 'Borland Delphi 6');
end;
```

Textout uslubi orqali matn ekranga chiqarilgandan so'ng qalam matn chiqarish sohasining yuqori o'ng burchagiga keltiriladi.

Agar matn uzunligi ma'lum bo'lmasa, chiqarilgan matn o'ng chegarasi koordinatalarini PenPos xossasiga murojaat qilib aniqlash mumkin.

Misol:

```
with Form1.Canvas do begin
  TextOut(0, 10, 'Borland ');
  TextOut(PenPos.X, PenPos.Y, 'Delphi ');
end;
```

Grafik primitivlarni chizish usullari

CHiziq

To'g'ri chiziq LineTo usuli orqali amalga oshiriladi.

Komponent.Canvas.LineTo(x,u)

LineTo usuli qalam joriy pozitsiyasidan berilgan koordinatali nuqttagacha to'g'ri chiziq chizadi. Boshlangich nuqtani kerakli nuqtaga ko'chirish uchun MoveTo usulidan foydalanish mumkin.

Tutashgan chiziq

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

Polyline usuliga misol tariqasida ma'lum qiymat o'zgarishi grafigini chizuvchi protseduracini keltiramiz:

```
procedure TForm1.Button1Click(Sender: TObject);
var
  gr: array[1..50] of TPoint;
  x0,y0: integer;
  dx,dy: integer;
  i: integer; begin
  x0 := 10; u0 := 200; dx :=5; dy := 5;
  for i:=1 to 50 do begin
    gr[i].x := x0 + (i-1)*dx;
    gr[i].y := y0 - Data[i]*dy;
  end;
  with form1.Canvas do begin
    MoveTo(x0,y0); LineTo(x0,10);
    MoveTo(x0,y0); LineTo(200,y0);
    Polyline(gr);
  end;
end;
```

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

Aylana va ellips

Aylana yoki ellips chizish uchun Ellipse usuli chaqiriladi. Usulni chaqirish instruksiyasi umumiy ko'rinishi:

Ob'ekt.Canvas.Ellipse(x1,y1, x2,u2).

Bu erda x1, y1, x2, u2 – ellipsni o'z ichiga olgan minimal turtburchak koordinatalari. Agar turtburchak kvadrat bo'lsa aylana chiziladi.

Ellips yoyini chizish uchun Arc usuli qo'llaniladi va u quyidagi umumiy ko'rinishga ega:
Ob'ekt.Canvas.Arc(x1,y1,x2,u2,x3,u3,x4,u4)
Bu erda:

- x1, y1, x2, u2 - yoyga tegishli bo'lgan ellips yoki aylana parametrlari;
- x3, u3 - yoy boshlang'ich nuqtasi parametrlari;
- x4, u4 - so'ngi nuqtasi parametrlari.

Yoy soat miliga teskari tartibda chiziladi.

To'rtburchak

To'rtburchak Rectangle usuli bilan chizilib, bu usulni chaqirish instruksiyasi umumiy ko'rinishi quyidagicha:

Ob'ekt.Canvas.Rectangle(x1, y1,x2, y2)

Bu erda x1, y1 va x2, u2 — chapgi yuqori va o'nggi pastgi burchaklar koordinatalari.

RoundRec usuli burchaklari yumaloq to'rtburchak chizishga imkon beradi. RoundRec usulini chaqirish instruksiyasi quyidagi kurinishga ega:

Ob'ekt.Canvas.RoundRec(x1,y1,x2, u2, x3, u3)

Bu erda:

- x1, y1, x2, u2 – turtburchak parametrlari;
- x3 i u3 — chorak kismi yumalok burchak chizish uchun ishlatiladigan ellips kattaligi.

Ya'na ikki usul muyqalamdan foydalanib to'rtburchak chizishga imkon beradi. FillRect usuli ichi bo'yalgan to'rtburchak chizadi, FrameRect - faqat kontur. Bu usullarda faqat bitta parametrga ega -TRect tipidagi struktura. Quyidagi misolda FillRect va FrameRect usullari orqali forma yuzasiga qizil turtburchak soha va yashil konturli to'rtburchak chizuvchi protsedura keltirilgan.

```
procedure TForm1.Button1Click(Sender: TObject);
var
r1, r2: TRect;
begin
r1 := Rect(20,20,60,40);
r2 := Rect(10,10,40,50);
with form1.Canvas do begin
Brush.Color := clRed;
FillRect(r1);
Brush.Color := clGreen;
FrameRect(r2);
end;
end;
```

Ko'pburchak

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.

Quyida polygon usuli yordamida uchburchak chizish protsedurasi keltirilgan:

```

procedure TForm1.Button2Click(Sender: TObject);
var
pol: array[1..3] of TPoint;
begin
pol[1].x := 10;
pol[1].y := 50;
pol[2].x := 40;
pol[2].y := 10;
pol[3].x := 70;
pol[3].y := 50;
Form1.Canvas.Polygon(pol);
end;

```

Sektor

Ellips yoki aylana sektori pie usuli bilan chizilib, chaqirish instruksiyasi quyidagi umumiy ko'rinishga ega:

Ob'ekt. Canvas.Pie(x1,y1,x2,y2,x3,u3,x4,u4)

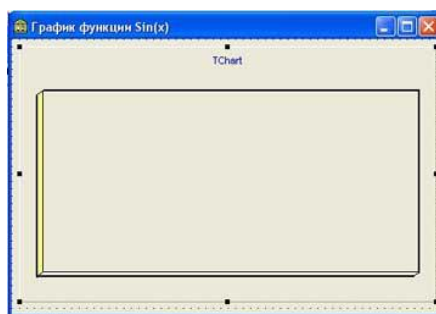
Bu erda:

x1, y1, x2, y2 - ellips yoki aylana parametrlari;

- x3, u3, x4, u4 - sektor chegarasini tashkil qiluvchi to'g'ri chiziqlar oxirgi nuqtalari koordinatalari.

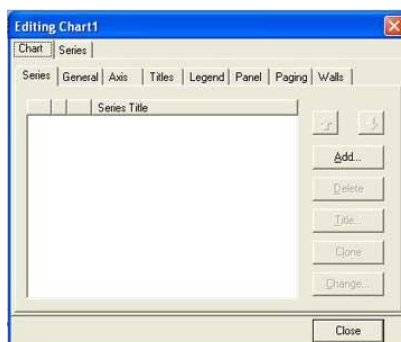
Delphi sonli berilganlarni grafik tasvirlash

Sonli berilganlarni grafik tasvirlash uchun Chart komponentidan foydalaniladi. Ushbu komponent Additional sahifasida joylashgan. Komponent ko'p sonli maxsus xossalarga, hodisa va metodlarga ega. Eng sodda metod bo'lgan $\sin(x)$ funktsiya grafigini qurishni ko'rib o'tamiz. Birinchi komponent formaga joylashtiriladi. Uning ustida sichqoncha o'ng tugmasi bosilib, yordamchi menyu chaqiriladi. Ushbu menyuning Edit Chart opsiyasi yordamida komponent redaktorining ko'p sahifali oynasi chaqiriladi.



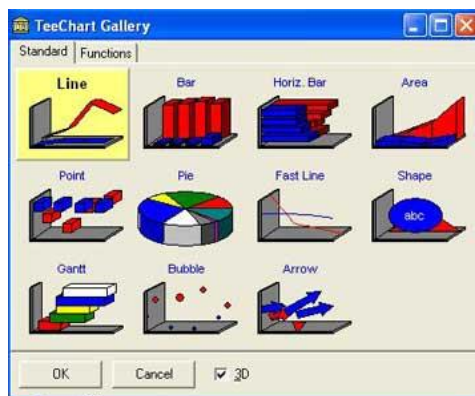
Rasm 1.

Rasm 1. ni ustiga sichqoncha bilan bosib rasm 2. ni xosil qilamiz.



rasm 2.

Ushbu oynaning Series zakladkasi TcharSeries sinfi ob'ektlariga murojaat imkoniyatini beradi. Ular sonlarni grafik tasvirlash vazifasini bajaradi. Berilganlarni tasvirlash uchun eng kamida bitta seriya yaratilishi kerak. Buning uchun redaktor oynasida Add tugmasi bosiladi va grafikning mos tipi tanlanadi.

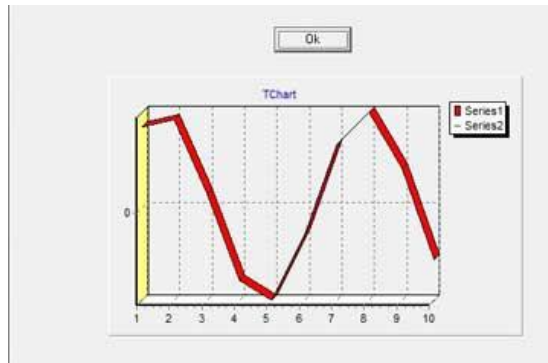


rasm 4.

Redaktor yopilgandan so'ng komponent grafikning taxminiy ko'rinishiga ega bo'ladi.

Ammo uning haqiqiy ko'rinishi ishlayotgan dastur da yaratilyvchi berilganlarga bog'liq bo'ladi va seriyaga TCharSeries ob'ekting AddX, AddY, yoki AddXY metodlari qo'shiladi (seriyaga murojaat qilish uchun Series[n] indekslangan xossadan foydalanish mumkin. Birinchi yaratilgan seriya 0 indeksga, ikkinchisi 1 va hokazo indeksga ega. Quyida sin(x) funksiyasi grafigini yaratish formasining OnActivate hodisasini qayta ishlovchi prosedurasini ko'rib o'tamiz.

Misol 1. sin(x) grafigi.



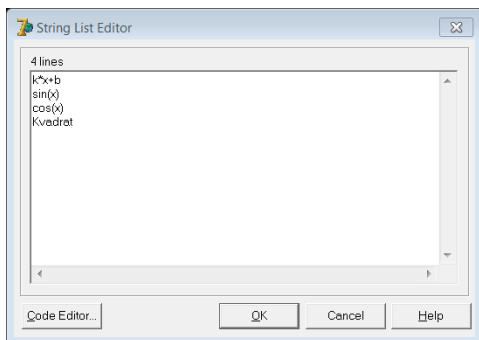
```
procedure TForm1.Button1Click(Sender: TObject);
var i:integer;
begin
  for i:= 1 to 10 do
    Chart1.Series[0].AddXY(i,sin(i));
  end;
end.
```

Funksiya grafiklari bilan ishlovchi dastur.

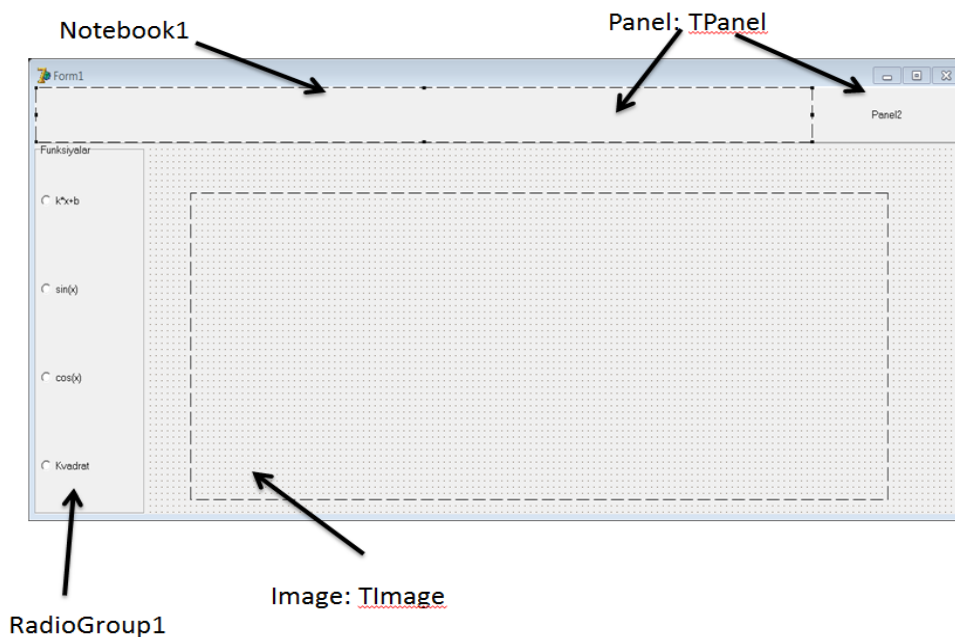
Delpi tilida grafik imkoniyatlari bilan tanishib chiqdiq. Endi yuqoridagi Canvas metodidan foydalanib funksiya grafiklarini chizuvchi dastur tuzamiz. Buning uchun dastlab formaga kerakli bo'lgan komponentalar ni qo'yamiz bu komponentalar quyidagilar:

TForm1 = class(TForm)	Image2: TImage;
Panel1: TPanel;	Image3: TImage;
RadioGroup1: TRadioGroup;	Image4: TImage;
Notebook1: TNotebook;	Label5: TLabel;
Notebook2: TNotebook;	Shape1: TShape;
Image1: TImage;	Shape2: TShape;
Panel2: TPanel;	Shape3: TShape;
Button1: TButton;	Shape4: TShape;
Label1: TLabel;	Label6: TLabel;
Edit1: TEdit;	Edit5: TEdit;
Edit2: TEdit;	Edit6: TEdit;
Label2: TLabel;	Edit7: TEdit;
Edit3: TEdit;	Label7: TLabel;
Label3: TLabel;	Label8: TLabel;
Edit4: TEdit;	Label9: TLabel;
Label4: TLabel;	

RadioGroup1 ning xususiyatlar qatoridan (**Object Inspector → Properties**) Itemsxususiyati tanlanib unga quyidagilar yoziladi:



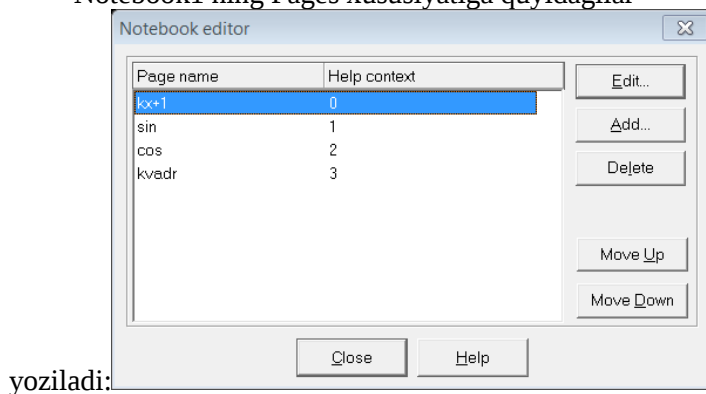
Forma ko'rinishi quyidagicha bo'ladi:



Komponentalarning har birining **Align** xususiyatlarini o'zgartirib chiqamiz.

Bu yuqoridagi rasimdagidek bo'ladi. so'ng **Image1** ning **Align** xususiyatini **alClient** ga o'zgartiriladi. Ranglar kerakli ranga o'zgartiriladi (Properties→Color).

Notebook1 ning Pages xususiyatiga quyidagilar



yoziladi:

Bu Notebook1 ning har bir funksiya uchun alohida varaq oshilishini taminlaydi. Panel2 ga Button yani tugmasha tashlaymiz.

RadioGroup1 ning OnClick xususiyatiga Notebook1.PageIndex:=RadioGroup1.ItemIndex; ni yozamiz. Bu bizga radiogroupni qaysi birini belgilasak Notebook1 ning ham o'cha varag'I ochilishini taminlaydi.

Notebook1 ning harbir varag'iga harbir funksiya haqidagi kiritilishi kerak bo'lgan malunotlarni kiritamiz:

Chiziqli funksiya uchun:

$Y = k * X + L$ funksiyasi. Koeffitsientlar k va l ni kiriting:

$Y =$ $X +$

Sin(x) funksiya uchun:

$Y = \sin(aX)$ funksiyasi. Koeffitsient - a ni kiriting

$Y = \sin($ $* X)$

Cos(x) funksiya uchun:

$y = \cos(ax)$ funksiyasi. Koeffitsient - a ni kiriting

$Y = \cos($ $* X)$

Kvadrat funksiya uchun:

$y = ax^2 + bx + c$ kvadrat funksiyasi. Koeffitsientlar a, b, c larni kiriting

$Y =$ $X^2 -$ $X +$

Umumiy kod va Protseduralar quyidagicha bo'ladi:

Protseduralar

unit Unit1;

interface

uses

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

type

TForm1 = class(TForm)

Panel1: TPanel;

RadioGroup1: TRadioGroup;

Notebook1: TNotebook;

Notebook2: TNotebook;

Image1: TImage;

Panel2: TPanel;

Button1: TButton;

Label1: TLabel;

Edit1: TEdit;

Edit2: TEdit;

Label2: TLabel;

Edit3: TEdit;

Label3: TLabel;

Edit4: TEdit;

Label4: TLabel;

Image2: TImage;

Image3: TImage;

Image4: TImage;

Label5: TLabel;

```

Shape1: TShape;
Shape2: TShape;
Shape3: TShape;
Shape4: TShape;
Label6: TLabel;
Edit5: TEdit;
Edit6: TEdit;
Edit7: TEdit;
Label7: TLabel;
Label8: TLabel;
Label9: TLabel;
Label10: TLabel;
procedure Button1Click(Sender: TObject);
procedure RadioGroup1Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;
var
    Form1: TForm1;
    xo,yo,i,j,z:integer;
    k,l,a,b,c,d,e:real;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var x1,y1:integer;
begin
    Notebook2.PageIndex:=RadioGroup1.ItemIndex;
    xo:=Notebook2.width div 2;
    yo:=Notebook2.height div 2; //koordinata o'qlarining (0,0) niqtasini belgilash
    if RadioGroup1.ItemIndex=0 then begin
        if (Edit1.Text='') or (Edit2.Text='') then showmessage('koeffitsientlar to'liq kiritilmadi')//boch
bo'lsa xatolik haqida xabar berish
        else begin
            Notebook2.PageIndex:=0;
            Image1.Canvas.FillRect(rect(0,0,2*xo,2*yo));
            Image1.Canvas.MoveTo(0,yo);
            Image1.Canvas.LineTo(2*xo,yo);
            Image1.Canvas.MoveTo(xo,0);//koordinata o'qlarini chizish
            Image1.Canvas.LineTo(xo,2*yo);
            k:=StrToFloat(Edit1.Text);
            l:=StrToFloat(Edit2.Text);
            for i:=-xo to xo do begin
                x1:=i+xo;
                y1:=round(yo-(k*i+l));//Chiziqli funksiya
                Image1.Canvas.Pixels[x1,y1]:=clred;
            end; end;

```

```

end;
if RadioGroup1.ItemIndex=1 then begin
  if Edit3.Text="" then showmessage('koeffitsientlar to"liq kiritilmadi')
  else begin
    Notebook2.PageIndex:=1;
    Image2.Canvas.FillRect(rect(0,0,2*xo,2*yo));
    Image2.Canvas.MoveTo(0,yo);
    Image2.Canvas.LineTo(2*xo,yo);
    Image2.Canvas.MoveTo(xo,0);
    Image2.Canvas.LineTo(xo,2*yo);
    a:=StrToFloat(Edit3.Text);
    for i:=-xo to xo do begin
      x1:=i+xo;
      y1:=yo-round(200*sin(a*i*pi/180)); //sin(x) funksiyasi
      Image2.Canvas.Pixels[x1,y1]:=clred;
    end; end;
  end;
if RadioGroup1.ItemIndex=2 then begin
  if Edit4.Text="" then showmessage('koeffitsientlar to"liq kiritilmadi')
  else begin
    Notebook2.PageIndex:=2;
    Image3.Canvas.FillRect(rect(0,0,2*xo,2*yo));
    Image3.Canvas.MoveTo(0,yo);
    Image3.Canvas.LineTo(2*xo,yo);
    Image3.Canvas.MoveTo(xo,0);
    Image3.Canvas.LineTo(xo,2*yo);
    b:=StrToFloat(Edit4.Text);
    //i:=-xo;
    for i:=-xo to xo do begin
      x1:=i+xo;
      y1:=yo-round(200*cos(b*i*pi/180)); //sin(x) funksiyasi
      Image3.Canvas.Pixels[x1,y1]:=clred;
    end;
    //Timer1.Enabled:=true;
  end
end;
if RadioGroup1.ItemIndex=3 then begin
  if (Edit5.Text="" or (Edit6.Text="" or (Edit7.Text="" then showmessage('Koeffitsientlar to"liq kiritilmadi')
  else begin
    Notebook2.PageIndex:=3;
    Image4.Canvas.FillRect(rect(0,0,2*xo,2*yo));
    Image4.Canvas.MoveTo(0,yo);
    Image4.Canvas.LineTo(2*xo,yo);
    Image4.Canvas.MoveTo(xo,0);
    Image4.Canvas.LineTo(xo,2*yo);
    c:=StrToFloat(Edit5.Text);
    d:=StrToFloat(Edit6.Text);

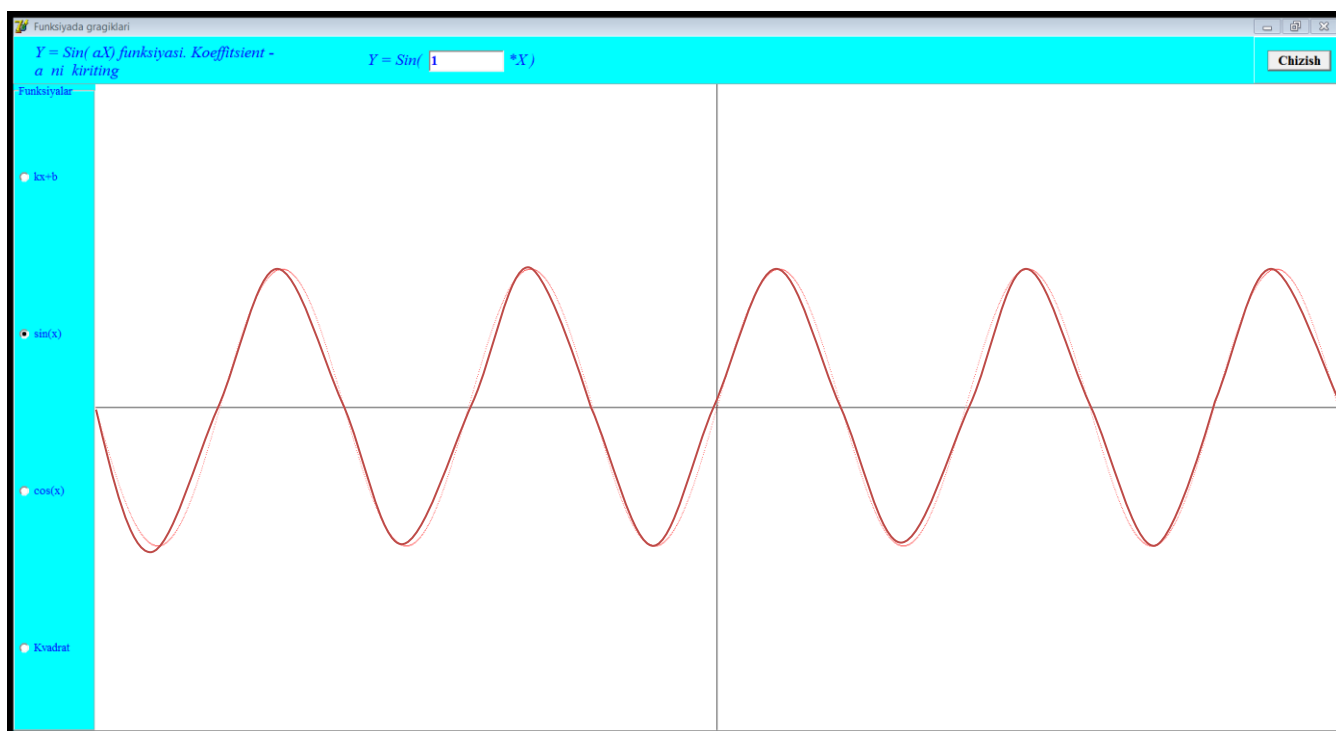
```

```

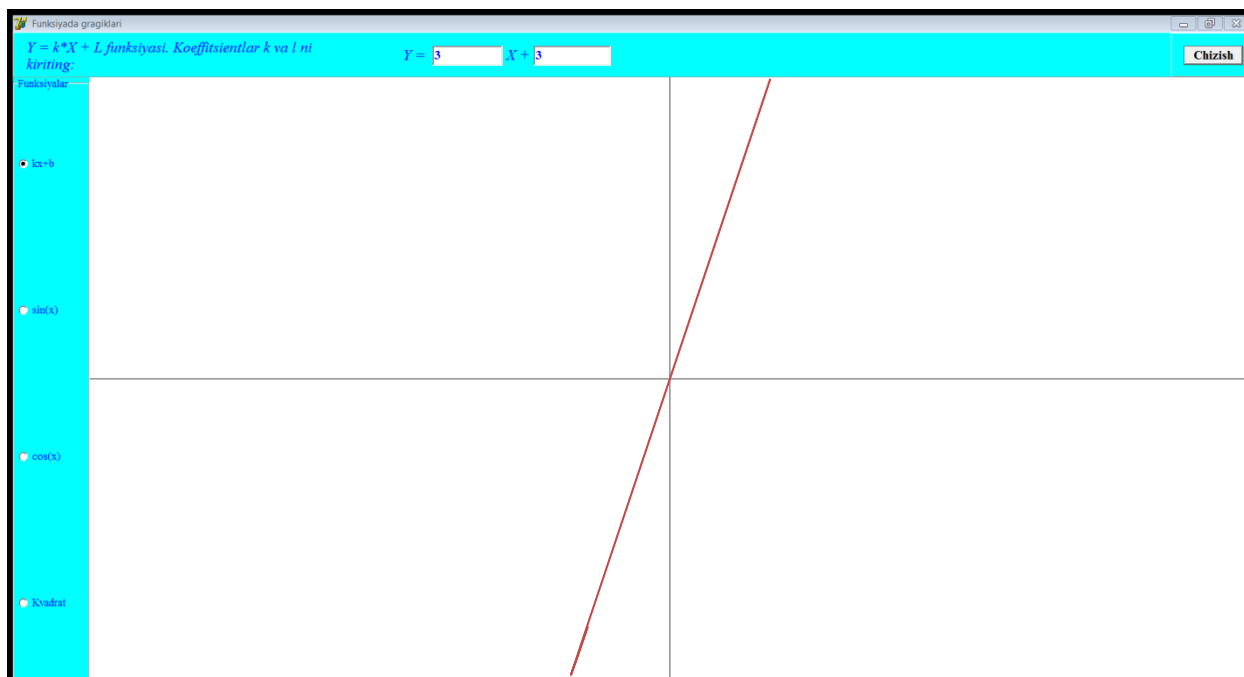
e:=StrToFloat(Edit7.Text);
  for i:=-xo to xo do begin
    x1:=i+xo;
    y1:=yo-round(c*i*i+d*i+e); // kvadrat funksiya
    Image4.Canvas.Pixels[x1,y1]:=clred;
  end; end;end;
  i:=-xo;
end;
procedure TForm1.RadioGroup1Click(Sender: TObject);
begin
  Notebook1.PageIndex:=RadioGroup1.ItemIndex;
end;
end.

```

Radiogroup ni tanlashimiz orqali ko'rsatilgan funksiyalar ning grafiklarini chizishimiz mumkin. Masalan biz $\sin(kx)$ funksiyasining grafinichizsak quyidagicha bo'ladi:



Chiziqli funksiya grafigi quyidagicha chiziladi:



Xulosa

Delphi dasturchiga grafik dasturlar sxema, chertej, illyustratsiyalar yaratishga imkon beradi. Dastur grafikani ob'ekt (forma eki Image komponentasi) yuzasiga chiqaradi. Buning uchun Canvas xususiyatidan foydalaniladi. Canvas orqali turli xil geometrik chakllar (kvadrat, aylana, ellips v.h.k) chizishimiz mumkin, shu bilan birga funksiyalarning grafiklarini chizishda ham foydalanishimiz mumkin. Grafiklarni chizganda biz yuqorida foydalangan pixels (nuqtalar qo'yish) metodidan foydalanishimiz yoki tug'ri chiziqlar chiza oluvchi MoveTo(x,y) va LineTo(x,y) juftidan ham foydalanishimiz mumkin. Bu usulda biz yuqorida foydalangan nuqtalarni bir biriga tutashtirish orqali funksiyalar grafiklarini chizishimiz mumkin bo'ladi.

Funksiya grafiklarini chizishning yana biro oson va qulay yo'li bu biz yuqorida atab o'tgan TChart komponentasi orqali chizishdir. Bu usul asosan sonli grafiklarni tasvirlash uchun foydalaniladi. Bu usul ishlashga juda qulay va oson yo'l bo'lib hisoblanadi. Chuningdek bu komponenta orqali diagrammalar chizmalar va h.k larni ham chizish imkonyati ham bor.

Xulosa qilib aytadigan bo'lsak Delphi tilida funksiya grafiklarini chizishning birnechta usullari mavjud bo'lib ulardan oson va kam vaqt talab qiluvchi usulni tanlab olish dasturchining o'ziga bog'liqdir.

Foydalanilgan adabiyotlar:

1. Aripov M.M., Yakubov A.X., Sagatov M.V., Irmuhamedova R.M. va boshqalar. Informatika. Axborot texnologiyalari. O'quv qo'llanmasi. 1-qism. Toshkent: 2005, 334 b.
2. Aripov M.M., Yakubov A.X., Sagatov M.V., Irmuhamedova R.M. va boshqalar. Informatika. Axborot texnologiyalari. O'quv qo'llanmasi. 1,2 qism. Toshkent: 2005, 334 b.
3. Nazirov SH.A., Qobulov R.V. va boshqalar. Komp'yuter va ofis jixozlaridan foydalanish.- Toshkent 2007 y.
4. Xolmatov T.X., Taylaqov N.I. "Amaliy matematika, dasturlash va komp'yuterning dasturiy ta'minoti".-T.: Mehnat,2000.
5. Nazirov SH.A., Qobulov R.V. Ob'ektga mo'ljallangan dasturlash.-T 2007 y.
6. Arxangel'skiy A.Ya. Delphi 7, Uchebnoe posobie, Moskva, Izdatel'stvo BINOM, 2004.

Internet saytlari

1. www.ziynet.uz
2. www.kun.uz
3. www.tami.uz