

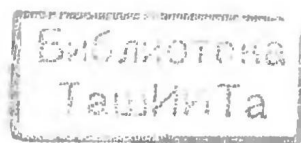
«O'zbekiston temir yo'llari» DATK  
Toshkent temir yo'l muhandislari instituti

# **DASTURLI TA'MINOT VA KOMPYUTERDA MASALALAR YECHISH**

## **3 BORLAND DELPHI VOSITASI**

O'quv qo'llanma

Mazkur o'quv qo'llanma O'zbekiston Respublikasi Oliy va O'rta Maxsus Ta'lim Vazirligi Oliy O'quv Yurtlariaro Ilmiy-Uslubiy Birlashmalar Faoliyatini Muvofiqlashtiruvchi Kengash Prezidiumining qarori buyicha OO'Yurtlari uchun o'quv qo'llanma sifatida tasdiqlangan



Toshkent — 2006

"Dasturli ta'minot va kompyuterda masalalar yechish" o'quv qo'llanmasi uchta qismdan iborat. Birinchi qismda EHM larining rivojlanishi, ularning avlodlari va "Informatika" fanining kelib chiqishi, zamonaviy kompyuterlarning strukturasi to'g'risida ma'lumot berilgan, operatsion tizimlar va servisli dasturlari haqida kerakli axborotlar keltirilgan. Bundan tashqari masalalarni kompyuterda yechish uchun tayyorlash va ularni algoritm lash ishlariga bag'ishlangan.

Ikkinchi qism Object Pascal algoritmik tilida dasturlash ishlariga bag'ishlangan.

Uchinchi qismda Delphi algoritmik til berilgan va kompyuter yordamida ayrim masalalarni sonli usul bilan yechish imkoniyatlari keltirilgan.

Ko'rilgan har bir mavzu yetarli misollar va tekshiruv savollari bilan berilgan.

Mualliflar qo'lyozmani chop etishda foydali maslahatlar va fikr-mulohazalar orqali yordam ko'rsatganlarga minnatdorchilik bildiradilar.

Qo'llanma texnik mutaxassislik bo'yicha ta'lim olayotgan talabalar, aspirantlar va texnik xizmatchilariga mo'ljallangan.

Mazkur o'quv qo'llanma O'zbekiston Respublikasi Oliy va o'rta maxsus ta'lim vazirligi oliy o'quv yurtlaridagi Ilmiy-uslubiy birlashmalar faoliyatini Muvofiqlashtiruvchi kengash prezidiumining qarori bo'yicha OO'Y lari uchun o'quv qo'llanma sifatida tasdiqlangan.

Taqrizchilar: "Kibernetika" institutining yetakchi ilmiy xodimi t.f.d., professor I. Alimov;

Toshkent Davlat madaniyat institutining "Informatika" kafedrası dotsenti, t.f.n. R. Ibragimov;

Toshkent temir yo'l muhandislari institutining "Hisoblash texnikasi" kafedrası dotsenti, t.f.n. T.R. Nurmuhamedov.



## 1. BORLAND DELPHI VOSITASI

Strukturali obyekt bo'yicha oriyentatsiyalangan dasturlashning vositalaridan biri **Delphi** tili hisoblanadi. **Delphi** strukturali oriyentatsiyalangan til (Object Pascal) bilan foydalanadi va u bir tomondan 4GL tillarga xarakterli dasturlashga ta'sirli kuch va oddiylik qo'shadigan bo'lsa, ikkinchi tomondan 3GL tilning natijali tomonini o'ziga olgan. **Delphi** iling'or dasturlash konsepsiyalarini to'la quvvatlaydi, ular qatori meroslikni saqlash, poliformizm xususiyatlar va hodisalar ustidan boshqarish.

**Delphi** — bu bir nechta muhim texnologiyalar kombinatsiyasi:

- yuqori mahsuldorli mashina kodida ishlaydigan kompilyator;
- komponentlari obyekt bo'yicha oriyentatsiyalangan modelni tashkil qiladi;
- dasturli o'xshashliklardan vizual ravishda (demak yuqori tezlikda ham) amaliyotni barpo etish;
- berilganlar bazasini barpo etishda masshtablanadigan vositalar mavjud.

Hozirgi vaqtgacha **Delphi** ning ikki versiyasi ma'lum. Ularning biri - **Delphi Client Server** arxitektura bo'yicha amaliyot ishlarini bajaruvchilarga mo'ljallangan, ikkinchisi esa, **Delphi for Windows** bo'lib, qolgan dastur tuzuvchilarga xizmat qiladi. Biz **Delphi** ning ikkinchi versiyasini ko'rib chiqamiz.

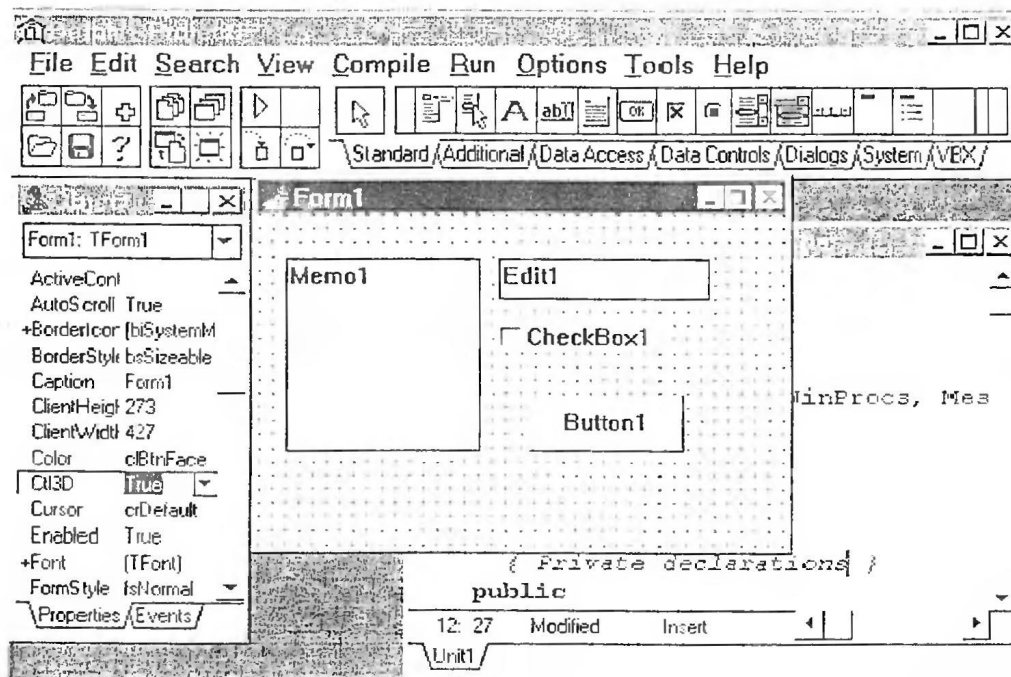
**Delphi for Windows** o'z tarkibiga **Delphi Client Server** ning to'plamli qismlarini qamrab oladi va u yuqori mahsuldorli shaxsiy amaliyotlarni Dbase va **Paradox** tipdagi lokalli SUBD larni barpo etuvchilarga mo'ljallangan hamda u bilan boshqa har bir foydalanuvchi ishlatishi mumkin.

**Delphi** ga quyidagilar kiradi:

- Object Pascal kompilyatori (bu til Borland Pascal 7.0 ning kengaytmasi hisoblanadi);
- Report Smith 2.5 hisobot generatori;
- amaliyotni vizual ravishda qurish muhiti;
- vizualli komponentlar kutubxonasi;
- InHrBase lokalli serveri.

## 1.1. DELPHI ni ISHGA TUSHIRISH

Delphi ni ishga tushirish uchun Windows ning bosh menyusidan "Programmi" buyrug'idan ichki menyuga kirish va u yerda Borland Delphi orqali Delphi ga chiqish kerak (1.1 shakli) so'ngra sichqon klavishasini bosilsa Delphi yuklanadi, shunda monitor ekranida panel paydo bo'ladi.



1.1-shakl.

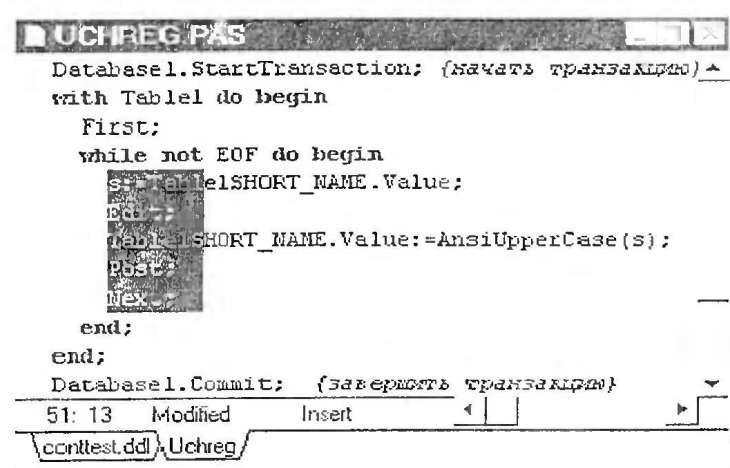
Delphi ishga tushirilganda derazani yuqori qismida gorizontal yo'nalishda komponentlar ikonkalari joylashadi. Kursorni biror ikonkani ostiga keltirilsa sariq rangli to'g'ri to'rtburchakda yordamlovchi axborot chiqadi. Ikonchalar to'plamidan siz kerakligini tanlab olib, amaliyotni qurishga imkon topasiz. Komponentlar qatori tarkibida vizual hamda mantiqiylari bor. Vizual komponentlariga tugmachalar, tahrirlash maydonlari kiradi, jadvallar, hisobotlar — bu mantiqiy komponentlar.

Tushunarli, Delphi da dastur vizual ravishda barpo etiladigani sababli, har bir komponentlar grafikli shakllarga ega, bu esa ularni ishlatilishida qulaylik yaratadi. Ishlayotgan dasturda faqat vizual komponentlar ko'rinadi. Komponentlar har bir qator betida ishlatish funksiyalari bo'yicha guruhlangan bo'ladi. Misol uchun, Windows "common dialogs" tashkil etuvchi komponentlari "Dialogs" deb nomlangan qator betda joylashgan.

**Delphi** ishlovchilarga maksimal qulay muhitni moslashni imkonini beradi. Siz hech qanday qiyinchiliksiz komponentlar to'plami, instrumental chiziqlikchasini o'zgartirishingiz mumkin hamda sintaktik xatolarni rang bilan ajralishini moslashtirishingiz ham mumkin.

Shuni aytib ketish kerakki, **Delphi** da siz o'z komponentlar guruhini tashkil qilib, uni imkoniyatlar qatoriga joylashtirishingiz, kerak bo'lganda ularni o'zgartirib yoki ayrimlarini chiqarib tashlab yuborishingiz ham mumkin.

### *Intellektualli muharriri*

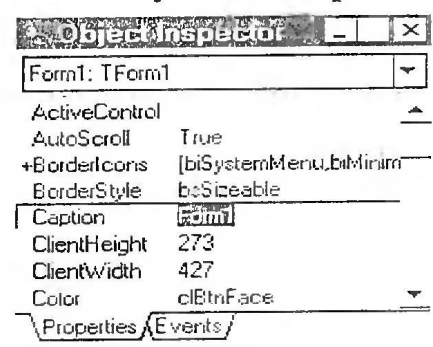


Dasturlarni muharrirlash ishlaridan yozuvlar foydalanish va makroslar ijro etish, matnli bloklar bilan ish olib borish, klavishalar kombinatsiyasini moslashtirish va satrlarni rang bilan ajratish orqali amalga oshiriladi.

### *Grafikli sozlovchi*

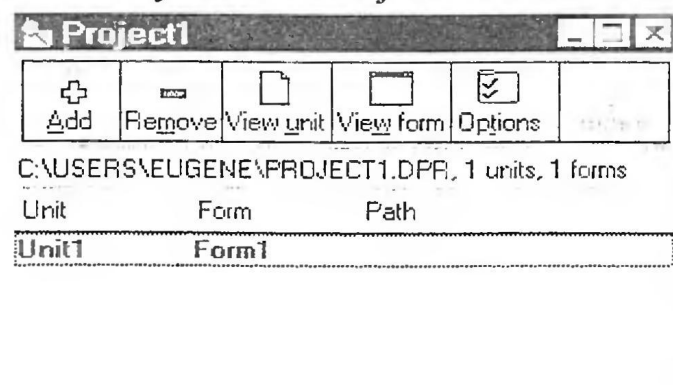
**Delphi** muharririga kiritilgan juda ham kuchli bo'lgan dastur sozlovchisiga ega, u bilan mashina kodidagi dasturning xatolarini topib, ularni tuzatish mumkin. Siz dasturni to'xtatish nuqtasini o'rnatishingiz, o'zgaruvchilarni tekshirish va ularni o'zgartirishingiz, qadamma-qadam ishlatish yo'li orqali dasturni aniq ishlayotgan yo'li to'g'risida ma'lumot ham olishingiz mumkin. Agarda dasturni aniqroq ravishda sozlash kerak bo'lsa, siz **Turbo Debugger** ga imkoniyatli bo'lgan assemblerli instruksiyalarini va protsessor registrnlarni tekshirib ko'rishingiz mumkin.

### *Obyektlar inspektori*



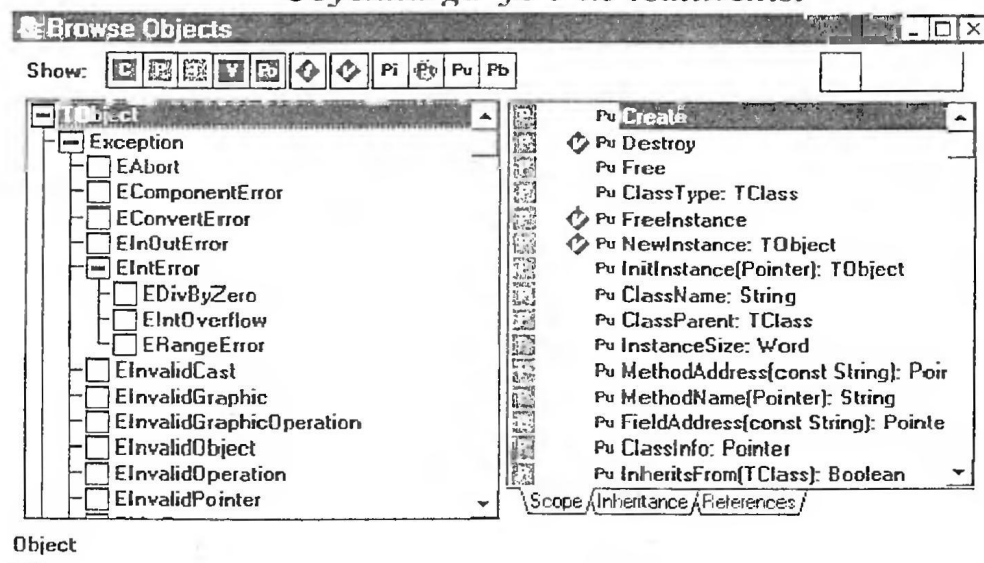
Bu instrument alohida derazani tashkil qiladi, shu derazada dasturni loyihalash paytida obyektlarni xossa va hodisalarining qiymatlarini o'rnatishingiz mumkin.

### Loyihalar menejeri



Bu ishlab chiqiruvchiga tegishli loyihada barcha modullarni ko'rib chiqish imkoniyatini beradi va loyihalar ustidan boshqarish uchun qulay mexanizm bilan ta'minlaydi. Loyihalar menejeri tanlab olingan shakl va boshqalarni faylning nomi, vaqt/sanasini ko'rsatadi. Tegishli nomga sichqon bilan shikillatib, zudlik bilan matn yoki shaklga o'tish mumkin.

### Obyektlarga yo'l ko'rsatuvchisi



Bu imkoniyatli obyektlar kutubxonasini ko'rsatadi va sizni amaliyotingiz kerakligini topadi. U yerda obyektlarni bir-biriga bog'lanishlarini, kutubxonadagi kompilyatsiyalashtirilgan modullarni, mashina kodlarining global nomlarini ko'rish mumkin.

### Menyu dizayneri

Barpo etilgan shablonlarni saqlovchi menyu tashkil qilish mumkin va keyinchalik u bilan o'z amaliyotingizda foydalanishingiz mumkin bo'ladi.

### ***Ekspertlar***

Bu instrumentalli dasturlar to'plami bo'lib, sizning amaliyotingizni loyihalash va sozlash ishlarini yengillashtiradi. Bundan tashqari mustaqil ishlab chiqilgan ekspertlarni kiritish imkoniyati bor. Bunday imkoniyat yordamida uchinchi firmalar maxsus **Delphi** uchun ishlab chiqilgan instrumentlar bilan **Delphi CASE** kengaytirishi mumkin.

#### ***Ekspertlar o'z tarkibiga quyidagilarni olgan :***

- Berilgan bazasi bilan ishlovchi shakllar eksperti
- Amaliyotlarning uslubi va shablonlari eksperti
- Shakllarning shablonlari eksperti

**RAD Pack** tarkibiga resurslarni qaytadan o'zgartiruvchi ekspert kiradi, u **Delphi** shaklida **Borland Pascal 7.0** da tayyorlangan.

#### ***Interaktiv ravishda o'rgatuvchi tizim***

U **Delphi** ni to'laroq o'zlashtirishga yordam beradi. U, nafaqat savollarga javob aytib beruvchi tizim, ishlovchini o'z muhitida **Delphi** ning imkoniyatlarini ham ko'rsatib turadi.

*Berilganlar bazasiga kira olish komponentlari va berilganlarni vizuallashtirish*

Obyektlar kutubxonasi vizual komponentlar to'plamiga ega, ular yordamida **SUBD** tizimini "klient-server" arxitekturasi bilan bog'lanishni ancha osonlashtiradi. Obyektlar **Borland Database Engine** pastki sathini o'ziga olishgan.

Maxsus komponentlar to'plami ko'zda tutilgan, ularning biri berilganlarga kira olishga javobgar bo'lsa, ikkinchisi esa berilganlarni ko'rsatishni ta'minlaydi. Bundan tashqari ular yordamida berilganlar bilan bog'lanish, berilganlar ichidan kerakligini to'plash, ularni nusxalash va h.k. ishlarni bajarish mumkin.

Berilganlarni vizuallashtirish komponenti jadval, maydon, ro'yxat ko'rinishda ko'rsatish imkoniyati bor.

#### ***Berilganlar bazasining yondoshini ishlab chiqish***



**Delphi** vizualli komponentlar kutubxonasi yordamida ishonchli yondoshchuvchini tezda barpo etish mumkin, ularni "klient-server" arxitekturasining yondoshuvchilari-gacha kengaytirish mumkin. Boshqacha qilib aytiladigan bo'lsa, siz **InterBase** lokal

serveri ishlaydigan yondoshuvni barpo etasiz, keyin esa, SQL-LINKS orqali sizdan uzoqlashgan SQL serveri bilan bog'lanishingiz mumkin.

### *Obyektli vizual komponentlar kutubxonasi*

**Delphi** da ishlab chiqarishda (hamda Delphi tomonidan ham) foydalanadigan komponentlar yondoshlarini ishlab chiqarish muhitiga kiritilgan va u o'z tarkibiga obyektlar tiplaridan tashkil qilingan to'plam bo'lib, amaliyotlarni qurishda asos sifatida olinadi.

yoki o'zakni **Visual Component Library (VCL)** deb ataladi. VCLda standartli boshqaruvchi elementlar, tahrirlash satrlari, boshqarishni statikli elementlari, ro'yxatlar bilan ishlanadigan tahrirlash satrlari, obyektlar ro'yxati deganlar mavjud.

Bundan tashqari ularda, avval faqat uchinchi firmalar kutubxonalaridan ega bo'lish imkoniyatlari bor: boshqaruvning jadvalli elementlari, xatcho'plar, ko'p betli yon daftarchasi.

VCL maxsus obyekt bilan ta'minlangan, u Windowsni grafikli qurilmalarining interfeysini tashkil qiladi va ular orqali ishlovchi hech qanday qiyinchiliksiz har xil detallarni chizishi mumkin.

Delphini asosiy xususiyati shundan iboratki, u amaliyotlarni tuzishda nafaqat vizual komponentlardan foydalanadi, u yangi komponentlar barpo etishi ham mumkin. Bu imkoniyat ishlovchiga boshqa muhitga o'tmasdan, aksi, mavjudli muhitni yangi instrumentlar bilan ta'minlashi mumkin. Bundan tashqari, aytib o'tmaslik sharti bo'yicha, Delphida mavjud komponentlarni yaxshilash yoki umuman ularni almashtirish mumkin. Shuni aytish joiziki, oddiy vizualli ishlash muhitlariga tegishli cheklanishlar Delphida yo'q. Delphi o'zi ham Delphi yordamida yozilgan cheklanishlar yo'qligiga ishonch bo'lishi mumkin.

Obyektlar guruhi iyerarxialik ravishda tuzilgan va uning tarkibiga abstraktli, oraliqdagi hamda tayyor komponentlar kiradi.

### *Delphini dasturlash tili*

**Delphi** ni dasturlash tili **Borland Object Pascal** tiliga asoslanadi. Bundan tashqari, unda Windows va assemblarning pastki saviyali xususiyatlar ham bor.

Shakllar, modullar va "Two-Way Tools" ni ishlab chiqish usuli

**Shakllar** – bu obyektlar bo'lib, uning ichiga boshqa obyektlar joylashtiriladi va foydalanuvchi interfeysini barpo etish uchun



ishlatiladi. Modul kodlardan iborat bo'lib, sizning amaliyotingizni amalga oshiradi.

Shakllar to'g'risidagi axborot ikki tipdagi fayllarda saqlanadi — .dfm va .pas, shu bilan birga shu birinchi tipdagi fayl — ikkilik bo'lib, shaklning xususiyati va qiyofasini saqlaydi, ikkinchi tipdagi esa hodisalar ishlovchilari va komponentlar harakatining ishlashini amalga oshiradi. Ikkala fayllar avtomatik ravishda Delphi da sinxronlashadi, shu sababli agar sizni loyihangizga yangi shakl qo'shilsa, u bilan bog'langan .pas fayli avtomatik ravishda barpo etiladi va uning nomi loyihaga qo'shib qo'yiladi.

Bunday holda sinxronlashish Delphini two-way instrumentiga aylantiradi hamda vizual va kod ravishda taqdim etilishni bir-biriga moslashtiradi.

*Masalan*, faraz qilaylik, siz shakl harakatining ta'rifini qo'shdingiz, tugmachani bosish orqali derazada xabarnomani chiqarish maqsadida. Bunday ta'rif paydo bo'ladi, agarda sichqon bilan shakldagi Button obyektiga nisbatan ikki marotaba shiqillatilsa, yoki obyektlar Inspektorida Events betidagi **OnClick** qatorini ikki marotaba sichqon bilan shiqillatsak. Har qanday holda ham Delphi protsedura barpo etadi yoki usulni sarlavhasini chiqazadi, qayerga siz kod qo'shib qo'ysangiz bo'ladi.

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

Bunday kodni barpo etib, **Delphi** avtomatik ravishda TForm1 obyektning deklaratsiyasini shakllantiradi, unda hodisaga ishlov beruvchi **BittonClick** protseduraga ega.

```
Tform1 = class (Tform)  
Button : Tbutton ;  
procedure Button1Click (Sender :TObject) ;  
private  
(Private declarations)  
public  
(Public declarations)  
end;
```

Albatta, siz bu kodni olganingizdan keyin, bimalol qarorga kelishingiz mumkin, avtomatik ravishda barpo etilgan nomlar sizni qoniqtirmasa ularni almashtirasiz. Masalan, Button1 ni Warning ga. Buni obyektlar Inspektori yordamida Button1 uchun Name xususiyatini o'zgartirish orqali amalga oshiriladi. Enterni bosishingiz bilan, Delphi avtomatik ravishda tegishli sinxronlashtirishni kodda bajaradi. Tform obykti kod ravishda mavjud bo'lgani sababli, siz erkin ravishda ixtiyoriy boshqa maydon, protsedura, funksiya yoki object definitionlarni qo'shib qo'yishingiz mumkin. Misol uchun, siz o'z protsedurangizni vizual ravishda bajarmasdan kod ravishda yozib qo'yishingiz mumkin.

Keyingi misol bu qanday qilib bajarilishini ko'rsatadi.

Ishlov beruvchisi Tobject tipdagi argumentni qabul qiladi, u bizga, bu hodisani kim initsiallashtirganini aniqlashga imkon beradi. Bu foydali qachonki, bir nechta tugmachalar ishlov uchun umumiy protsedurani chaqiraganda foydali bo'ladi.

```
Tform1 = class(Tform)
Warning: Tbuton;
procedure WarningClick (Sender :Tobject);
procedure NewHandler(Sender :Tobject) ;
private
(Private declarations)
public
(Public declarations)
end;
```

Bu yerda sinxronlashtirishning ikkinchi tomoniga duch kelindi. Vizualli muhit bunday holda fahmlaydiki yangi protsedura obyektga qo'shilgani va ularning tegishli nomlari obyektlar Inspektorida paydo bo'ladi.

### *Yangi obyektlarni qo'shish*

**Delphi** — bu eng avval komponentlarni ishlatishga asoslangan ishlab chiqarish muhiti. Siz umuman yangi komponentlarni komponentlar qatoriga qo'shib qo'yishingiz mumkin. Siz komponentlarni Delphi ichida barpo etishingiz mumkin yoki VBX yoki OLE 2.0 boshqaruvchi elementi sifatida barpo etilgan komponentlarni kiritishingiz, yoki C hamda C++ da



dll ko'rinishda yozilgan komponentlar bilan foydalanishingiz mumkin.

Yangi komponentlarni kiritish ketma-ketligi uch qadamdan iborat:

- mavjud tipdagi komponentlarning xosiyatlarini saqlab qolish
- yangi maydon, xossa va usullarni aniqlash
- komponentlarni ro'yxatga olish.

Bular hammasi Install Components menyusi orqali bajariladi.

### ***Boshqaruvchi VBX elementlarini qo'shish***

Delphi VBX obyektli kengaytmani generatsiyalashtirish natijasida komponent sifatiga o'tadi.

Masalan, agar siz Borland kompaniyani Vizual Solutions dan SaxComm VBX installizatsiya qilsangiz, Delphida avtomatik ravishda Tcomm tipdagi obyekt generatsiyalashadi, u standartli TVBXControlni xosiyatlarni saqlab qoladi. Siz komponentlarni installizatsiya qilayotganingizda, Delphi ularni kompilyatsiya qilib komponentlar kutubxonasiga tadi.

Vakillik orqali vaziyatni dasturlash juda ham oddiy.

Biror obyekt boshqa obyektga ayrim voqealarga javob berishni vakillik beradigan bo'lsa, buni vakillik deyiladi.

Bunday model ayrim holatlarda dasturlashni juda ham soddalashtiradi. Masalan, Windows controls uchun qo'shimcha harakat berish o'rniga, siz voqeani ishlovchi protsedurani bog'lab qo'ysangiz bo'ladi va u foydalanuvchi tomonidan har bir sichqon shiqillatish yoki klavishani bosilganda avtomatik ravishda chaqiriladi.

### ***Sinflarga dalil keltirish***

Sinflarga dalil keltirish qo'shimcha eguvchanlik daraja orttiradi, qachonki, siz dinamik ravishda obyektlarni barpo etsangiz, ularni kodlari faqat bajarilish paytida ma'lum bo'ladigan bo'lsa. Misol uchun, foydalanuvchi o'z hujjatini har xil tipdagi Obyektlardan barpo etiladigan bo'lsa, u menyu yoki imkoniyat qatoridan ularni terib boshlaydi. Haqiqatda, bu texnologiya Delphi ni qurishda ham ishlatilgan.

### ***Maxsus vaziyatlarga ishlov berish***

Jiddiy amaliyotlar ishonchli ravishda maxsus vaziyatlarning ishlovini bajarishi kerak, agarda mumkin bo'lsa, dasturni ishga solish yoki buning ilojisi bo'lmasa, uni muntazam ravishda tugatishlorim. Maxsus vaziyatlarni ishlovini beruvchi kodlarni yozish har doim oddiy masala bo'lmagan va u qo'shimcha xatolarning paydo bo'lishiga manba bo'lgan.

Delphida bu C++ tarzda uyushtirilgan. Maxsus vaziyatlar obyektlar sifatida keltirilgan, ular tegishli xato to'g'risida xususiyatli axborot ega (xatoning tipi va uning joyi). Foydalanuvchi xatolarni ishlovini aytib o'tmaslik holdagiga yuklashi yoki o'zining ishlov beruvchisini yozishi mumkin.

Maxsus vaziyatlar ishlovi exception-handling blocks (bundan tashqari uni protected blocks ham deb ataladi) ko'rinishda ham amalga oshiriladi va ularga try va end kalitli so'zlari o'rnatiladi. Bunday tipdagi bloklarning ikkitasi mavjud: try...except va *finally*.

Umumiy konstruksiya taxminan shunday ko'rinishda bo'ladi :

```
try
(bajariladigan operatorlar)
except
    on exception1 do statement1; (vaziyatga ta'sir)
    on exception2 do statement2;
else
    (aytib o'tmaslik sharti bo'yicha operatorlar)
end;
```

try...finally konstruksiyasi har qanday vaziyat paydo bo'lmasin ishlab chiqaruvchi to'la ishonchda bo'ladiki, maxsus vaziyatning ishlovi boshlanmasdan albatta biror kod bajariladi (masalan, resurslarni bo'shatish).

```
try
(bajariladigan operatorlar)
finally
(shartsiz ravishda bajariladigan operatorlar)
end;
```

## 1.2 DELPHINI DASTURLASH MUHITI

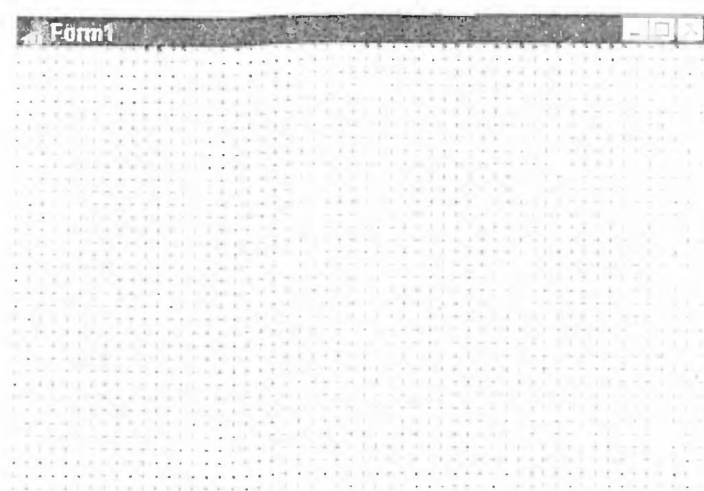
### 1.2.1. DASTURLASH MUHITINING ASOSIY QISMLARI

Quyida Delphi ning tarkibidagi asosiy qismlari keltirilgan:

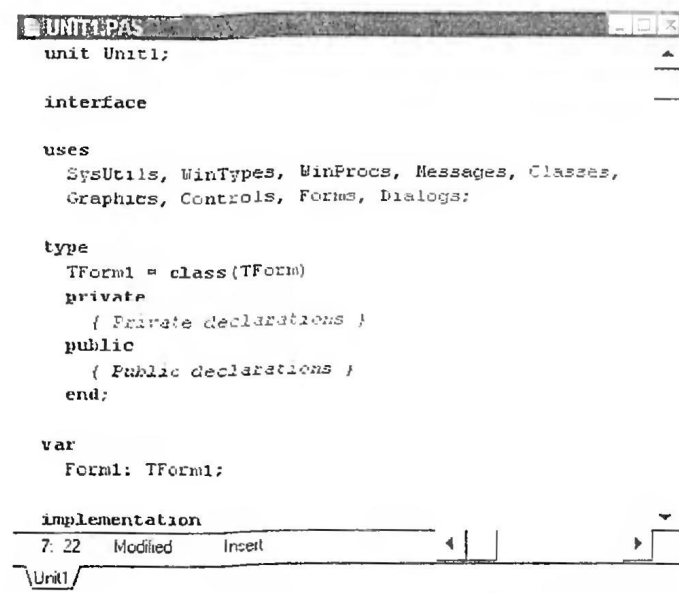
1. Shakllar dizayneri (Form Designer);
2. Dastlabki matn Muharrirning derazasi (Editor Window);
3. Komponentlar qatori (Component palette);
4. Obyektlar Inspektori (Object Inspector);
5. Ma'lumotnoma to'plami (On-Line help).

Bundan tashqari Delphining tarkibida boshqa muhim qismlar ham bor: instrumentlar jadvali, tizimli menyu hamda sizga dasturni va dasturlash muhitini sozlash uchun kerak bo'ladigan imkoniyatlar mavjud.

Delphida dastur tuzuvchilari ko'p vaqtini Shakllar Dizayneridan Dastlabki matn Muharririga (qisqacha rus tilida Redaktor deb nomlanadi) o'tishga sarflaydilar. Ishni boshlashdan oldin, siz ikki muhim elementni tanib olishingiz mumkinligiga ishonch hosil qilishingiz kerak. Shakllar dizayneri shakl 10.2 da, Redaktor derazasi esa shakl 10.3da ko'rsatilgan.



1.2-shakl.



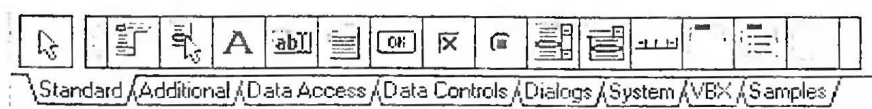
1.3-shakl

Shakllar Dizayneri dastlab bitta bo'sh derazadan iborat, siz uni komponentlar qatoridan olinadigan har turli obyektlar bilan to'ldirasiz.

Shakllar Dizayneri muhimligiga qaramasdan, dasturchini asosiy vaqtini o'tkazadigan joy Redaktor hisoblanadi. Mantiq dasturning harakat qildiruvchi kuchi hisoblanadi va Redaktor aynan o'sha joy, qayerda siz o'z dasturingizni yozishingiz mumkin.

Komponentlar qatori (shakl 1.4 ga qarang) kerakli obyektlarni Shakllar Dizaynerida joylashtirish uchun imkon beradi. Komponentlar qatorini ishlatish uchun sichqon bilan oddiy ravishda bir marta shiqillatish kerak va keyin ikkinchi marta – Shakllar Dizaynerida. Tanlab olgan obyektningiz loyihalanayotgan derazada paydo bo'ladi va uni sichqon yordamida harakatlantirish mumkin.

Komponentlar qatori obyektlar betma-bet guruhlanishidan foydalanadi. Qatorni pastki qismida xatcho'plar to'plami joylashgan – Standart, Additional, Dialogs va h.k. Agar siz biror xatcho'pni sichqon bilan shiqillatsangiz, siz Komponentlar qatorini keyingi betiga o'tasiz. Betma-bet bo'lish prinsipi Delphi dasturlash muhitida keng foydalanadi va uni o'z dasturingizda oson ravishda ishlatishingiz mumkin (Additional betlarida xatcho'plarini yuqoridan pastga qarab betlarni tashkil qiluvchi komponentlar bor).



1.4-shakl. Komponentlar qatori – bu yerdan siz kerakli obyektlarni olib turib, ularni o'z shaklingizga joylashtiringiz mumkin.

Faraz qilaylik, Tedit komponentini shaklga joylashtirmoqchisiz, siz uni bir joydan ikkinchi joyga siljitishingiz mumkin. Bundan tashqari, obyekt atrofdagi chegarali chiziqlarni uni o'lchamlarini o'zgartirish uchun ishlatishingiz ham mumkin. Boshqa komponentlarning ko'pini shu holda harakatlantirish mumkin. Biroq, dastur ishlayotgan vaqtida ko'rinmaydigan komponentlar (Tmenu yoki TdataBase kabi) o'z shakllarini o'zgartirmaydi.

Shakllar Dizayneridan chap tomonda Obyektlar Inspektorini ko'rishingiz mumkin (1.5-shakl). Shunga e'tibor beringki, Obyektlar Inspektoridagi axborot shakldan tanlab olingan obyektga qarab o'zgaradi. Eng muhimi, har bir komponent haqiqiy obyekt hisoblanadi va siz uning ko'rinishini va harakatlanishini Obyektlar Inspektori yordamida o'zgartirishingiz

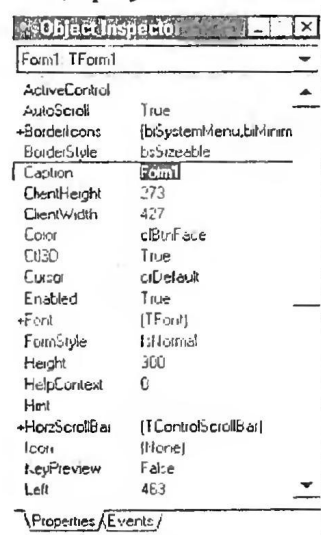
mumkin.

Obyektlar Inspektori ikki betdan iborat, ularni har birini berilgan komponentning harakatini aniqlash uchun foydalanish mumkin. Birinchi bet – bu xususiyatlar ro'yxati, ikkinchisi – hodisalar ro'yxati. Agar komponentlarni aniqlash bo'yicha biror o'zgartirish kiritmoqchi bo'lsangiz, u holda siz buni odatda Obyektlar Inspektorida bajarasiz. Misol uchun, Caption, Left, Top, Height va Width xususiyatlarini o'zgartirib, siz Tlabel komponentning nomi va o'lchamini o'zgartirishingiz mumkin.

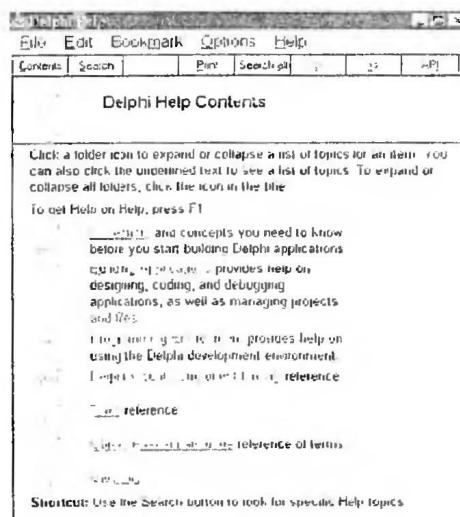
Obyektlar Inspektori ostidagi xatcho'plarni xususiyatlar va hodisalar betlarini varaqlash uchun ishlatishingiz mumkin.

Hodisalar beti Redaktor bilan bog'langan; agar siz sichqon bilan biror punktning o'ng tomoniga nisbatan shiqillatsangiz, u holda bunday hodisaga tegishli kod avtomatik ravishda Redaktor yoziladi, Redaktorni o'zi darhol fokus oladi va sizda darrov berilgan hodisa ishlovchisining kodini qo'shib qo'yish imkoniyatingiz bo'ladi. Delphini dasturlash muhitining bunday nuqtayi nazaridan keyinchalik ham muhokama qilinadi.

Delphi muhitining oxirgi muhim qismi – Ma'lumotnoma to'plami (on-line help). Bu instrumentga kira olish uchun tizimli menyuning Help va keyin Contents punktlarini shundaygina tanlab olish kerak. Ekranda, 1.6-shaklda ko'rsatilgan, Ma'lumotnoma to'plami (rus tilida Spravochnik) paydo bo'ladi.



1.5-shakl.



1.6-shakl. Ma'lumotnoma to'plami – har 1.6-shakl. qanday axborotni tezlik bilan izlovchi.

Ma'lumotnoma to'plami kontekstno-qaramli hisoblanadi; F1 klavishi bosilganda, tegishli joriy hodisaga ma'lumotnoma chiqadi. Agar Delphi muhitida ishlash paytida noaniqlik paydo bo'lsa, F1 ni bosing va kerakli axborot ekranda paydo bo'ladi.

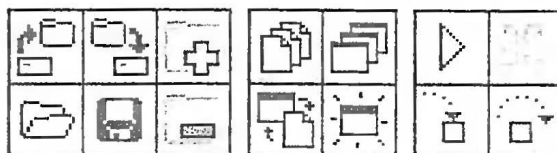
### ***Qo'shimchali elementlar***

Ular dasturlash muhiti uchun qo'shimchali elementlar hisoblanadi:

- Menyu (Menu System)
- Tezlik ravishda kira olish uchun tugmachalar paneli
- Suratchalar Muharriri (Image Editor)

Menyu Delphi muhitiga tezlik va eguvchanlik interfeysni taqdim etadi, chunki u "qaynoq klavishalar" to'plami bilan boshqaradi. Uning qulayligi yana shundan iboratki, u yerda ikonka yoki piktogrammalar o'rniga aniqroq va tushunarliroq bo'lgan so'z yoki qisqacha jumlar ishlatiladi. Menyudan keng doiradagi masalalarni yechishda foydalanish mumkin, aniqroq qilib aytiladigan bo'lsa, u fayllarni ochish va yopish, sozlovchi ustidan boshqarish yoki dasturlash muhitini sozlash singari umumiy bo'lgan masalalarga qo'llaniladi.

SpeedBar Komponentlar qatoridan chap tomonidan, to'g'ridan-to'g'ri menyu ostida joylashgan (shakl 1.7). SpeedBar menyu orqali qilinadiganlardan ko'pini bajaradi. Agar, sichqon SpeedBarning biror ikonkasida ushlab qolinsa, u holda siz mazkur ikonkani vazifasini tushinturuvchi ma'lumotnoma chiqishini ko'rasiz.



1.7-shakl.

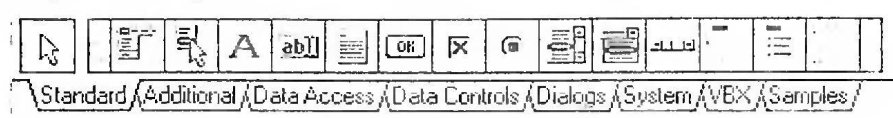
## **1.2.2 STANDARTLI KOMPONENTLAR**

Delphi dasturlash muhiti bilan tanishtirishni davom ettirish uchun Komponentlar qatorini birinchi betining tarkibi to'g'risida aytib o'tish kerak bo'ladi.

Komponentlar qatorining birinchi betida 14 ta obyekt joylashtirilgan (1.8-shakl.). Ular foydalanuvchiga muhim deb topilgan. Uzoq vaqt ichida tugmacha, ro'yxat, kiritish derazasi va boshqalarsiz kifoyalanadigan inson topilmasa kerak. Bu obyektlar muhimligi



Windows uchun sichqon yoki deraza singari desak xato bo'lmaydi. Har bir betdagi komponentlar to'plami va ularning tartibi konfiguratsionli hisoblanadi. Mavjud komponentlarga siz ularning soni va tartibini o'zgartirib yangilarini qo'shisiz mumkin.



1.8-shakl. Komponentlar, qatorning birinchi be yangilarini qo'shisiz mumkin, ida joylashgan.

Delphining standartli komponentlari quyida ularning ishlatilishi to'g'risidagi izohlar bilan keltirilgan. Ular bilan tanishish paytida oldingizda kompyuter bo'lsa foydali bo'ladi, chunki komponent qanday ishlashini va u bilan qanday harakatlanishni kuzatib turishingiz kerak.

 Tmain Menu dastur ichiga asosiy menyuni joylashtirishga imkon beradi. TmainMenuni shakl ustiga joylashtirilsa, oddiy ikonka singari bo'lib qoladi. Bunday tipdagi ikonkalar "ko'rinmas komponentlar" deb ataladi, chunki ular dastur ishlayotgan vaqtida ko'rinmas holda bo'ladi. Menyuni barpo etish uch qadamli: (1) TmainMenuni shakl joylashtirish, (2) Menyu Dizaynerini Obyektlar Inspektorida Items xususiyati orqali chaqirish, (3) Menyu Dizaynerida punktlarni belgilash.

 TpopupMenu qalqib chiqadigan menyuni barpo etishga imkon beradi. Bu menyu sichqon o'ng klavishasi bosilganda paydo bo'ladi.

 TLabel matnni ekranda aks ettirishga xizmat qiladi. Agar Obyektlar Inspektoridagi Font xususiyatiga sichqon bilan ikki marotaba shiqillatsangiz, siz shrift va rang belgilarni o'zgartirishingiz mumkin. Bir qator kodni yozib turib, dastur ishlayotgan paytida ham uni oddiygina bajarishingiz mumkin.

 TEdit – Windowsni kiritish uchun standartli boshqaruvchi elementi. U dastur ishlayotgan vaqtida kalta matnning fragmentini kiritib, uni ekranda aks ettirish mumkin.

 Tmemo – Teditning boshqacha xili. Katta matnlar bilan ishlashda nazarda tutmoq lozim. Tmemo so'zlarni ko'chirishi mumkin, matnning fragmentini Clibborard da saqlash va ularni tiklash va boshqa muharrirni vazifasiga kiruvchi funksiyalar.



Tmemoni cheklanishi bor. Matnning hajmi 32 Kb dan oshmasligi kerak, bu 10–20 betni tashkil qiladi. (VBX va Delphining "tug'ishgan" komponentlari mavjud, bu yerda bu cheklanish olib tashlangan).

📖 Tbutton dastur ishlab turgan paytida tugmachani bosish orqali biror amalni bajarish imkonini beradi. Delphida hammasi oddiy ravishda qilinadi. Tbuttonni shakl ustiga joylashtirib, sichqonni ikki marotaba shiqillatib hodisa ishlovchisini tayyorlanadi. Keyin ishlovchiga kod to'lg'aziladi (qo'lda yoziladigan qismi chizib ko'rsatilgan):

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

```
MessageDlg(' Are you there?', mtconfirmation, mbYesNoCancel,0) ;
```

```
End;
```

☹ TcheckBox matn satrini yoningizda kichkina derazachada aks ettiradi. Derazachada belgi qo'yib qo'yish mumkin, qaysi biri biror narsa tanlangan degan ma'noni bildiradi. Masalan, agar kompilyatorni sozlovchi dialogli derazachani ko'rsak (Compiler betidagi Options|Project menyusi), u holda u asosan "CheckBox" lardan iboratligini ko'ramiz.

☹ TradioButton bir nechta opsiyalardan birini tanlab olish imkoniyatini beradi. Agar siz yanada Linker Options betidagi Options|Project menyusini ochsangiz, ko'rasizki, Map file va Link buffer file seksiyalari RadioButton to'plamlaridan iborat.

☹ TlistBox aylantirish orqali ko'rsatiladigan ro'yxat uchun kerak. Misol sifatida ListBox, Windows muhitida esa – File|Open menyusidagi fayllar ro'yxatidan kerakligini tanlab olish. Fayllarning nomi yoki direktoriyasi ListBox da joylashgan bo'ladi.

☹ TcomboBox ko'p jihatidan ListBoxni eslatadi, faqat farqi shundan iboratki, ListBox yuqorisidagi kichkina kiritish maydonida axborotni kiritish imkonini beradi. ComboBox ning bir nechta tiplari bor, ularning ichida pastga qarab ochiladigani mashhur (drop-down combobox), uni faylni ko'rish dialogli derazachaning pastki qismida ko'rish mumkin.

☹ Tscrollbar – aylantirish tizimi, tahrirlash obyektlarida avtomatik ravishda paydo bo'ladi, ListBox larda matnni ko'rish uchun uni aylantirish kerak bo'lganda.

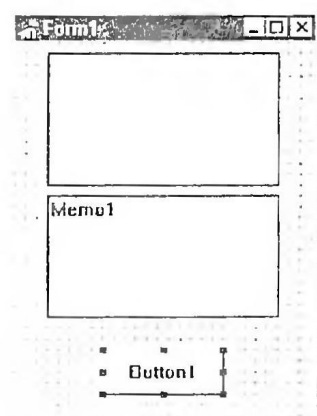
☉ TgroupBox vizualli maqsadlar uchun ishlatiladi va shakllarda komponentlar bo'yicha siljishning tartibini Windowsga ko'rsatadi (TAB klavishi bosilganda).

☉ Tpanel – TgroupBox ga o'xshash bo'lgan boshqaruvchi element, undan dekorativ maqsadda foydalaniladi. Tpanel ni ishlatish uchun, uni oddiy ravishda shaklga joylashtiriladi va undan keyin boshqa komponentlar qo'yiladi. Endi Tpanel siljiganda, u bilan birgalikda shu komponentlar ham siljiydi. Bundan tashqari Tpanelni instrumentlar chizg'ichini va statusning derazasini barpo etishda ishlatiladi.

☉ TScrollBar – bu shakldagi joyni taqdim etadi, siz uni vertikal va gorizontal yo'nalishlarda o'zgartirishingiz mumkin. Bu imkoniyatni o'chirsak, shakl o'zicha shunga o'xshab ishlayveradi, biroq, ayrim paytda shaklning faqat bir qismini aylantirish kerak bo'lsa, shunday vaziyatda TScrollBar ishlatiladi.

Bu hammasi Komponentlar qatorining birinchi betidagi obyektlar to'la ro'yxati. Agar sizga qo'shimcha axborot kerak bo'lsa, uni F1 klavishasini bosib, paydo bo'ladigan Ma'lumotnoma to'plamidan olish mumkin.

### 1.2.3. OBYEKTLAR INSPEKTORI TO'G'RISIDA BATAFSILROQ



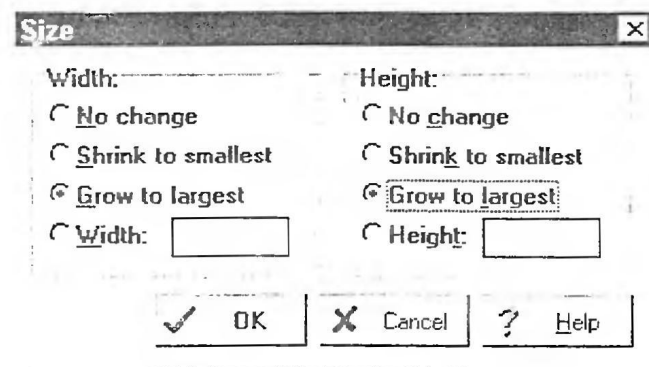
1.9-shakl.

Obyektlar Inspektori (Object Inspector) ni oldinroq qisqacha ko'rib chiqdik. Endi shu muhim elementni chuqurroq ko'rib chiqsak. Obyektlar Inspektori to'g'risida asosiy tushunchamiz shundan iboratki, u shaklga tanlangan ixtiyoriy obyektning xarakteristikasini o'zgartirish uchun ishlatiladi, Undan tashqari, shaklni o'z xususiyatini ham o'zgartiradi. Obyektlar Inspektorini o'rganishning eng yaxshi yo'li – u bilan ishlab ko'rish. Boshlash uchun yangi loyiha oching, menyuning File|New Project punktini ochib, shaklda, 1.9-shakl da ko'rsatilgandek, Tmemo Tbutton va TlistBox obyektlarni qo'ying.

Avval, Ctl3D xususiyati misolida (aytib o'tmaslik prinsipi bo'yicha u ishga solingan bo'ladi) xususiyatlar bilan ishlashni ko'rib chiqamiz. Shaklni tanlang, unga sichqon bilan shiqillatib, Obyektlar Inspektoriga

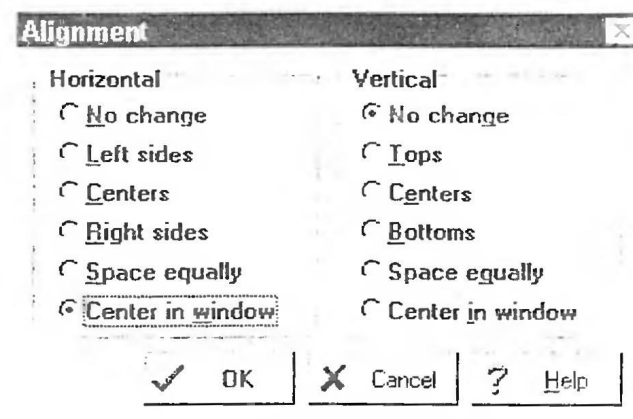
o'ting va ikki marotaba sichqon bilan shiqillatib bir necha marta Ctl3D xususiyatining qiymatini o'zgartirib ko'ring. Shaklning Ctl3D xususiyatini o'zgarishi avtomatik ravishda shaklga joylashgan bog'lanuvchi derazachalarida Ctl3D ning xususiyati ham o'zgaradi.

Shaklga qayting va Ctl3D ning qiymatini True ga qo'ying. Endi "Shift" klavishasi bosing va sichqon bilan Tmemo, keyin esa, TlistBox ga shiqillating. Endi ikkala obyektlar ham chekkalarida kichik kvadratlarga ega bo'ladilar, ular obyektlar tanlab bo'linganligini ko'rsatadi.



1.10-shakl.

Ikkita yoki undan ko'proq obyektlarni bir vaqtda tanlab oling, siz ular ustida bir qator operatsiyalar bajarishingiz mumkin. Masalan, shaklda harakatlantirish. Keyin menyuning Edit|Size punktini olishga urinib ko'ring va Grow to Largest da ikkala maydonga eni (Width) va balandligi (Height), 1.10 shaklda ko'rsatilgandek, o'rnating. Endi ikkala obyekt bir xil o'lchamli bo'lib qoldi. Keyin menyudan Edit|Align punktini tanlab oling va gorizontal bo'yicha tenglashish uchun Centerning qiymatini qo'ying (1.11-shaklga qarang).

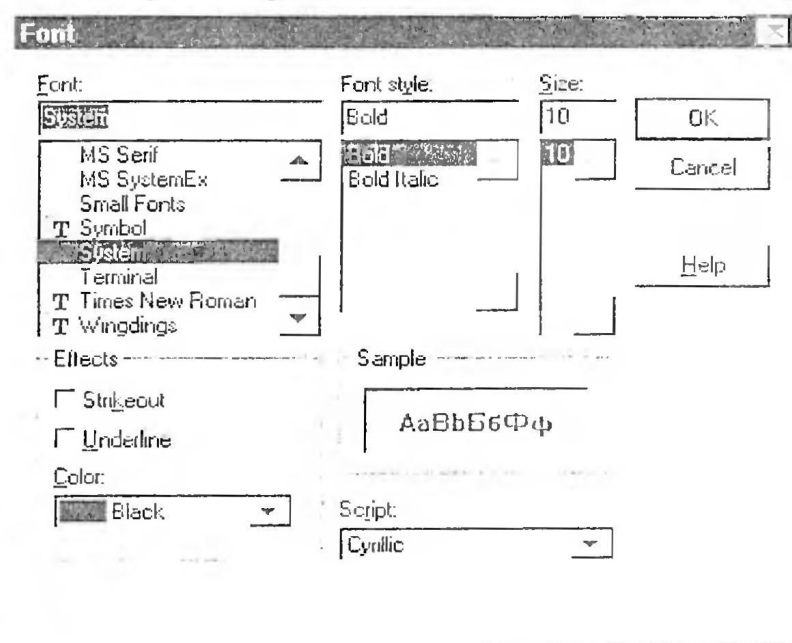


1.11-shakl.

Siz bir yo'la ikkita komponent olganingiz sababli, Obyektlar Inspektori o'zgaradi — u faqat Obyektlarga umumiy bo'lgan maydonlarni ko'rsatadi. Bu, demak siz tomoningizdan bajarilgan o'zgarishlar bir obyektga emas, barcha tanlangan Obyektlarga ta'sir ko'rsatadi.

Obyektlar xususiyatining o'zgarishini Color xususiyat misolida ko'rib chiqamiz. Obyektlar Inspektorida uning qiymatini o'zgartirishning uch xil usuli bor. Birinchisi, oddiy ravishda rangni (clRed) yoki rang raqamini yozib ko'rsatish kerak. Ikkinchi yo'l — o'ng tomondagi kichkina strelkani bosib, ranglar ro'yxatidan kerakligini tanlab olish. Uchinchi yo'l — sichqon bilan Color xususiyatini kiritish maydonida ikki marotaba shiqillatilsa, rang tanlash dialogi paydo bo'ladi.

Font xususiyati Color xususiyati tarzida ishlaydi. Buni ko'rish uchun, avval Tmemo obyekt uchun Font xususiyatini tanlang va sichqon bilan kiritish maydonida ikki marotaba shiqillating. Shriftlarni sozlash dialogi 1.12-shaklda ko'rsatilgandek paydo bo'ladi. Tanlang, masalan, New Times Roman shriftini va qandaydir juda katta, masalan 72, o'lcham o'rnatib. Keyin dialogli derazachani paski o'ng burchagida ComboBox yordamida fonning rangini o'zgartiring. Siz OK tugmachasini bossangiz, Tmemo obyektidagi matn ko'rinishi tubdan o'zgaradi.



1.12-shakl. Shriftlarning tanlash dialogi sizga shriftning tipi, o'lchami va rangini o'rnatish imkonini beradi.

Obyektlar Inspektori bo'yicha qisqacha sayohatimizning oxirida ListBox obyektning Items xususiyatiga sichqon bilan ikki marotaba shiqillating. Shunda dialog paydo bo'ladi, u yerda ListBoxda aks ettirish uchun satr kiritishingiz mumkin. Har bir satrda bittadan bir necha so'zlarni terib chiqing va OK tugmachasini bosing. Matn ListBox da aks ettiriladi.

#### **1.2.4. DASTURLARNI SAQLAB QOLISH**

Siz birqancha harakat qilib dasturni tayyorladingiz, endi uni saqlab qolmoqchisiz. shundagina siz keyinchalik u bilan yana ishlash uchun dasturni yuklaysiz.

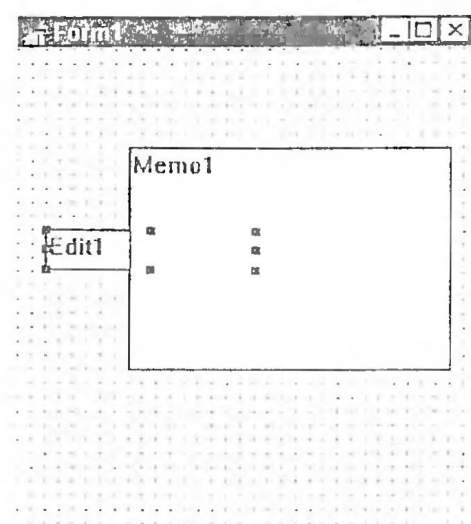
Birinchi qadam — dastur uchun ichki direktoriya barpo etish. Eng yaxshisi, o'zingizni dasturlaringizni saqlash uchun direktoriya barpo etgan ma'qul bo'ladi va uning ichida mazkur muayyan dastur uchun ichki direktoriya barpo eting. Masalan, MYCODE direktoriyasini barpo etishingiz mumkin va uni ichida ikkinchi TIPS1 direktoriyasini, qayerda hozirgina tayyorlagan dasturingizni eslab qolasiz

Ichki direktoriya barpo etilgandan keyin, dasturingizni saqlab olish uchun menyudan File|Save Project punktini tanlab oling kerak. Ikkita faylni saqlab qolish kerak. Birinchisi, modul (unit), qaysini ustidan siz ish olib borgansiz, ikkinchisi, loyihaning bosh fayli, qaysi sizni dasturingizga "ega". Modulni MAIN.Pas nomi bo'yicha, loyihani esa TIPS1.DPR nomi bilan saqlab qoling (har qanday fayl Pas kengaytmasi bilan va dastur boshida "unit" so'zi bilan modul hisoblanadi).

#### **1.2.5. TBUTTON, DASTLABKI MATN, SARLAVHALAR VA Z-TARTIBLASH**

Obyektlar Inspektori va Shakllar Dizaynerini yana bir necha imkoniyatlari bor.

Yangi loyiha barpo eting. Shaklga Tmemo obyektini joylashtiring, keyin esa Teditni shunday joylashtiringki, u Tmemo ni yarmisini yopib tursin, 1.13-shaklda ko'rsatilgandek. Endi menyudan Edit|Send to Back punktini tanlang, bu Teditning ichkari tomoniga. Tmemo obyektini orqasiga qarab siljiydi. Buni komponentning Z tartibli o'zgarishi deb nomlanadi. Z harfi ishlatilishi sababi, matematikada uchinchi o'lcham u bilan belgilanadi. X va Y bilan eni va balandligi belgilansa, Z bilan chuqurligini belgilash uchun ishlatiladi.



1.13-shak. Tedit obyektning yarmisi Tmemo bilan yopilgan

Agar shaklda siz biror obyektни "yo'qotib" qo'ysangiz, Obyektlar Inspektorining yuqori qismida joylashgan ComboBox ro'yxatidan topib olishingiz mumkin.

Tbutton tugmachasini shaklning pastki qismiga o'rnating. Endi Obyektlar Inspektorini shunday cho'zingki, toki Name va Caption xususiyatlar ekranda bir vaqtda ko'rinarli bo'lmaguncha. Endi tugmachaning nomini Terminate ga o'zgartiring. E'tibor bering, (Caption) sarlavhasi shu zumdayoq almashib qoldi. Bunday ikkiyoqlama o'zgarishni kuzatish mumkin, qachonki Caption xususiyati avvalroq o'zgarmagan bo'lsa.

Tugmacha ustidagi siz ko'rayotgan yozuv — Caption xususiyatining ichidagi narsa, Name xususiyati ichki dalillarga xizmat qiladi, siz ularni dastur kodini yozayotganingizda ishlatasiz. Agar hozir siz Redaktor derazasini ochsangiz, quyidagi kodlar fragmetini ko'rasiz :

```
Tform = class (Tform)
Edit1: Tedit ;
Memo1 : Tmemo ;
Terminate : Tbutton ;
private
{ Private declarations}
public
{Public declarations}
end;
```

Bu fragmentda Tbutton tugmachasi Terminate deb nomlanadi, chunki siz Name xususiyati orqali shunday qilib uni nomladingiz. E'tibor bering, Tmemo nom aytib o'tmaslik prinsipi bo'yicha beriladigan nom.

Shaklga o'ting va sichqon bilan Tbutton obyektini shiqillating. Siz birdaniga Redaktor derazasiga tushib qolasiz, qayerda shunga o'xshash kodni ko'rasiz :

```
procedure TForm.TerminateClick (Sender : TObject);  
begin  
end;
```

Bu kod avtomatik ravishda barpo etildi va u ham doim bajariladi qachonki, foydalanuvchi Terminate tugmachasini bosadigan bo'lsa. Qo'shimcha sifatida, siz endi fayl boshida sinf aniqlanishi TerminateClick usuliga dalil kiritishini ko'rishingiz mumkin:

```
Tform1 = class(TForm)  
Edit1 : TEdit ;  
Memo1 : TMemo ;  
Terminate : TButton ;  
procedure TerminateClick (Sender ^ TObject) ;  
private  
{Private declarations}  
public  
{Public declarations}  
end;
```

Yuqorida berilganlarni o'zlashtirish uchun biroz vaqtingizni sarflang. Eng avval shakldagi tugmachaga qarang. Shu tugmachaga sichqon bilan ikki marotaba shiqillatasiz va tegishli kod fragmenti avtomatik ravishda Redaktorga kiritiladi.

Endi bir satr kodni yozish vaqti keldi. Bu juda ham oddiy bir so'z — Close iborali kod :

```
procedure TForm1.TerminateClick(Sender: TObject) ;  
begin  
Close ;  
end;
```

Qachonki shu kod bajariladigan bo'lsa, bosh shakl (demak va hamma yondoshlar ham) yopiladi. Kodni tekshirish uchun dasturni ishga soling va Terminate tugmachasini bosing. Agar hammasi to'g'ri



qilingan bo'lsa, dastur yopiladi va siz dizayn rejimiga qaytib kelasiz.

Keyingi qismga o'tishdan oldin, Obyektlar Inspektoriga o'ting va Name xususiyatini tugmacha uchun ixtiyoriy boshqaga o'zgartiring, masalan, OK. O'zgarishlarni kiritish uchun Enter ni bosing. Redaktorni ko'rsangiz, siz tomondan yozilgan kod o'zgargan bo'ladi:

```
procedure TForm1.OKClick(Sender: TObject);
begin
  Close;
end;
```

E'tibor bering, sinfni aniqlashda ham shuning singari o'zgarishlar sodir bo'ldi:

```
TForm1 = class(TForm)
  Edit : TEdit;
  Memo1 ^ TMemo;
  Terminate : TButton;
  procedure OkClick (Sender : TObject);
private
  {Private declarations}
public
  {Public declarations}
end;
```

### 1.3. LOYIHA USTIDAN BOSHQARISH

#### *Bu bobda*

- Loyihaga shakl va modullarni qo'shish va yo'qotish
- Ishchi joyda derazalar bilan boshqarish
- Windows uchun bajariladigan fayllarni barpo etish
- Dasturlash muhitini nozik ravishda sozlash to'g'ridagilarni bajarish mumkinligi haqida so'z boradi.

Har qanday loyiha hech bo'lmaganda u bilan bog'langan oltita faylga ega bo'ladi. Ularning uchitasi muhitdan loyihani boshqarishga kiradi va dasturchi ularni to'g'ridan-to'g'ri o'zgartirmaydi. Mana bu fayllar:

- Loyihaning bosh fayli, dastlab PROJECT1.DPR deb ataladi.
- Dasturning birinchi moduli (unit) u ishni boshida avtomatik ravishda paydo bo'ladi. Aytib o'tmaslik prinsipi bo'yicha fayl UNIT1.PAS deb nomlanadi, lekin uni ixtiyoriy boshqa nom bilan aytish mumkin,

MAIN.PAS kabi.

- Bosh shaklning fayli, aytib o'tmaslik prinsipi bo'yicha UNIT1.DFM deb nomlanadi, uni bosh shaklning tashqi ko'rinishi to'g'risidagi axborotni saqlash uchun ishlatiladi.

- Avtomatik ravishda barpo etiladigan va loyiha uchun ikonkaga ega bo'lgan PROJECT1.RES fayli.

- Aytib o'tmaslik prinsipi bo'yicha PROJECT1.OPT deb nomlanadigan fayl, u berilgan loyiha bilan bog'langan o'rnatuvlarni saqlovchi matnli fayl hisoblanadi. Masalan, siz o'rnatgan kompilyatorning direktivalarini saqlab turadi.

- Ish makonining holati to'g'risidagi axborotga ega bo'lgan PROJECT1.DSK fayli.

Tushunarli, agar loyihani boshqa nom bilan saqlanadigan bo'lsa, u holda RES, OPT va DSK kengaytmali fayllarning nomi ham o'zgaradi.

Dasturni kompilyatsiyalashtirishdan keyin fayllarning kengaytmalari bo'ladi :

DCU — kompilyatsiyalashtirilgan modullar ;

EXE — bajariluvchi fayl ;

DSM — dasturni muhitda ishga soluvchi xizmatchi fayl, juda ham katta, ish tugatilganda uni o'chirish tavsiya etiladi;

~PA, ~Dp — Muharrirning backup fayllari.

### *Menyuning "File" punkti*

Agar loyihani saqlash kerak bo'lsa, u holda bosh menyuning "File" punktini tanlang (sichqon yordamida yoki Alt Q Enter klavishasi bo'yicha). Menyuning "File" punkti quyidagicha ko'rinishli:

| <i>New Project</i> |             |
|--------------------|-------------|
| Open Project       | } 1 seksiya |
| Save Project       |             |
| Save Project As    |             |
| Close Project      |             |
| New From           | } 2 seksiya |
| New Unit           |             |
| New Component      |             |
| Open File          |             |
| Save File          |             |
| Save File As       |             |
| Close File         |             |

|             |   |           |
|-------------|---|-----------|
| Add File    | } | 3 seksiya |
| Remove File |   |           |
| Print       | } | 4 seksiya |
| Exit        | } | 5 seksiya |
| 1 PREV1.DPR | } | 6 seksiya |
| 2 PREV2.DPR |   |           |

*Ko'rib turganingizdek, bu yerda oltita seksiya bor, ularning xizmati mana bunday:*

- Birinchi seksiya loyihani butunlay boshqarish imkoniyatini beradi.
- Ikkinchi seksiya loyihaning shakllar, modullar va komponentlari ustidan nazorat tutadi.
- Uchinchi seksiya fayllarni qo'shish va loyihadan chiqarib yuborish imkonini beradi.
- To'rtinchi seksiya chop etilishni boshqaradi.
- Beshinchi seksiya — Delphidan chiqish.
- Oltinchi seksiya ilgari tahrirlangan loyihalar ro'yxatini taqdim etadi.

Keyinroq ko'rasizki, menyuning "File" punktidagi operatsiyalarning ko'pini Loyiha Menejeri (Project Manager) yordamida bajarish mumkin, uni menyuning View punktidan chaqiriladi. Ayrim operatsiyalar SpeedBar orqali bajarish imkoni bor. Bunday strategiya Delphiga xarakterli: u bir masalani yechish uchun bir necha yo'l taqdim etadi, vaziyatga qarab, siz ularning ichidan samaralirog'ini tanlab olasiz.

Menyuning "File" punktining har bir satri ma'lumotnomalar to'plamida sharhlangan. Menyuning "File" punktini tanlang va F1 ni bosib qo'ysangiz, ekranda ma'lumotnomalar to'plami paydo bo'ladi.

Birinchi seksiya punktlarining ko'pi muqarrar. "New Project" yangi loyihani boshlaydi, "Open Project" mavjud loyihani ochadi va h.k.

Ikkinchi seksiyaning dastlabki ikkita punkti yangi shakl yoki yangi modulni barpo etishga imkon beradi. "New Form"ni tanlab, siz yangi shaklni va u bilan bog'langan modulni barpo etasiz; "New Unit" tanlab, bir modulni barpo etasiz.

"New Component" yangi vizualli komponentning tayyorlovini quruvchi dialogni chaqiradi. Natijada modul barpo etiladi, uni kompilyatsiyalashtirish va Komponentlar qatoriga kiritish mumkin bo'ladi.

"Open File" kerak bo'lganda har qanday modulni ochadi yoki oddiy matnli faylni. Agar modul shaklni ta'riflayotgan bo'lsa, u holda bu shakl ham ekranda paydo bo'ladi.

Yangi modul barpo etilayotganda, aytib o'tmaslik prinsipi bo'yicha Delphi unga nom beradi. Siz bu nomni "Save File As" punkti yordamida biror ma'noliroq nomga o'zgartirishingiz mumkin (masalan, Main.Pas).

"Save File" loyiha barchasini emas, faqat tahrirlanayotgan faylni saqlab qolishingiz mumkin.

"Close File" Muharrirning derazasidan fayllarni yopadi.

Shunga e'tibor berish lozim : File | Save Project yoki Ctr + S bosish orqali muntazam ravishda loyihani saqlamoq kerak.

### 1.3.1. BOSHQA MENYULAR BILAN ISHLASH

#### *Tuzatish — menyuning "Edit" punkti*

"Edit" punkti "Undo" va "Redo" buyruqlariga ega, ular juda ham foydali bo'ladi, qachonki muharrirda ishlash paytida noto'g'ri harakatning oqibatini yo'qotish zarur bo'lib qolsa, masalan, agar matnning kerakli fragmenti tasodifan chiqarib yuborilgan bo'lsa.

Shuni eslab qolingki, Ma'lumotlar to'plami (on-line-help) qanday qilib menyuning Option | Environment punktini "Undo" buyruqni sozlash uchun ishlatish kerakligini izohlaydi. Agar siz resurslari cheklangan mashinada ishlayotgan bo'lsangiz, "Undo" buyruqlarning imkoniyatli sonini cheklash imkoniyati kerak bo'ladi.

"Cut", "Copy", "Paste" va "Delete" buyruqlari — Windowsning boshqa yondoshlaridagi singari, lekin ularni nafaqat matnlar uchun balki vizualli komponentlar uchun ham ishlatish mumkin.

"Bring To Front", "Send To Back", "Align" va "Size" buyruqlar ilgariroq muhokama qilingan edi. Bu to'rtta punktlar shaklning tashqi ko'rinishini tezda "bezash" uchun yordam beradilar.

### *Izlash — menyuning "Menu" punkti*

"Search"da "Find Error" (xatoni topish) buyruq bor, u dasturni ishlash paytidagi xatoni topishga yordam beradi. Qachonki xatolar to'g'risidagi xabarnomada uni adresi ko'rsatilgan bo'lsa, siz menyuning Search | Find Error punktini tanlab, shu adresni kiritishingiz mumkin. Agar bu imkoniyati bo'lsa, muhit sizni dasturda xato paydo bo'lgan joyga olib keladi.

### *Ko'rinish — menyuning "View" punkti*

Menyuning "View" punktining tarkibi :

- Project Manager (Loyiha Menejeri).
- Project Source — Muharrirga loyihaning bosh faylini (DPR) yuklaydi.
- O'rnatish, u Object Inspectorni ekranda ko'rsatish yoki ko'rsatmaslikni aniqlaydi.
- Alignment Paletteni ko'rsatish yoki ko'rsatmaslikni aniqlovchi o'rnatuv.
- Browser — dastur obyektlarining iyerarxik bog'lanishlarini ko'rish, dastlabki matnlarda identifikatorni izlash uchun vositani chaqiradi va h.k.
- Watch, Breakpoint va Call Stack — dasturni sozlovchi protsedurasi bilan bog'langan va u keyinroq ta'riflanadi.
- Component List — komponentlar ro'yxati, Komponentlar qatoriga alternativa. Komponentni nomi bo'yicha topish yoki sichqon qurilmasi yo'q bo'lgan hollarda ishlatiladi.
- Windows List — Delphi muhitida ochilgan derazalar ro'yxati.
- Toggle FormG'Unit, Unit, Forms — shakldan tegishli modulga o'tish, modul yoki ro'yxatdan shaklni tanlash.
- New Edit Windows — Muharrirning qo'shimcha derazasini ochadi. Qachonki bir faylning ikki versiyasini ko'rib chiqish kerak bo'lib qolganda bu boydali bo'ladi.
- SpeedBar va Somponent — ularni aks etirish kerakligini bildiruvchi o'rnatuvlar.

### *Loyiha — menyuning "Compile" punkti*

Menyuning "Compile" punktida loyihani kompilyatsiyalashtirish (compile) yoki qayta qurish (build) mumkin. Agar Compile yoki Run tanlab olinsa, u holda Delphi faqat oxirgi kompilyatsiyadan keyingi vaqt ichida o'zgargan modullarni qaytadan kompilyatsiyalashtiradi. Ikkinchi tomondan, Build all dastlabki matnlari kira olishli bo'lgan barcha modullarni qaytadan kompilyatsiyalashtiradi. Syntax Check buyrug'i faqat dasturning kodlarini tekshiradi, lekin DCU fayllarini yangilamaydi.

Eng pastda - Information punkti bo'lib, u dastur to'g'risida axborot chiqazadi: kod segmentlari, berilganlar va stekning o'lchamlari, lokal dinamikli xotiraning o'lchami va kompilyatsiyalashtirilgan satrlar soni.

### *Ishga solish (Push) — menyuning "Run" punkti*

"Run"ni dasturni kompilyatsiyalashtirish va ishga solish hamda buyruq satrining parametrlarini dasturga uzatish uchun ishlatish mumkin. Shu yerda sozlash rejimi uchun opsiyalar ham mavjud.

### *Loyiha opsiyalari — menyuning Option | Project punkti*

"Options" — tizimli menyuning eng murakkab qismi hisoblanadi. Bu boshqarish markazi, u yerdan siz loyiha va Delphining barcha ishchi muhiti uchun o'rnatuvlarni o'zgartirishingiz mumkin. "Options" da yettita punkt mavjud :

- Project — loyiha ;
- Environment — atrof ;
- Tools — instrumentlar ;
- Gallery — arxiv ;
- Open Library — kutubxonani ochish ;
- Install Components — komponentni instalizatsiyalashtirish;
- Rebuild Library — kutubxonani qaytadan barpo etish.

Birinchi to'rtta punktlar dialogli derazani chaqiradi. Quyida menyuning "Option" punktining umumiy ta'rifı keltirilgan:

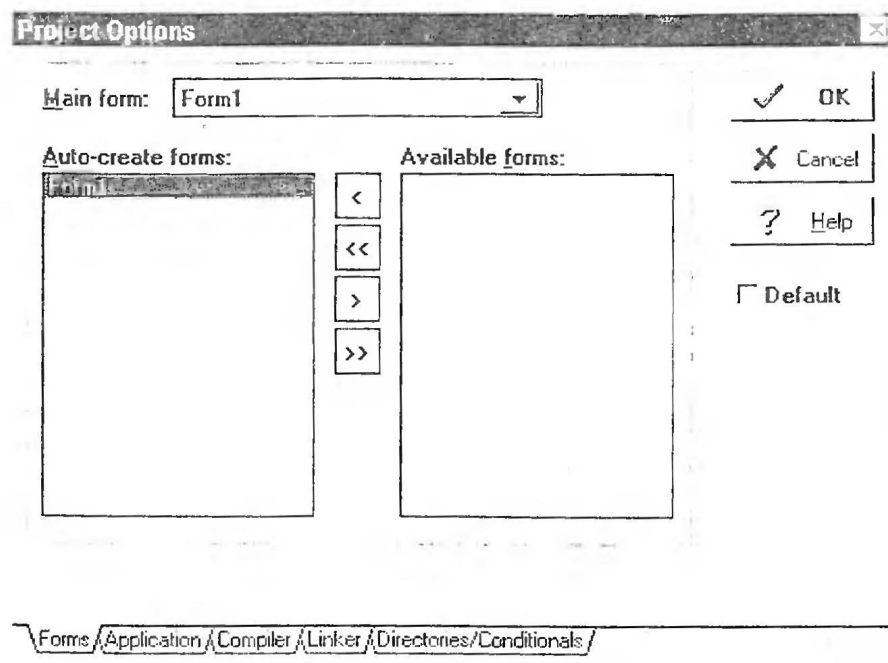
- Project — o'rnatuvni tanlash, ular to'g'ridan-to'g'ri joriy loyihaga ta'sir etadilar, misol uchun, stekni tekshiruvchi kompilyator direktivalari.
- Environment — dasturlash muhitini (IDE) o'zini konfiguratsiyalash. Masalan, bu yerda Muharrirda ishlatiladigan ranglarni o'zgartirish mumkin.
- Tools — bosh menyuning "Tools" punktiga tashqi dasturlarning chaqiruvini qo'shish yoki chiqarib yuborish imkonini beradi.
- Gallery — Shaki va Loyiha Ekspertlari uchun hamda ularning "tayyorlov"lariga maxsus o'rnatuvlar aniqlash imkonini beradi. Ekspertlar va "tayyorlov"lar dastur interfeysining konstruksiyalashini tezlashtirilgan yo'lini taqdim etadi.
- Oxirgi uchta punktlari Komponent qatorini konfiguratsiyalash imkonini yaratadi.
- Option | Project punktning dialogi besh betdan iborat :
- Forms (Shakllar) beti — bu betda loyihaga kiritilgan barcha shakllar keltirilgan. Siz shuni ko'rsatishingiz mumkinki, dasturni dastlabki ishlash paytida shaklni avtomatik ravishda barpo etish kerakmi yoki uni o'zingiz barpo etasizmi.
- Application (Yondosh) beti — bu betda siz dasturning elementlari bo'lgan tayyorlovlar, yordamlash fayli va ikonkani aniqlashingiz mumkin.
- Compiler (Kompilyatsiyalash) beti — bu bet kodni generatsiyalash o'rnatuvini, bajarish vaqti, sintaktik, sozlash va boshqa xatolar ishlovining boshqarishini kiritgan.
- Linker (komponovshik-tuzuvchi) beti — bu betda yondoshni tashkil etuvchi jarayon shartlari aniqlanadi.
- Directories/Conditionals (Katalog/Shartlar) beti — bu betda mazkur loyihaga maxsus direktoriyalar ko'rsatiladi.

Bundan tashqari bu punktdan tizimning versiyasi, foydalanadigan dasturlar paketi to'g'risida ma'lumotli axborot olish mumkin.

Har bir bet to'g'risida quyida batafsil ta'rif berilgan. Loyihaning barcha o'rnatuvlari OPT kengaytma bilan matnli fayllarda saqlanadi va siz ularga qo'l bilan tuzatishlar kiritishingiz mumkin.



## Forms beti



1.14-shakl.

Forms betida loyihaning bosh shaklini tanlash mumkin. Siz tomondan kiritilgan o'zgarishlar tegishli DPR faylida aks etiriladi. Masalan, quyidagi loyihada Form1 bosh shakl hisoblanadi, chunki u dasturning bosh blokida birinchi bo'lib paydo bo'ladi:

```
program Project ;
uses Forms,
    Unit1 in 'UNIT1.PAS' {Form1},
    Unit2 in 'UNIT2.PAS' {Form2} ;
{$R * .RES}
begin
    Application.CreateForm(TForm1, Form1);
    Application.CreateForm(TForm2, Form2);
    Application.Run ;
end.
```

Agar kodni mana bunday qilib o'zgartirilsa,

```
begin
    Application.CreateForm(TForm2, Form2);
    Application.CreateForm(TForm1, Form1);
```

```
Application.Run ;  
end.
```

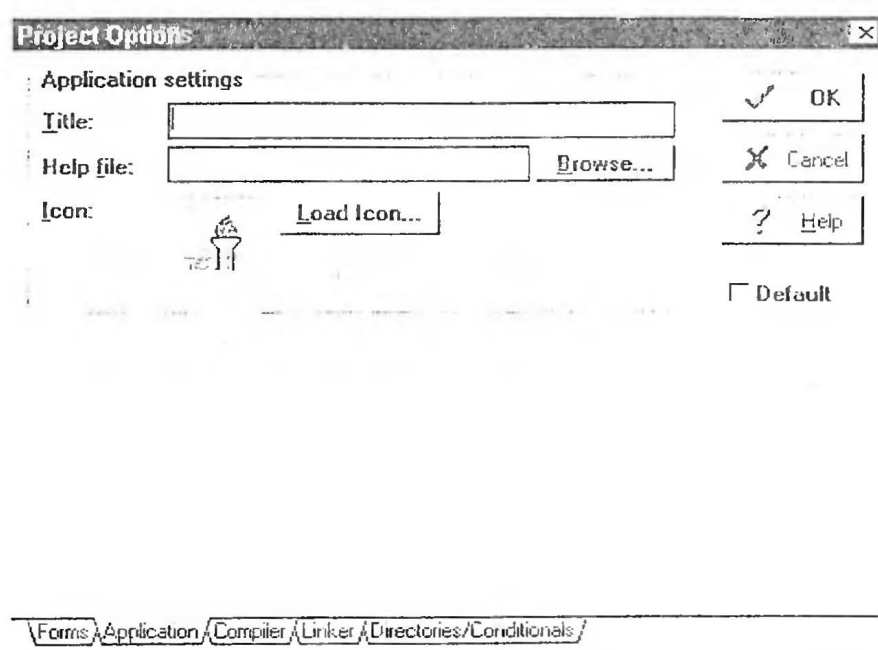
u holda loyiha bosh shakli sifatida Form2 bo'ladi.

Bundan tashqari siz bu betni dasturning dastlab ishlash paytida mazkur shakl avtomatik ravishda barpo etilishini aniqlashingiz mumkin. Agar shakl avtomatik ravishda emas, balki dasturning ishlab borishida barpo etilayotgan bo'lsa, u holda buning uchun Create protsedurasi bilan foydalanish kerak.

Zero, Uses seksiyasida figurali qavs ichida shaklning nomi Loyiha Menejeri uchun muhim hisoblanadi va uni chiqarib yuborish arzimaydi. Loyiha faylida qo'lda umuman hech qanday o'zgarish kiritish kerak emas, agar DLL ni barpo etmaslikka qaror qilgan bo'lsangiz, bu to'g'risida keyinroq ko'rib chiqamiz.

### *Applications beti*

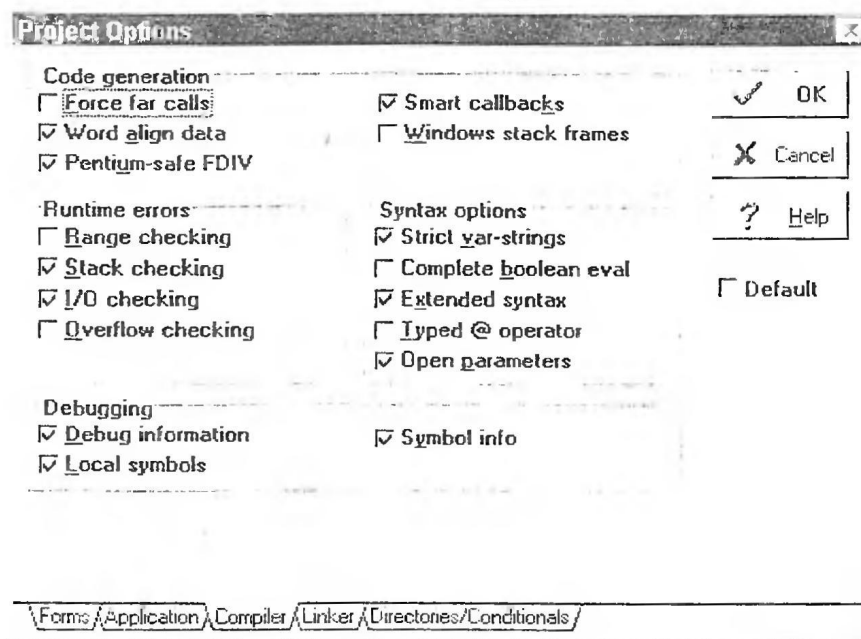
Applications betida (tegishli 10.15 shaklga qarang) siz loyiha uchun sarlavha (Title), yordamlashuvchi fayl (Help file) va piktogramma (Icon)ni berishingiz mumkin.



1.15-shakl. Yondosh uchun umumiy o'rnatuvlar beti.

### Compiler beti

Bundan oldinroq aytib o'tilgandek, menyuning "Options|Project punktdagi o'rnatuvlar tegishli OPT kengaytmali faylida saqlanib qoladi. Kelinglar, Compiler (tegishli 1.16-shaklga qarang) betidagi kompilyatorning direktivalarini ko'rib chiqsak.



1.16-shakl. Kompilyatorning direktivalarini aniqlash uchun bet.

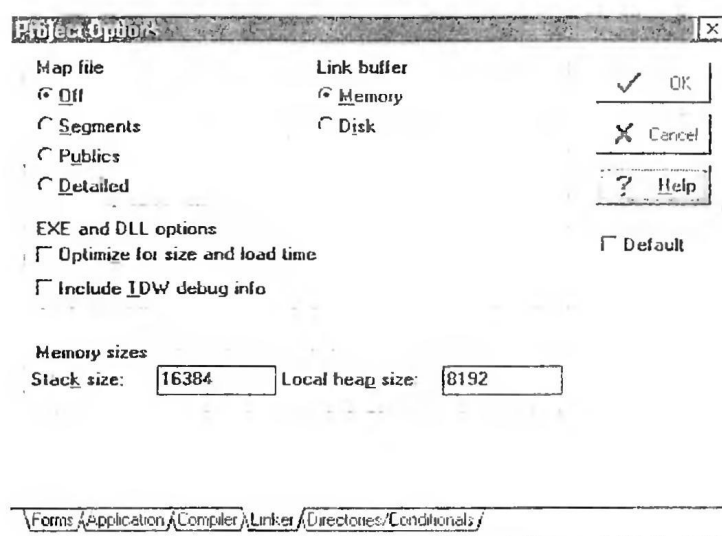
Quyidagi jadval qanday qilib har xil direktivalar qanday qilib OPT faylida, Compiler betida va dastur kodi ichida aks ettirishini ko'rsatadi.

| OPT File | Options Page                | Editor |
|----------|-----------------------------|--------|
| F        | Force Far Calls             | {SF+}  |
| A        | Word Aling Date             | {SA+}  |
| U        | Pentium-Safe FDIV           | {SU+}  |
| K        | Smart Callbacks             | {SK+}  |
| W        | Windows (3.0) Stack Frame   | {SW+}  |
| R        | Range Checking              | {SR+}  |
| S        | Stack Checking              | {SS+}  |
| I        | IO Checking                 | {SI+}  |
| Q        | Overflow Checking           | {SQ+}  |
| V        | Strict Var Strings          | {SV+}  |
| B        | Complete Boolean Evaluation | {SB+}  |
| X        | Extended Syntax             | {SX+}  |

|   |                    |          |
|---|--------------------|----------|
| T | Typed @ Operator   | { \$T+ } |
| P | Open Parameters    | { \$P+ } |
| D | Debug Information  | { \$D+ } |
| L | Local Symbols      | { \$L+ } |
| Y | Symbol Information | { \$Y+ } |
| N | Numeric Processing | { \$N+ } |

### *Linker beti*

Kelinglar, endi keyingi 1.17.shaklda ko'rsatilgan Linker betiga o'tamiz.



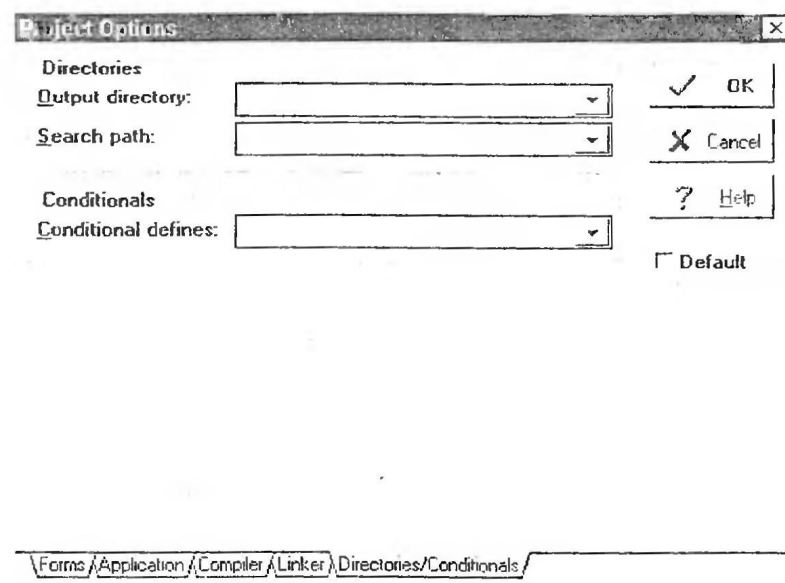
1.17-shakl. Linker (tuzuvchi) beti.

Sozlovchining o'rnatuvlari quyida ko'rsatilgan. Agar tuzuvchining buferi xotirada joylashgan bo'lsa, tuzilish tezroq bajariladi.

Stek (Stack Size) va lokal dinamikli xotiraning (Heap Size) o'lchamlari juda ham muhim. Aytib o'tmaslik prinsipi bo'yicha Delphi Stack Size va Heap Size lar uchun har biriga 8192 bayt qilib o'rnatadi. Dasturda stekning o'lchamini o'zgartirish kerak bo'lib qolsa, uni 32 Kbgacha oshirish mumkin. Bu ikki o'lchamning yig'indisi 64 Kb dan oshmasligi kerak, aks holda dasturning kompilyatsialash xatosi chiqadi.

### *Directories/Sonditionals beti*

Directories/Sonditionals beti (tegishli 1.18-shaklga qarang) direktoriyalar sonini kengaytirish imkonini beradi, ularda kompilyator va tuzuvchi DCU fayllarini izlaydi.



1.18-shakl. Directories/Sonditionals beti

DELPHI.INI faylida yana bir direktoriyalar ro'yxati bor. Shuni eslab qolinki, OPT faylida loyiha uchun direktoriyalar ro'yxati bor, DELPHI.INI faylida esa — ixtiyoriy loyiha uchun.

Output directory — chiqaruv direktoriyasi, u yerga EXE va DCU fayllari kompilyatsiyadan keyin joylashadi.

Search path — tuzilish paytida DCU fayllarini izlovchi direktoriyalar ro'yxati. Direktoriyalar nuqtali vergul bilan ajratib keltiriladi.

Conditional defines — tajribali dasturchi uchun va loyihaning barpo etilishining birinchi bosqichida kerak bo'lmaydi. Axborot uchun Ma'lumotlar to'plami (on-line help)ni chaqirish mumkin.

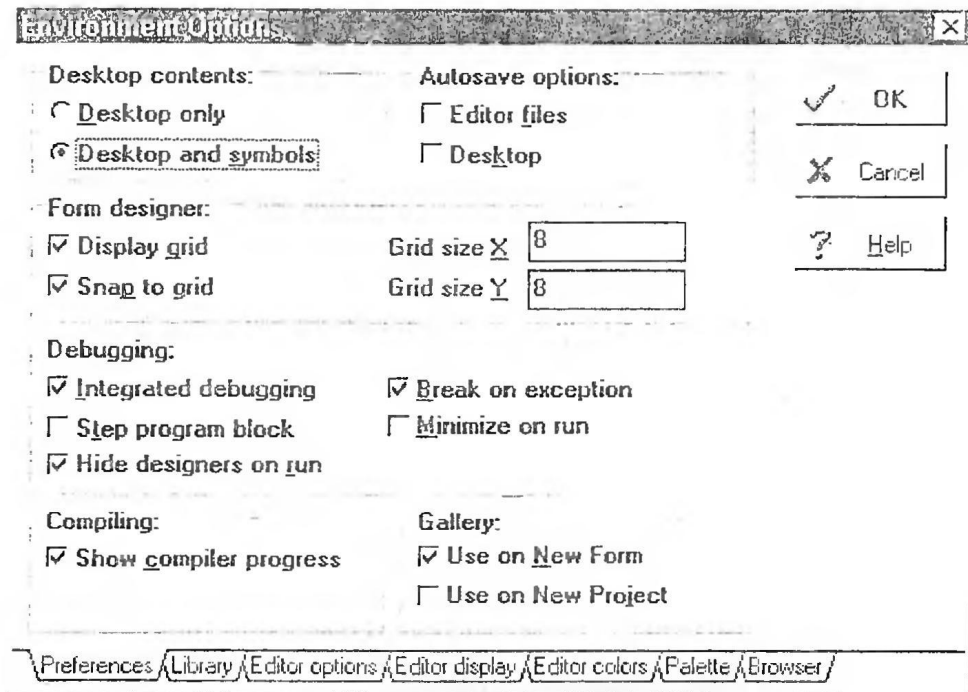
### 1.3.2. DASTURLASH MUHITINI KONFIGURATSIYALASH (IDE)

Menyuning "Options | Environment" punkti sizga bet va elementlarning katta to'plamini taqdim etadi, ular IDEning tashqi ko'rinish va ishini aniqlaydilar. Delphi quyidagi muhim sozlovlarni bajarish imkonini beradi :

1. Loyihadan nimalar avtomatik ravishda saqlanishini aniqlash.
2. IDEning rangini o'zgartirish.

3. Muharrirda sintaktikning yoritilishini o'zgartirish.
4. Komponentlar qatorining tarkibini o'zgartirish.
5. IDEning "qaynoq klavisha"larini ko'rsatish.

Menyuning "Options | Environment" punktdagi birinchi beti quyidagi 1.19. shaklda ko'rsatilgan.



1.19-shakl. Preferences beti.

"Desktop Contents" guruhida Delphidan chiqilganda nimalar saqlanishi aniqlanadi. Agar Desktop Only tanlansa, bu direktoriyalar va ochilgan derazalar to'g'risidagi axborotni saqlab qoladi, agar Desktop And Symbols tanlansa, bu o'shalardan tashqari brouzerga (browser) bo'lgan axborotni ham saqlab qoladi.

"Autosave" guruhida dastur ishga tushirilganda nimalarni saqlab qolish ko'rsatiladi. Agar Editor Files o'rni tanlangan bo'lsa, u holda Run | Run, Run | Trace Into, Run | Step Over, Run | Run To Cursor buyruqlari bajarilgan paytida yoki Delphidan chiqishda Muharrirdagi barcha modifikatsiyalangan fayllar saqlanib qoladi. Agar Desktop o'rni tanlangan bo'lsa, loyihaning yopilishida yoki Delphidan chiqishda ishchi muhit saqlanib qoladi. Agar siz keyinroq loyihani ochsangiz, u holda u xuddi yopilishdan oldingi ko'rinishga ega bo'ladi.

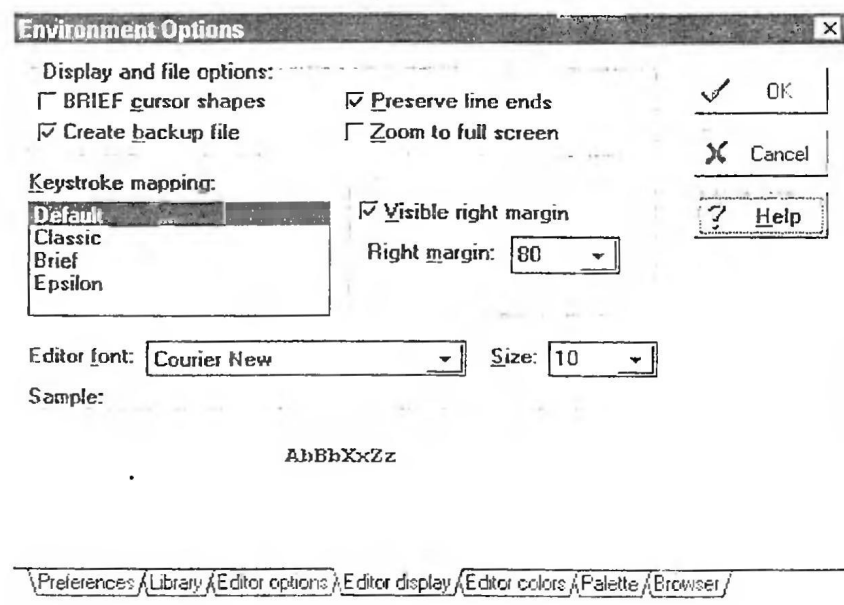
"Form Designer" guruhida to'rni (grid) ekranda ko'rsatish va unda obyektlarni to'g'rilash kerakligini hamda to'rning kataklari o'lchamini o'rnatish mumkin.

"Debugging" guruhida: Integrated Debugging — kiritilgan sozlovchini ishlatish kerakmi; Step Program Block — sozlovchi modulning birinchi satrida to'xtaydi, agarda shu satrda sozlovchi axborot bo'lsa; Break On Exception — maxsus vaziyat paydo bo'lganda dasturni to'xtashga qo'yadi; Minimize On Run — dasturni ishga qo'yilganda Delphini yig'ishtirish kerakmi? Dastur yopilgandan keyin Delphining muhiti tiklanadi. Hide Designers On Run — yondoshni ishga qo'yilganda Dizaynning derazalarini (Obyektlar Inspektori, shakllar) berkitadi.

Show Compiler Progress — dasturni kompilyatsialash jarayonini aks ettiruvchi derazani ko'rsatish kerakmi?

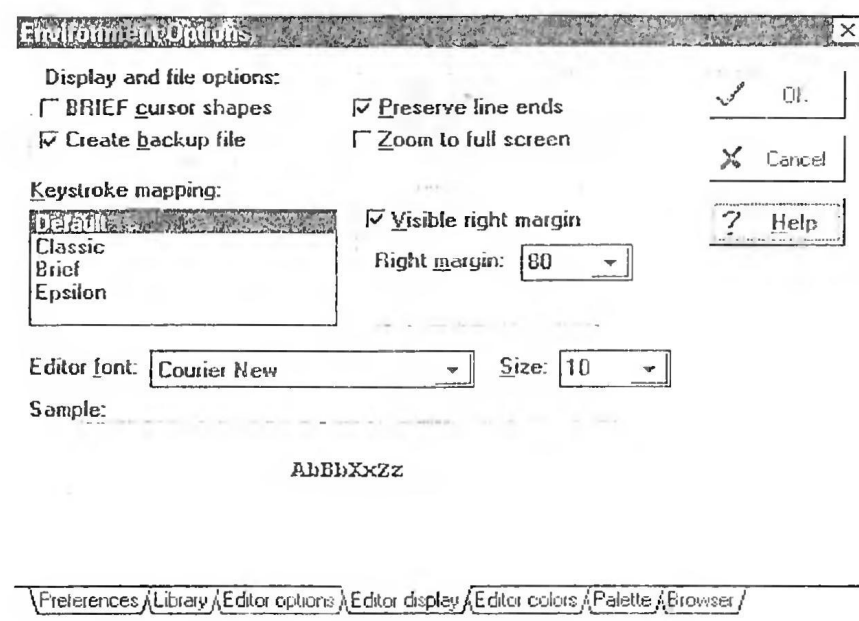
"Gallery" — qanday vaziyatlarda "yo'lakni" (galeriyani) taqdim etish kerak (tayyorlov va eksperlar kolleksiyasi).

Editor Options, Editor Display va Editor Colors betlari IDE tomonidan foydalanadigan "qaynoq klavisha"ni hamda ranglarni o'zgartirish imkonini beradi. Editor Display va Editor Colors betlari quyida 1.20, 1.21-shakllarda ko'rsatilgan.



1.20-shakl. Editor Display beti.





1.21-shakl. Editor Solors beti.

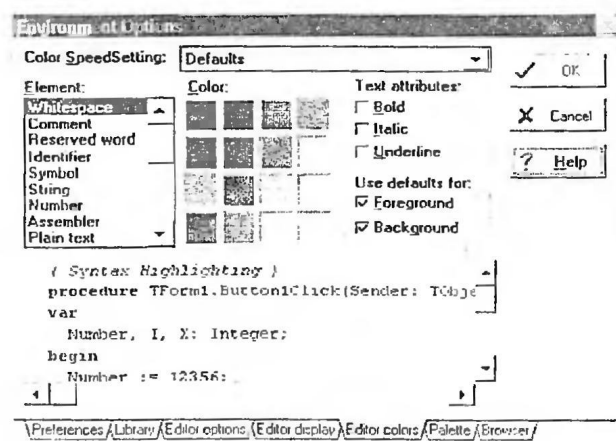
Muharrir tomonidan ishlatiladigan "qaynoq klavisha"larning xizmatini o'zgartirish uchun bir necha usullar mavjud. Masalan, ko'p foydalanuvchilar shunga o'rganib qolganlarki, F5 klavishasi Muharrirning derazasini maksimal holatga keltiradi. Buning uchun ular "Classic"(Keystroke mapping: Classic) deb nomlanadigan klavishalardan foydalanishlari kerak. Klavishalar konfiguratsiyasining to'rtta ko'rinishi bor :

- "Default" — Microsoftga xarakterli. Agar siz Windowsda ko'p ishlamagan yoki klavishalarni shu joylashiga o'rganib qolgan bo'lsangiz, u holda bu sizga to'g'ri keladi.
- "Classic" — Borland C++ va Borland Pascal veteranlariga ko'proq ma'lum bo'lgan. WordStarning ko'p klavishalar kombinatsiyasini quvvatlaydi va sozlovchi eski puxta usul bilan boshqariladi.
- Qolgan ikki ko'rinishlar — Epsilon va BRIEF muharrirlariga taqlid qiladi. Sizga to'g'ri kedadi, agarda u bilan tanish bo'lsangiz.

Klavishalar xizmati to'g'risidagi aniq ta'rifni Ma'lumotlar to'plami (Help | Topic Searchda "key mapping"ni termog kerak)da topish mumkin.

IDening rangini Editor Colors betida o'zgartirish mumkin.

Va, nihoyat, Editor Options betida quyidagi 10.22-shakl ko'rsatilgan.



1.22-shakl. Editor Options betida Muharrirning ishidagi nozik detallarni sozlash mumkin.

Mazkur betdagi o'rnatuvlarni ko'pi foydalanuvchilarning ko'pchiligiga muhim emas, shuning uchun ularning ayrimlarida to'xtalib o'tamiz.

"Use syntax highlight" — Dastlabki matn muharririda sintaktik konstruksiyalarni rang bilan ajratish kerakmi?

"Find text at cursor" — agar yondirilgan bo'lsa, u holda izlash uchun (Ctrl+F) catr sifatida kursor turgan so'z olinadi.

Barcha opsiyalar to'g'risida Ma'lumotlar to'plamidan (F1) batafsilroq o'rganish mumkin.

O'rnatuvlar Windowsning direktoriyasida joylashgan DELPHI.INI faylida saqlanadi.

## 1.4 KOMPONENTLAR QATORI TO'G'RISIDA QISQACHA MA'LUMOT

Bu bobda Delphi ning Komponentlar qatoridan standartli va qo'shimcha komponentlar to'g'risida qisqacha ma'lumot berilib boriladi (Standart va Additional betlari), dialog betlari Dialogs), tizimli komponentlar (System ,tnb), Visual Basic (VBX) formatdagi obyektlar betlari. Berilganlarga kira olish va ularni ekranda aks ettirish to'g'risidagi komponentlar keyinroq ko'riladi.

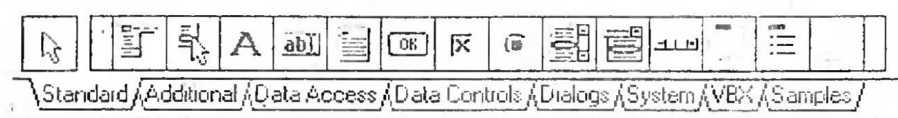
### 1.4.1 STANDARTLI KOMPONENTLAR

Komponentlar qatorining birinchi betidagi komponentlari bilan siz 10.2 bobida tanishib chiqdingiz.

Bu bobda Additional betidagi komponentlar bilan tanishtirib o'tamiz.

#### 1.4.2 ADDITIONAL BETIDAGI KOMPONENTLAR

Standart betida Windows 3.0 da paydo bo'lgan boshqaruvchi elementlar keltirilgan. Additional betida joylashgan obyektlar dasturning foydalanuvchi interfeysi chiroyliroq qilib barpo etilgan.



1.23-shakl.

#### Komponentlar ro'yxati :

 **TbitBtn** — tugmacha, Tbutton singari, biroq unda suratchani joylashtirish mumkin (glyph). TbitBtn oldindan belgilanadigan bir necha tiplari bor (bkClose, dkOK va boshqalar), ular tanlab olinganda tugmacha tegishli ko'rinishni oladi. Bundan tashqari, modal derazada (Form2.ShowModal) tugmacha bosilganda derazaning yopilishiga olib keladi tegishli modalli natija bilan (Form2.ModalResult).

 **TSpeedButton** — tezlik bilan buyruqlarga (SpeedBar) kira olishni ta'minlaydigan panelni barpo etish. Misol — SpeedBar Delphi muhitidagi Komponentlar qatoridan chap tomonda. Odatda bu tugmachaga suratchalar (glyph) joylashtiriladi.

 **TTabSet** — gorizontal xatcho'plar. Odatda TnoteBook bilan birgalikda ko'p betli derazachalarni barpo etishda foydalaniladi. Betning nomlanishi Tabs xususiyatida berilishi mumkin. Oddiyroq qilib buni shakl barpo etilayotgan (OnCreate) dasturida ko'rsatgan ma'qul bo'ladi :

```
TabSet1.Tabs := Notebook1.Pages ;
```

Xatcho'pi tanlab olinganda betlar varaqlanishi uchun, TtabSet uchun OnClick hodisani ishlovchisida quyidagini yozib qo'yish kerak :

```
Notebook1.PageIndex := TabSet1.TabIndex ;
```

 **TNoteBook** — ko'p betli dialogni barpo etishda foydalaniladi, har bir betda o'z obyektlar to'plami joylashgan. TtabSet bilan birgalikda ishlatiladi.

 **TtabbedNotebook** — ko'p betli dialog xatcho'plari bilan o'rnatilgan, bu hodisada - xatcho'plar yuqorida.

 **TMaskEdit** — TEditning o'xshashi bo'lib, lekin formatlangan kiritish imkoniyatlari bilan. Format EditMask xususiyatida aniqlanadi. EditMask uchun xususiyat muharririda ayrim formatlarning sarlavhasi bor: sana, valutarlar va h.k. Maska uchun maxsus simvollarni ma'lumotnoma to'plamidan ko'rib olish mumkin.

 **TOutline** — bog'langan berilganlarni iyerarxik munosabatini taqdim etadi. Masalan, direktoriyaning daraxti.

 **TStringGrid** — matnli berilganlarni jadval ravishda taqdim etish uchun xizmat qiladi. Jadvalni har bir elementiga kira olish imkoniyati Cell xususiyati orqali kelib chiqadi.

 **TDrawGrid** — ixtiyoriy tipdagi berilgalarni jadval ko'rinishda taqdim etish uchun xizmat qiladi. Jadvalni har bir elementiga kira olish imkoniyati CellRect xususiyati orqali kelib chiqadi.

 **TImage** — shaklda grafikli tasvirni aks ettiradi. Formatlardan BMP, ICO, WMFlarni o'zlashtiradi. Agar suratchani dasturning dizayni vaqtida kiritilsa, u kompilyatsiyalashib, EXE fayliga qo'shiladi.

 **TShape** — shaklda oddiy grafikli obyektlarni aks ettirish uchun xizmat qiladi: aylana, kvadrat va h.k.

 **TBevel** — interfeysni bo'rttirib, bezaladigan element

 **THeader** — bu element jadvallar sarlavhasini o'lchamlarini o'zgaruvchanli ravishda bezatish.

 **TScrollBar** — bu ekrandan ancha katta bo'lgan sohani shaklga aylantirib ko'rsatish imkonini yaratadi. Shu sohada o'z obyektlarimizni joylashtirishimiz mumkin.

### 1.4.3 DIALOGS BETI

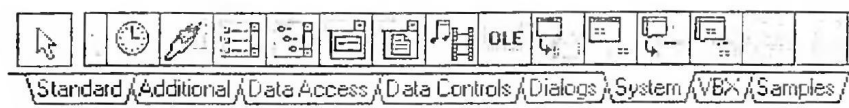


1.24-shakl.

Dialogs betida Windowsning standartli dialoglarini chaquruvchi komponentlar taqdim etilgan. Dialoglarning tashqi ko'rinishi foydalanilayotgan Windowsning versiyasiga bog'liq. Mazkur betda taqdim etilgan obyektlar ishlash vaqtida ko'rimsiz va dialoglarni chaqirish dasturda amalga oshiriladi, masalan:

If OpenFileDialog.Execute then  
 Image . Picture.LoadFile (OpenFileDialog.FileName);  
 Windows dialoglarini Dialogs betida paydo bo'lish tartibi:  
 OpenFileDialog — faylni tanlash;  
 SaveDialog — faylni saqlash;  
 FontDialog — shriftni sozlash;  
 ColorDialog — rang tanlash;  
 PrintDialog — chop etish;  
 PrinterSetupDialog — printerni sozlash;  
 FindDialog — catrni izlash;  
 ReplaceDialog — almashtiruv bilan izlash.

#### 1.4.4 System beti



1.25-shakl.

Bu bet komponentlar to'plamidan iborat bo'lib, ular bilan ayrim tizimli servislar bo'lgan taymer, DDE, OLE va h.k. lar kabilarga kira olish ta'minlanadi.

 **Timer** — taymer, OnTimer hodisasi Interval xususiyatida ko'rsatilgan vaqt oralig'ida vaqti-vaqti bilan chaqirilib turiladi. Davr vaqti 1 dan 65535 ms gacha tashkil qilinishi mumkin.

 **TPaintBox** — chizish uchun joy. Sichqon bog'langan hodisaning ishlovi beruvchisiga TpaintBoxda sichqonni absolut koordinatalari emas, nisbiy koordinatalar uzatiladi.

 **TFileListBox** — maxsus ListBox bo'lib, u yerda ko'rsatilgan direktoriyadagi (Directory xususiyati) fayllar aks ettiriladi. Fayllar nomiga maska qo'yish mumkin, buning uchun Mask xususiyati xizmat qiladi. Bundan tashqari, FileEdit xususiyatida maskani tahrirlash uchun TEdit obyektini ko'rsatish mumkin.

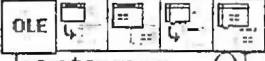
 **TDirectoryListBox** — maxsus ListBox bo'lib, u yerda joriy disk direktoriyasining strukturasi aks ettiriladi. FileList xususiyatida TFileListBox ni ko'rsatish mumkin, u yerda boshqa direktoriyaga o'tish avtomatik ravishda kuzatib turiladi.

 **TDriveComboBox** — maxsus ComboBox joriy diskni tanlash exey. DirList xususiyatiga ega, u yerda TdirectoryListBox ni ko'rsatish mumkin, u boshqa diskka o'tilishini kuzatib turadi.

 **TFilterComboBox** — fayl nomining maskasini tanlash uchun maxsus ComboBox. Maskalar ro'yxati Filter xususiyatida aniqlanadi. FileList xususiyatida TFileListBox ko'rsatiladi va unga maska o'rnatiladi.

!!!! Oxirgi to'rtta komponentlar yordamida (TFileListBox, TDirectoryListBox, TDriveComboBox, TfilterComboBox) fayl aniqlash uchun o'z xususiy dialogni qurish mumkin, buning ustiga buning uchun hattoki bir qator kodni yozishning keragi yo'q.

 **TMediaPlayer** — multimediyali qurilmalarni (CD-ROM, MIDI va boshqalar) boshqarish uchun xizmat qiladi. Play, Stop, Record va boshqalar tugmachalari bilan boshqarish paneli ko'rinishida bajarilgan. Multimediani ishga tushirish uchun tegishli asbob-uskunalaridan tashqari dasturli ta'minot ham kerak bo'ladi. Qurilmalarni ulash va dasturli ta'minotni o'rnatish Windows muhitida bajariladi. Masalan, AVI formatida video yozuvini ko'rsatish uchun, Microsoft Video dasturli ta'minotni o'rnatish kerak bo'ladi (Windows 3.0, 3.1, WFW 3.11).

 **TOLEContainer** — OLE obyektlariga ega bo'lgan konteyner. OLE 2.02 tomonidan quvvatlanadi. Bu to'g'risida keyin batafsilroq ma'lumot beramiz.

**TDDEClientConv, TDDEClientItem, TDDEServerConv, TDDESerItem** — bu 4 ta obyektlar DDE ni tashkil qilishga ishlatiladi. Ular yordamida DDE-server hamda DDE-kliyent kabilarning amaliyotlarini tuzish mumkin.

### VBX betlari



1.26-shakl.

Microsoft Visual Basic (VBX) ning obyektlari o'zicha standart bo'lgani sababli va bunday obyektlar kutubxonasi mavjudligi uchun Delphi da shu formatga sig'ishlik nazarda tutilgan. VBX ning 1.0 versiyasini Delphini Komponentlar qatoriga kiritib, "o'zinikiday" komponentlarini ishlatish mumkin.



**TBiSwitch** — ikki pozitsiyali o'zgartiruvchi kalit



**TbiGauge** — progress indikatori



**TBiPict** — Timage ni o'xshashi



**TChartFX** — ishga doir grafika

## 1.5. DELPHI DAGI XUSUSIYATLAR

Har bir komponent, shakl siz tomoningizdan joylashtiriladigan, Obyektlar Inspektori (Object Inspector) ning derazasida aksini topadi. Sizga ma'lumki, Object Inspector ikki betli — "Properties" (xususiyatlar) va "Events" (hodisalar). Delphi da dastur barpo etish deb komponentlarni shaklga (aytgandek, shakl o'zi ham komponent hisoblanadi) "tushirish" va ular orasidagi bog'lanishlarni sozlash ishlarini anglamoq kerak. Komponentlar orasidagi bog'lanishlarni sozlash quyidagicha amalga oshiriladi:

- shu komponentlar xususiyat qiymatlarini o'zgartirish orqali;
- hodisalarga adekvatli ta'sir yozish orqali.

Hodisalar to'g'risida batafsilroq keyingi bobda ko'rib chiqamiz, hozir esa fikrimizni xususiyatlarga jalb etamiz va, kerak bo'lganda, imkoniyat boricha, hodisalarga javob topishni aytib o'tamiz.

Xususiyat komponentning muhim atributi hisoblanadi, buni siz, ehtimol, anglab olishga ulgurgan bo'lsangiz kerak. Foydalanuvchiga (dasturchiga) xususiyat biror qiymatga ega bo'lgan strukturaning oddiy maydoni bo'lib ko'rinadi. Biroq, oddiy maydondan farqi shundan iboratki, biror xususiyat qiymatining o'zgarishi darhol shu komponentni vizual ko'rinishini o'zgarishiga olib keladi, chunki xususiyat o'z ichiga o'qish va yozish bilan bog'langan usullarni (harakatlarni) olgan. Ular, o'z navbatida, kerakli rasmni yangilashni ichiga olgan. Xususiyatlar ikki asosiy maqsadlarga xizmat qiladi. Birinchidan, ular shakl yoki komponentning tashqi ko'rinishini aniqlaydi. Ikkinchidan esa, xususiyatlar shakl yoki komponentning xatti-harakatini aniqlaydi.

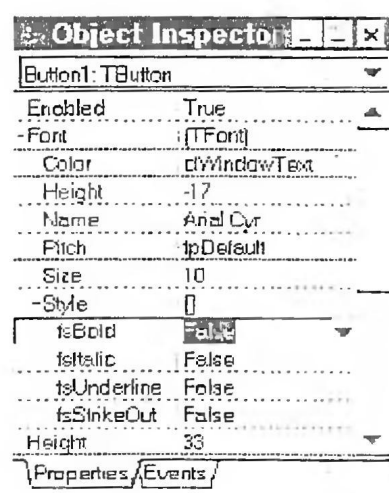
"Tabiatiga", ya'ni ichki tuzilishiga ko'ra, xususiyatlarning bir necha tiplari mavjud:



- oddiy xususiyatlar — ularga qiymatlari son yoki satr bo'lganlar kiradi. Masalan, Left va Top xususiyatlar qiymat sifatida komponentni yoki shaklni chap tomondagi yuqorisining holati butun sonlar beriladi. Caption va Name xususiyatlari satrni tashkil qiladi va komponent yoki shaklning sarlavha va nomini aniqlaydi.

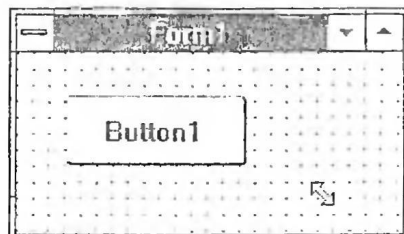
- tanlab olinadigan xususiyatlar — ularning qiymatlari taqdim etilgan to'plamdan (ro'yxatdan) olinishi mumkin. Eng oddiy misol — bu Boolean tipdagi xususiyat, qaysi True yoki False qiymatlarga ega bo'lishi mumkin.

- ichga kiritilgan xususiyatlar — ularni ichiga kiritilgan qiymatlar (yoki obyektlar) tashkil qiladi. Bunday xususiyatlarni Object Inspector ularning nomini chap tomonida "+" belgisini qo'yish orqali ajratadi. Bunaqa xususiyatlarning ikki xili bor: to'plamlar va kombinatsiyalangan qiymatlar. Object Inspector to'plamlarni kvadratli qavs ichiga olib ko'rsatadi. Agar to'plam bo'sh bo'lsa, uni [] holda beriladi. "To'plam" ko'rinishdagi ichiga kiritilgan xususiyatlar uchun o'rnatiladigan qiymat odatda Boolean tipli bo'ladi. Bunday xususiyatga misol sifatida keng tarqalgan o'z ichiga bulevli to'plamni olgan Style xususiyati ko'rsatish mumkin. Kombinatsiyalashtirilgan qiymatlar Obyektlar Inspektorida ayrim qiymatlar kolleksiyasi sifatida aks ettiriladi, ularni har birining berilganlari o'z tipiga ega bo'ladi (1.27-shakl). Ayrim xususiyatlar, masalan Font, o'z qiymatlarini o'zgartirilishiga dialogli derazachani chaqirish imkoniyati bor. Buning uchun Obyektlar Inspektorining shu xususiyatni ko'rsatayotgan qatorida uchta nuqtali kichkina tugmachani shiqillatib qo'yish kerak.



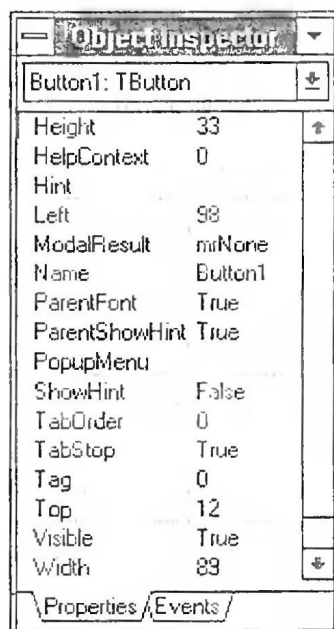
1.27-shakl.

Delphi komponentlar xususiyatlari bilan osonlik ravishda harakatlanishni loyihalash rejimida (design time) hamda dasturni bajarish rejimida (run time) ham o'tish mumkin.



1.28-shakl.

Loyihalash rejimida xususiyatlar bilan harakatlanish Shabl Dizayneri yordamida (Form Designer) yoki, siz ko'rganingizdek, Obyektlar Inspektorining "Properties" betide namoyon bo'ladi. Masalan, Height (balandligi) va Width (eni) xususiyatlarni o'zgartirish uchun, uni biror burchagini sichqon bilan ilintirib, kerakliha ikki tomonga surilsa yetarli bo'ladi. Agar Height va Width xususiyatlarning yangi qiymatlarini oddiy ravishda Object Inspector derazasida qo'yilsa, xuddi shu natijaga erishish mumkin.



1.29-shakl.

Boshqa tomondan, dasturni bajarish rejimida foydalanuvchi (dasturchi) Obyektlar Inspektorida aks ettirilgan xususiyatlar bilan harakatlanish imkoniga ega bo'lishdan tashqari, ularning kengroq ro'yxati bilan boshqarishi mumkin. Bu qanday bajarilishini keyingi bobda ko'rib chiqamiz.

### **1.5.1. BAJARISH REJIMIDA VIZUALLI KOMPONENTLARNING XUSUSIYATLARI USTIDA BOSHQARISH**

Bajarish rejimida komponentlarni xususiyat qiymatlarining barcha o'zgarishlari to'g'ridan-to'g'ri Paskal tilida kod satrini yozish orqali amalga oshirish mumkin. Bajarish rejimida Object Inspector bilan foydalanish imkoni yo'q. Biroq, komponentlar xususiyatiga kira olishni dasturli yo'li orqali oson ravishda olish mumkin. Biror xususiyatni o'zgartish uchun siz bajarishingiz kerak bo'lgan narsa — bu kodning oddiy satrini quyidagiga o'xshash ravishda yozish kerak bo'ladi:

```
MyComponent.Width := 35;
```

Yuqorida keltirilgan satr komponentning enini (Width) 35 qiymatli qilib o'rnatadi. Agar komponentning Width xususiyati mazkur dastur satrini bajarish paytigacha 35 ga tengmas bo'lsa, siz vizual ravishda komponent o'z enini o'zgartirishini kuzatasiz.

Shunday qilib, Obyektlar Inspektorida hech qanday sir yo'q. Object Inspector shunchaki loyihalash rejimida qulay ravishda bajariladiganlarni bajarish rejimida dasturli yo'li bilan amalga oshirish mumkin bo'ladi. Undan tashqari, yuqorida aytib o'tilgandek, komponentda Obyektlar Inspektorining derazasida aks etmaydigan xususiyatlar ham bo'lishi mumkin.

Delphi asosida yotgan obyektli-yo'nalgan Paskal tili asosli prinsipi sifatida vizualli komponentlarni taqdim etilayotganlarga moslikni mavjud etgan.

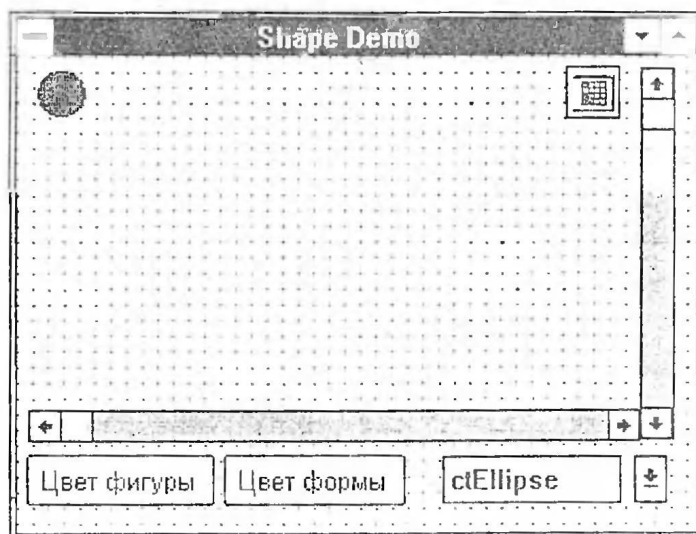
Delphi ni ishlab chiqaruvchilari o'z oldilariga shunday maqsad qo'yganlarki, agar, masalan, Button (tugmacha) komponentni taqdim etilishida biror kodni chiqaradi, qaysi tugmachani ekranda vizual ko'rinishiga mos ravishda olib keladi va uni klaviaturada topsangiz bo'ladi. Xuddi shu prinsipdan xususiyat tushunchasi paydo bo'ldi.

Agar siz Button komponentning Width va Height xususiyatlarini o'zgartirsangiz, tugmacha tegishli ravishda o'z eni va balandligini

o'zgartiradi. Width va Height xususiyatlarni o'zgartirilgandan keyin, sizga obyektga uni ko'rinishini qaytadan chizish kerakligi to'g'risida ko'rsatma berishning hojati yo'q, garchi oddiy dasturlashda siz xuddi shunday qilishingiz kerak edi. Aksi, ular bu berilganlarni "jonli" qiladilar va buning hammasi sizning ko'z oldingizda kelib chiqadi. Xususiyatlar sizga shunday xomxayolni beradiki, xuddi siz dasturli taqdim bilan duch kelmay, real obyekt bilan ish olib borayotgandeksiz.

1.30-shaklda tasvirlangan SHAPEDEM.DPR dasturi har xil usullarni namoyish qiladi, ular yordamida dasturni ishlashi orqali foydalanuvchining interfeysini o'zgartirish mumkin. Bu dastur qanday qilib interfeysni o'zgartirish orqali osongina "delfiyli" yondoshlarni barpo etish mumkinligini namoyish qilishdan tashqari biror foydali ish bajarmaydi.

SHAPEDEM dasturi faqatgina TShape obyektga ega. U, shaklda joylashgan ikki siljituvchi tasmacha bilan birgalikda bir necha tugmachalarga ega. Bu dastur shu bilan qiziqki, u TShape obyektning ishlash rejimida o'lcham, rang va uni ko'rinishini shaklni o'zinikiday o'zgartirish imkoniyati bor.

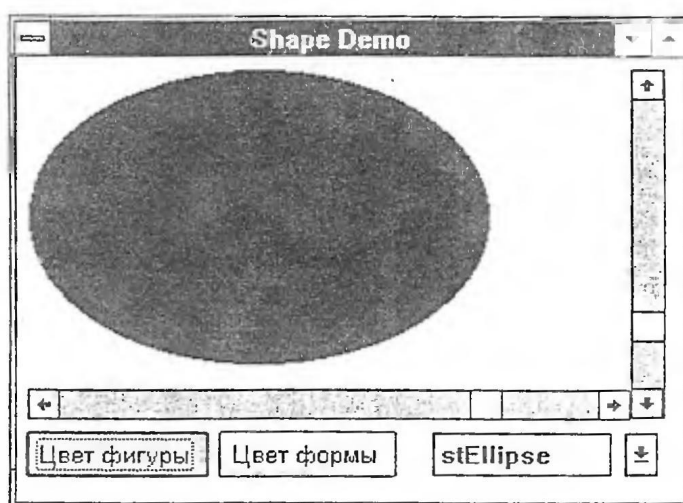


1.30-shakl. SHAPEDEM dasturi ikkita siljitish tasmachasi va bir necha tugmachaga ega.

Listing A SHAPEDEM dasturning kodini ko'rsatadi. Shu dasturni bosh modulining kodini qism-qism ravishda keltiramiz — ularni yozishni imkoniyati boricha.

*Listing A: SHAPEDEM.DPR dasturning dastlabki kodi.*

```
program Shapedem;  
uses  
Forms,  
Main in 'MAIN.Pas' {Form1} ;  
begin  
Application.CreateForm(Tform1, Form1) ;  
Application.Run ;  
end.
```



1.31-shakl.

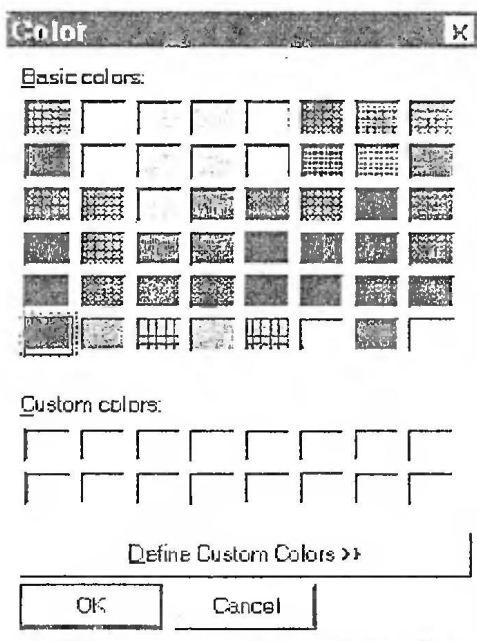
Bizning misolimizda siljituvchi tasmachalar (ScrollBars) ekran o'rta qismida tasvirlangan figuraning o'lchamini o'zgartirish uchun ishlatiladi, 1.31-shaklda ko'rsatilgandek. Figuraning yangi ko'rinishini tanlash uchun tushadigan ro'yxat (ComboBox) bilan foydalaning, figuraning yoki derazaning (shaklning) rangini o'zgartirish uchun esa, standartli dialog derazachasi ishlatiladi. Ular "Svet figuro" va "Svet formo" tugmachalar bilan chaqiriladi.

Foydalanuvchi (dasturchi) ishlash rejimida biror elementning yoki butun derazaning (shaklning) rangini o'zgartirish uchun nimalar qilish kerak? Buning uchun faqatgina bir necha harakatni bajarilsa yetarli bo'ladi.

O'zingiz bunga ishonch hosil qiling. Derazaning rangini o'zgartirish uchun komponentlar qatoridan (u "Dialogs" betida joylashgan) ColorDialog komponentini tanlab olib, uni shaklga joylashtiring. Bundan tashqari, shaklga odatdagi tugmachani ham joylashtiring

(komponent Button "Standart" betda turadi). O'qishga qulay bo'lishi uchun, Object Inspektor yordamida komponentni nomini (Name xususiyati) "Button1" dan (aytib o'tmaslik prinsipi bo'yicha u shunday nomlanadi) "FormColor" ga o'zgartiring, uning sarlavhasini (Caption xususiyati) esa — "Svet formo" ga. "Svet formo" tugmachasi bo'yicha ikki marotaba sichqon bilan shiqillating — Delphi quyidagi ko'rinishdagi usulning tayyorlovini generatsiyalashtiradi :

```
procedure TForm1.FormColorClick(Sender: TObject);
begin end;
Endi ikki qator oddiy kodni kiriting :
procedure TForm1.FormColorClick(Sender: TObject);
begin
If ColorDialog1.Execute then
Form1.Color :=ColorDialog1.Color ; end;
```



1.32-shakl.

Mazkur kod ishlash vaqtida "Svet formo" tugmachasini bosish orqali rang tanlash uchun standartli dialog derazachani 1.32-shaklda ko'rsatilgandek chiqaradi. Agar shu dialogli derachadagi OK tugmachani bossangiz, quyidagi satr bajariladi:

```
Form1.Color := ColorDialog1.Color;
```

Bu kod Form1 shaklning Color xususiyatini dialogli ColorDialog1 derazachadan tanlab olingan rangga o'rnatadi. Bu juda ham oddiy!!!

Xuddi shunday texnikani shaklning (Shape komponenti, Tshape obyektidan) rangini o'zgartirish uchun ishlatish mumkin. Siz bajaradigan hamma narsa — bu shaklga boshqa tugmachani o'rnatish kerak, xohish bo'lsa uni nomini "ShapeColor" ga,

sarlavhasini "Svet figuro" o'zgartirib, u bo'yicha sichqon bilan ikki marotaba shiqillatib quyidagiga o'xshash usulni barpo etasiz:

```
procedure TForm1.ShapeColorClick(Sender : Toject) ;
begin
if ColorDialog1.Execute then
```

```
Shape1.Brush.Color := ColorDialog1.Color ;  
end;
```

Bundan oddiyroq nima ham bo'lishi mumkin?!!

Yuqorida yozilgan kod o'z-o'zini hujjatlaydigan hisoblanadi, ya'ni dasturlashdan oz-moz xabari bo'lgan odam ortiqcha mehnatsiz bu satrlar nimalarni bajarishini tushunib oladi; agarda u bundan oldin hujjatnomani o'qib chiqsa, u holda u uchun umuman hamma narsa ochiq-oydin bo'ladi.

Barcha bu harakatlarni avtomatik ravishda ham bajarish mumkin, masalan, foydalanuvchining diqqatini biror harakatga jalb etish uchun shaklning ayrim elementi rangini o'zgartirish mumkin.

Windows xabarnomalarining barcha mexanizmi, bajarish vaqtidagi komponentlarni bir - biri bilan bog'lanishi dasturchiga yashirilgan ekan, bu bilan dasturni barpo etish jarayoni juda yengilashgan. Windows muhitida murakkab dasturlar barpo etish juda ham ko'p dasturchilarga hammabop bo'lib qoldi. Masalan, figuraning o'lchamlarini siljituvchi tasmalar yordamida o'zgartirishni dasturlanishi "sof" Windowsda "case" tipdagi konstruksiyasida xabarnomalarning murakkab ishlovi talab qilinadigan bo'lsa, Delphi da bitta-yagona kod satrini yozilsa kifoya bo'ladi.

Boshlash uchun, shaklga ScrollBar ning ikki komponentini joylashtiring ("Standart" betida joylashgan) va birinchi komponentning Kind xususiyatini sbHorizontal ga, ikkinchisini esa, sbVertical ga o'rnatish. Obyektlar Inspektorida "Events" betiga o'ting va OnChange hodisaning har bir siljitish tasmalari javob qaytarish uchun usulning tayyorlovini barpo eting. Usulni har birida quyidagicha bir qatorni yozing :

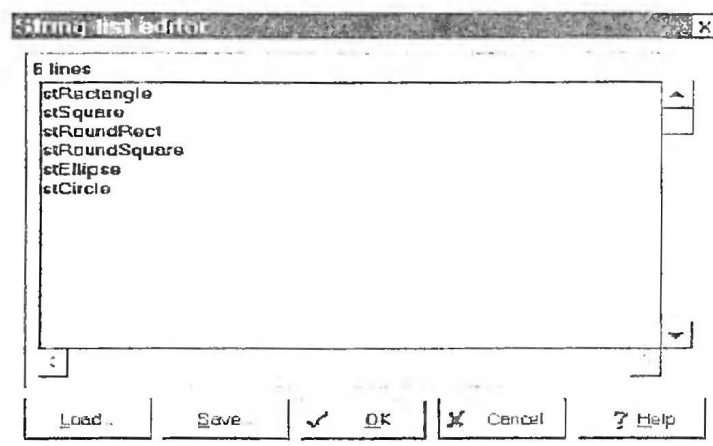
```
procedure TForm.ScrollBar1Change(Sender : TObject) ;  
begin  
Shape1.Width := ScrollBar1.Position * 3 ;  
end;  
procedure TForm.ScrollBar2Change(Sender : TObject) ;  
Shape1.Height := ScrollBar2.Position * 2 ;  
end;
```

Bu yerda ko'rsatilgan kod TShape figuraning Width va Height xususiyatlarini siljitish tasmalari "chopqirni" holatiga mos ravishda o'rnatadi (3 va 2 ko'paytkichlar faqat yaxshiroq taqdim etilishiga kiritilgan).



SHAPEDEM dasturning oxirgi qismi Delphi ni barpo etilishiga asos bo'lgan Object Pascal tilining katta imkoniyatlarini namoyish etadi. Siz ComboBox komponentining ro'yxatiga loyihalash hamda dasturni bajarish rejimlarida element kiritishingiz mumkin. Shu bilan birga loyihalash rejimida Items ro'yxatiga kerakli elementlarni oddiy ravishda kiritish uchun Obyektlar Inspektorining o'ng qismidagi qatorda shu (Items) xususiyatini ko'rsatayotgan uchta nuqtali kichkina tugmacha bo'yicha sichqon bilan shiqillatish kerak.

Sizning oldingizda matnli muharrirning (String List Editor) dialogli derazasi paydo bo'ladi, o'sha yerda siz elementlarni kiritasiz (1.33-shakl).



1.33-shakl.

Siz ko'rgan bo'lsangiz kerak, bu elementlar ro'yxating Shape1 (Shape) komponenti Shape xususiyatining opsiyalari ro'yxati bilan o'xshashligi bor. Boshqacha qilib aytiladigan bo'lsa, agar siz shaklda (oddiy ravishda uni sichqon bilan shiqillatib) komponent Shape1 ni ajratsangiz va Obyektlar Inspektorida Shape xususiyatiga qarasangiz, shu komponent qabul qiladigan figuralarni imkoniyatli ko'rinishlarining ro'yxatini ko'rasiz. Ularni, xuddi o'sha figuralar ko'rinishlarini, ComboBox1 komponentining ro'yxatida aytib chiqqandik. Mazkur ro'yxatni siz Delphi bo'yicha ma'lumotnoma to'plamidagi on-lineda topishingiz mumkin. Yoki, agar siz TShapeType sinfining dastlabki kodiga qarasangiz, o'sha yerda aytib o'tilgan TShapeType tipning barpo etuvchi xuddi o'sha elementlarini ko'rasiz:

```
TShapeType = (stRectangle, StSquare,
               StRoundRect,
               StRoundSquare, stEllipse, stCircle);
```

Shunday qilib, hamma aytilganlarning ma'nosi shundan iboratki, Delphi dasturining har bir obyektini orqasida Paskalda qandaydir kod to'g'ri keladi, "oddiy" dasturlashda ularga kira olish imkoniyatiga egasiz.. Sizdan hech narsa berkitilmagan. Bu degani, dasturni ishlash vaqtida tegishli kodni yozib, uning har qanday qismini ishlash harakatlarini o'zgartirishingiz mumkin bo'ladi. Bizning aniq hodisada, sizdan bir qator kodni yozish talab qilinadiki, u ishlaganda ComboBox1 ning tushirilgan ro'yxati bo'yicha foydalanuvchi tomonidan shiqillatilgan harakatga javobnoma bo'lsin. Bu javobnomaning kodini yozish uchun, loyihalash rejimida shaklda ComboBox1 komponentni ajrating (har qachongidek, unga sichqonni chap klavishasi orqali shiqillatib), keyin Obyektlar Inspektoridagi "Events" betiga o'ting. OnClick hodisani qarshisidagi bo'sh maydonga nisbatan ikki marotaba shiqillating. Muharrirda avtomatik ravishda usulning quyidagi tayyorlovi generatsiyalanadi.

```
procedure TForm1.ComboBox1Click(Sender: TObject);  
begin  
end;
```

Endi bir qator kodni kiriting, shunda usul quyidagi ko'rinishli bo'lsin:

```
procedure TForm1.ComboBox1Click(Sender: TObject);  
begin  
Shape1.Shape := TShapeType (ComboBox1.ItemIndex);  
end;
```

Bu kod qatori Shape1 komponentning Shape xususiyatini foydalanuvchi yoyilgan ro'yxatdan tanlab olgan ko'rinishda o'rnatadi. Ko'rsatilgan tiplarning tartibli a'zolari va ComboBoxda har xil elementlarning sonli qiymatlari mosligi tufayli bu kod ishlaydi. Boshqacha aytilganda, ko'rsatilgan tipning birinchi elementi 0 qiymatga ega, u ComboBoxda ko'rsatilgan birinchi elementiga mos keladi (1.33-shaklga qarang). Keling shu uslubni biroz batafsilroq ko'rib chiqamiz.

Agar siz chiqarilgan TShapeType tiplarning deklaratsiyasini ko'rib chiqsangiz, siz ko'rasizki, u "stRectangle" deb nomlangan. Belgilash tartibi bo'yicha, kompilyator bu elementga tartib raqamini 0 deb belgilaydi. Keyingi tartibdagi elementga 1 raqamni belgilaydi va h.k. Shunday qilib, "stRectangle", "stsquare" va h.k. so'zlar, haqiqatda, shu chiqarilgan tiplarda shunchaki tartib raqamlarni simvolik alomatini ko'rsatadi. ComboBox ro'yxatidagi elementlariga ham 0 dan boshlangan tartibli raqamlari bo'yicha dalil qilish mumkin. Xuddi shuning uchun

(shu holatda) ko'rsatilgan satrlarni TShapeType tipni deklaratsiyasidagilarga qat'iy moslikda kiritish kerak. Shunday qilib, "TShapeType (ComboBox1. ItemIndex)" tipni tubdan o'zgarishidan foydalanib, siz kompilyatorga Combobox elementlari va TShapeType tiplari orasidagi umumiylikni xabar qilasiz: ayni, ularni tartib raqamlarini.

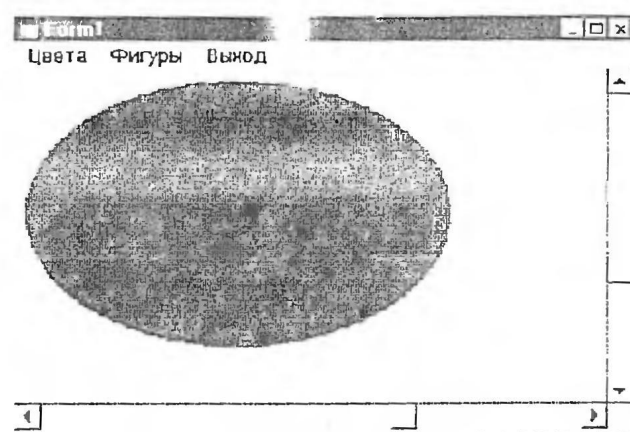
Shunday qilib, ko'rib turibsizki, Delphi juda ham eguvchan va quvvatli dasturli vosita hisoblanadi, u bilan siz tezda o'z dasturingizning mantiqini amalga oshirishingiz mumkin va taqdim etilgan yondoshlar bilan to'la foydalanish imkoniyati bor.

### 1.5.2. SHAPEDEM2 DASTURI

SHAPEDEM dasturni yozilishi hamda uni o'zlashtirilishi juda ham oddiy. Biroq, foydalanuvchi tomonidan deraza o'lchami o'zgartirilsa, u "ko'rimsiz" bo'lib qoladi. Keling uni shunday ravishda o'zgartiraylikki, dastur deraza o'lchami o'zgarishiga o'zi ishlov bersin, bir yo'la menyu komponentini o'rganamiz. Bu maqsadga erishish uchun quyidagilarni bajaramiz:

- Tugmachalar va yoyiladigan ro'yxatni ekrandan olib tashlaymiz va ularning o'miga shaklga menyu komponentini (MainMenu) o'rnatamiz.
- Siljituvchi tasmlarni o'z holatlarini deraza o'lchamiga qarab o'zgarishini "majburlaymiz".
- Siljituvchi tasmaning Position xususiyatini shunday o'zgarishga "majburlaymizki", u shaklning o'lchamini to'g'ri aks etirsin.

Shu o'zgarishlardan keyin dastur qanday ko'rinishli bo'lishini 1.34-shaklga qarab, bilish mumkin.



1.34-shakl.

SHAPEDEM2 dasturi foydalanuvchi tomonidan deraza o'lchamini o'zgartirilishiga ta'sir ko'rsatish imkoniyatiga ega.

V Listingi: SHAPEDEM dasturi FormOnResize uslubini kiritgan. Bosh moduli taqdim etiladi.

```
unit Vain ;
interface
uses
WinTypes, WinProcs, Graphics, Forms, Controls, ColorDlg,
StdCtrls, Menus, Dialogs, ExtCtrls ;
Type TForm1 = class(Tform)
ColorDialog1 : TcolorDialog ;
ScrollBar1 : TscrollBar ;
ScrollBar2 : TscrollBar ;
MainMenu1 : TmainMenu ;
Shapes1 : TMenuItem ;
ShapeColor1: TMenuItem ;
FormColor1 : TMenuItem ;
Shapes2 : TMenuItem ;
Rectangle1 : TMenuItem ;
Square1 : TMenuItem ;
Rect1 : TMenuItem ;
RoundRect1 : TMenuItem ;
RoundSquare1 : TMenuItem ;
Ellipes1 : TMenuItem ;
Circle1 : TMenuItem ;
Exit1 : TMenuItem ;
procedure NewShapeClick(Sender: Tobject);
procedure ShapeColorClick(Sender: Tobject);
procedure FormClorClick(Sender: Tobject);
procedure ScrollBar2Change(Sender: Tobject);
procedure ScrollBar1Change(Sender: Tobject);
procedure Exit1Click(Sender: Tobject);
private
{Private declarayions}

                Public
                {Publice declarayions}

end;
```

```

var Form1: TForm1;
implementation
{$R (.DFM)}
procedure TForm1.NewShapeClick(Sender: TObject);
begin
Shape1.Shape := TshapeType((Sender as TMenuItem).Tag);
end;
procedure TForm1.ShapeClick(Sender: TObject);
begin
if ColorDialog1.Execute then
Shape1.Brush.Color := ColorDialog1.Color ;
end;
procedure TForm1.FormColor Click(Sender: TObject);
begin
if ColorDialog1.Execute then
Form1.Color := ColorDialog1.Color ;
end;
procedure TForm1.ScrollBar2Change(Sender: TObject);
begin
Shape1.Height := ScrollBar2.Position ;
end;
procedure TForm1.ScrollBar1Change(Sender: TObject);
begin
Shape1.Width := ScrollBar1.Position;
end;
procedure TForm1.FormResize(Sender: TObject);
var Menu, Caption, Frame : Integer ;
begin
Caption := GetSystemMetrics (sm_cyCaption);
Frame := GetSystemMetrics(sm_cxFrame)*2 ;
Menu := GetSystemMetrics (sm_cyCaption);
Scrollbar1.Max := Width ;
Scrollbar2.Max := Height ;
Scrollbar2.left := Width - Frame - Scrollbar2.Width;
Scrollbar1.Top := Height - ScrollBar2.Width -Frame - Caption - Menu;
Scrollbar1.Width := Width - Scrollbar2.Width - Frame;

```

```

Scrollbar2.Height: = Height - Frame - Caption - Menu -
Scrollbar.Height;
end;
procedure TForm1.Exit1Click(Sender: TObject);
begin
Close;
end;
end.

```

Dastur uchun bosh menyu MainMenu yordamida barpo etiladi (komponentlar qatoridagi "Standart" betda joylashgan). Uni shaklga joylashtirib, sichqon bilan ikki marotaba shiqillating — menyuning muharriri ochiladi, u yerda sizga kerakli menyu punktlarini kiritishingiz mumkin va, agarda bunga xohish bo'lsa, ularning nomlarini o'qilishi oson bo'lgan nomlarga o'zgartirishingiz mumkin (Delphi tomonidan aytmasdan o'tmaslik prinsipi bo'yicha belgilanganlarni). SHAPEDEM2 dasturining menyusini uchta bosh punktlari bilan barpo etamiz: "Sveta" (ranglar), "Фигуры" (figuralar), "Выход" (Chiqish).

*Birinchi punkt uchun quyidagi ichki punktlarni barpo etamiz:*

- figuraning rangi
- derazaning rangi.

*Ikkinchisiga:*

- to'g'ri to'rtburchak
- kvadrat
- yumaloqlangan to'g'ri to'rtburchak
- yumaloqlangan kvadrat
- ellips
- aylana.

Menyuning uchinchi punkti hech qanaqa ichki punktlarga ega bo'lmaydi.

Menyuning barcha punktlarini va ichki punktlarini barpo etib bo'lgandan keyin SHAPEDEM2 dasturni ishlashi uchun figuraning tipi bog'langan har bir ichki punktlariga tartib raqamlari belgilanishi kerak. Buning uchun menyuni har bir punktida mavjud bo'lgan Tag xususiyatidan foydalanamiz. Tag xususiyati (Integer tipli) Delphi ning har bir komponentiga ataylab kiritilganki, dasturchilar ularni o'z ixtiyori bilan mumkin bo'lishi uchun. "to'g'ri to'rtburchak" punktning Tag

xususiyatiga 0 raqami qilib tayinlaymiz, 1 ni — "kvadrat" punktiga, 2 ni — "yumaloqlangan to'g'ri to'rtburchak" punktiga va h.k. Bunday qilib tayinlashning maqsadini keyinroq tushuntiriladi.

Ikki usul, ranglarni o'zgartirish uchun ichki punktlarda barpo etilganlar SHAPEDEM dasturdagilarga mos keladi:

```
procedure TForm1.ShapeColorClick(Sender: TObject);
begin
  if ColorDialog1.Execute then
    Shape1.Brush.Color := ColorDialog1.Color;
end;
procedure TForm1.FormColorClick(Sender: TObject);
begin
  if ColorDialog1.Execute then
    Form1.Color := ColorDialog1.Color;
end;
```

Ko'rib turganingizdek, dasturning birinchi versiyasiga nisbatan hech narsa o'zgargani yo'q, garchi bu usullar tugmachalar orqali emas, menyudan chaqirildi.

Shuning singari, figuraning ko'rinishini o'zgartirish uchun menyuning ichki punktlaridan tanlashga javobni amalga oshiruvchi usullar tushiriladigan ro'yxatdan figurani tanlash usuliga juda ham o'xshab ketadi:

```
procedure TForm1.NewShapeClick(Sender: TObject);
begin
  Shape1.Shape := TshapeType((Sender as TMenuItem).Tag);
end;
```

Bu kod ko'rsatib o'tiladigan TshapeType tipi dastlabki qiymat sifatida 0 bo'lgani tufayli to'g'ri "ishlaydi" va Tag xususiyatidagi ichki menyusida ham biz tartibli raqamlarni 0 dan boshlab yozib chiqdik.

Qayd qilamiz, biz as operatorini ishlatdik, u tiplarni biridan boshqasiga ishonchli ravishda o'zgartiradi: chunonchi, Sender (umumiy TObject tipiga ega) parametrini TmenuItem tipiga o'zgartirish. Odat bo'yicha, Delphi da Sender — bu boshqaruvchi element, u tarkibiga kirgan funksiyaning xabarlarini jo'natadi. Berilgan holatda, Sender menyuning punkti hisoblanadi va, demak, siz bu parametr bilan xuddi TmenuItem tipli qilib berilgan deb ishlayverasiz.



Operator as ni ishlatishdan asosiy maqsad shundan iboratki, u juda aniq sintaksisni ta'minlaydi, hattoki siz tiplarning o'zgartirilishini murakkab ikki saviyali tarzda o'tkazsangiz ham. Bundan tashqari, as dan dasturni ishlash rejimida o'zgartirishlarni tekshiruvini ta'minlaydi. Qachonki siz operator as bilan foydalanayotgan bo'lsangiz, siz shunga amin bo'lishingiz mumkinki, Sender ning TmenuItem ga o'zgartirilishi real ravishda bajariladi, agarda haqiqatda Sender TmenuItem tipiga ega bo'lsa.

SHAPEDEM2 dasturida siljituvchi ikkita tasmachalar ol'chami qanday bo'lishidan qat'li nazar har doim deraza chegarasi yonida joylashadi. Bu harakatlar bajarilishi uchun sizdan oldingilarga nisbatan murakkabroq bo'lgan dasturni yozish talab qilinadi. Ilgari ko'rsatilgandek, Delphi dasturchidan Windows-dasturlashning detallarini ko'rsatmasa ham, biroq u Windows API (amaliy foydalavuvchining interfeysi) funksiyalariga murojaat etishni man etmaydi.

Shunday qilib, Delphi past saviyali dasturlanishni Windows API darajasida ushlaydi. Qisqasi, agar sizga past saviyali dasturlanishning changalzoriga kirish kerak bo'lsa — marhamat !

```
procedure TForm1.FormResize(Sender: TObject);
var Menu, Caption, Frame : Integer;
begin
Caption := GetSystemMetrics(sm_cyCaption);
Frame := GetSystemMetrics(sm_cxFrame) * 2;
Menu := GetSystemMetrics(sm_cyMenu);
Scrollbar1.Max := Width;
Scrollbar2.Max := Height;
Scrollbar2.Left := Width - Frame - Scrollbar2.Height;
Scrollbar2.Height := Height - Frame - Caption - Menu;
Scrollbar1.Top :=
Height - Scrollbar2.Width - Frame - Caption - Menu;
Scrollbar1.Width := Width - Scrollbar2.Width - Frame;
end;
```

Yuqorida ko'rsatilgan kod, OnResize hodisasiga javob hisoblanadi. Bu hodisa Obyektlar Inspektorining "Events" betida boshqalar bilan birgalikda shakl (deraza) tanlash holatida keltirilgan. Siz kutganingizdek, OnResize hodisasi (xabarnomasi) shaklga (derazaga) har doim yuboriladi, qachonki foydalanuvchi sichqon bilan derazaning biror chekkasiga

"ilintirsa" va uning o'lchamini kattalashtirish yoki kichiklashtirishga urinsa. Biroq, bu xabarnoma (hodisa) deraza maksimalizatsiyalashtirilganda (minimizatsiyalashtirilganda emas) ham derazaga yuboriladi.

Bu usulda birinchi qilinadigan vazifa — tizimli parametrlar so'raladi, ular derazaning doira va menyusidagi sarlavhalarning o'lchamini bildiradi. Bu axborotga Windows API ning qismi bo'lgan `GetSystemMetrics` funksiyasini chaqirish orqali erishiladi. `GetSystemMetrics` funksiyasiga konstanta sifatida bir argument yuboriladi, u yuboriladigan axborotning ko'rinishini aniqlaydi. Masalan, agar siz funksiyaga `sm_cyCaption` konstantani yuborsangiz, javob sifatida deraza sarlavhasining balandligini pikselada olasiz. Bunday konstantalar to'la ro'yxati Delphi ning ma'lumotnomalar to'plamidagi (Help|Windows API|Alphabetical functions|User functions|GetSystemMetrics) on-Line da bor, bu yerda esa, shu ma'lumotnomadan qisqacha ko'chirma keltiramiz:

`SM_CXBORDER` — derazani chetlab o'tadigan ramkaning eni, qaysini o'lchamini o'zgartirish mumkin emas.

`SM_CYBORDER` — derazani chetlab o'tadigan ramkaning balandligi, qaysini o'lchamini o'zgartirish mumkin emas.

`SM_CYCAPTION` — deraza sarlavhasining balandligi, derazani chetlab o'tadigan ramkaning balandligi qo'shilgan holda, qaysini o'lchamini o'zgartirish mumkin emas.

`SM_CXCURSOR` — kursorning eni.

`SM_CYCURSOR` — kursorning balandligi.

`SM_CXFRAME` — derazani chetlab o'tadigan ramkaning eni, qaysini o'lchamini o'zgartirish mumkin.

`SM_CYFRAME` — derazani chetlab o'tadigan ramkaning balandligi, qaysini o'lchamini o'zgartirish mumkin.

`SM_SXFULLSCREEN` — to'la ekranli derazani kliyent qismining eni.

`SM_CYFULLSCREEN` — to'la ekranli derazani kliyent qismining balandligi (ekran balandligidan deraza sarlavhasining balandligini ayirganga ekvivalent).

`SM_CXICON` — ikonkalarining eni.

`SM_CYICON` — ikonkalarining balandligi.

`SM_CYMENU` — bir qatorli menyu tasma-sining balandligi. Bu balandlik derazani chetlab o'tadigan ramkani balandligidan ayirgan

qismi, qaysini o'lchamini o'zgartirish mumkin emas (SM\_CYBORDER).

SM\_CXMIN — derazaning minimal eni.

SM\_CYMIN — derazaning minimal balandligi.

SM\_CXSCREEN — ekranning eni.

SM\_CYSCREEN — ekranning balandligi.

SM\_MOUSEPRESENT — agar sichqon o'rnatilmagan bo'lsa, 0 dan boshqasi.

FormResize usulida dastur siljituvchi tasmalarning yangi o'lchamini hisoblaydi :

Scrollbar1.Max := Width ;

Scrollbar2.Max := Height ;

Scrollbar2.left := Width - Frame - Scrollbar2.Width ;

Scrollbar1.Height := Height - Frame - Caption - Menu ;

Scrollbar1.Top :=

Height - Scrollbar2.Width - Frame - Caption - Menu ;

Scrollbar1.Width := Width - Scrollbar2.Width - Frame ;

Bu yerda keltirilgan hisoblar oddiy matematik amallardan iborat. Masalan, vertikalga siljituvchi tasmaning chap tomoni butun ekranni (shaklni) enidan ramka va tasmaning enini ayirish kerak. Bu elementarli mantiq va uni dasturda amalga oshirib, biz siljituvchini vertikal tasmasini hosil qilamiz, derazani (shaklni) o'ng chekkasining yonida har doim joylashgan.

SHAPEDEM dasturida har bir tasmani Max xususiyatining qiymati aytib o'tmaslik prinsipi bo'yicha 100 teng bo'lib qolaverdi ; bu bildiradiki, tasmaning chopuvchisi imkoniyatli masofani (ham vertikal, ham gorizontal siljituvchi tasmalar uchun) o'tib bo'lgandan keyin Position xususiyati 100 o'rnatilgan bo'ladi. Agar chopuvchi boshga qaytib kelsa, Position xususiyati Min xususiyatiga teng bo'lib o'rtnashadi, u aytmaslik prinsipi bo'yicha 0 ga tengdir.

SHAPEDEM2 dasturida siz Min va Max xususiyatning qiymatlarini shunday o'zgartirishingiz mumkinki, siljituvchi tasmalarning Position qiymatlar diapazoni derazani (shaklni) joriy o'lchamini aks etsin, hattoki ishlash rejimida shakl o'z o'lchamini o'zgartirsa. Bu yerda FormResize usulidan tegishli satrlar keltirilgan.

```
procedure TForm1.FormResize(Sender : TObject);  
begin
```

```

...
Scrollbar1.Max := Width ;
Scrollbar2.Max := Height ;
...
end;

```

Yuqorida ko'rsatilgan ikki qator kod, oddiy ravishda siljituvchi tasmalarni maksimal qiymatlarini shaklni eniga va balandligiga mos ravishda o'rnatadi. Shundan keyin siz har doim shaklga o'rnatilgan figurani shaklning o'ziday qilib kattalashtirishingiz mumkin bo'ladi. Bunday o'zgartirishlar kiritilgandan keyin Position xususiyatini biror ko'paytuvchiga ko'paytirishni hojati qolmaydi.

```

procedure TForm1.Scrollbar2Change(Sender : TObject);
begin
  Shape1.Height := Scrollbar2.Position ;
end;

```

Siz shundan keyin SHAPEDEM2 dasturni ishlashga qo'ysangiz, shaklning o'lchami har qanday o'zgarishida ko'rasizki dastur benuqson ishlaydi. Undan tashqari, endi siz menyudan figura va ranglarni tanlab olishingiz mumkin, ular dasurga qat'iyroq ko'rinishni beradilar.

Oxirida bir kichik mulohaza kiritish lozim bo'ladi. Balki, siz buni sezgan bo'lsangiz kerak, aytib o'tmaslik prinsipi bo'yicha, har bir shakl ikkita siljituvchi tasmachaga ega (HorzScrollbar va VertScrollbar), ular har doim avtomatik ravishda paydo bo'ladilar, qachonki, shaklning o'lchami shu shaklda joylashgan boshqaruvchi elementlar bilan band qilingan doiradan kichik bo'lib qolsa. Ayrim paytda bu siljituvchi tasmalar juda ham foydali bo'lishi mumkin, lekin bizning hodisada ular umuman biz xohlamagan narsani ham berishi mumkin. Shu tufayli, ishonchlilik uchun, siz ularning ichki Visible xususiyatini False ga o'rnatishingiz mumkin.

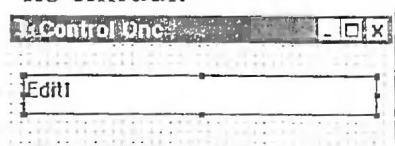
## 1.6. DELPHI DA USULLAR

Delphi ni to'la tushunish va hamma ustunlik tomonlarini sezish uchun, siz albatta Object Pascal tilini yaxshi o'zlashtirib olishingiz kerak. Yaxshi dasturchi qachon qo'lda kodlar yozish texnikasini yaxshi tushunib olgan bo'lsa, Delphi vizualli qismi boy hisoblanadi.

Bu bobning mavzularini muhokama qilish davrida biz bir necha oddiy misol ko'rib chiqamiz, ular Windows ni muhim boshqaruvchi elementlari bilan foydalanish texnikasini namoyish etadilar.

### 1.6.1. VIZUALLI VOSITALAR YORDAMIDA USULLAR BARPO ETISH

Oldingi ta'riflarimizda siz ko'rdingizki, usulning sintaktik "skeleti" vizuallari vositalar yordamida generatsiyalashtirilishi mumkin. Buning uchun, eslatib o'tamiz, Obyektlar Inspektorida sichqon bilan sizni qiziqtirayotgan hodisada kerakli komponentning sarlavhasi ro'parasidagi bo'sh satr bo'yicha ikki marotaba shiqillatish kerak bo'ladi. E'tibor beramiz, agar bu satr bo'sh bo'lmasa, u holda ikki marotabali shiqillatish sizni qayerda mazkur usul joylashgan Redaktor derazasiga ko'chiradi.



1.35-shakl



Bundan keyingi bayonimizni chuqurroq tushunish uchun dasturlashning obyektli qilib yo'llantirilgan konsepsiya to'g'risida qisqacha to'xtalib o'tamiz. Boshida obyektli qil-ib yo'llantirilgan dasturlashning asosli bo'lgan sinf(klass) tushunchasini aniqlaymiz. Sinf — bu Obyektlar kategoriya-si, ular bir xil xususiyat va harakatlarga egadirlar. Shunda obyekt biror sinfning oddiy nusxasi sifatida taqdim etiladi. Masalan, Delphi da "forma" (deraza) tipi sinf hisoblanadi, bu tipning o'zgaruvchisi esa - obyekt bo'ladi. Usul — bu sinfning qismi bo'lgan protsedura. Usullar sinflarni maydon va xususiyatlari bilan harakatlanadi (ixtiyoriy o'zgaruvchilari bilan ishlashi mumkin bo'lsa ham) va o'z sinfidagi ixtiyoriy maydon va usullarga avtomatik ravishda kira olishga ega.

Boshqa sinflarning maydon va usullariga kira olish imkoniyati ularni "himoyalangan" saviyasiga bog'liqdir. Hozircha sizga eng muhimi shuki, iboratki, usullarni vizuallari vositalar hamda kodni qo'lda yozish orqali ham barpo etish mumkin.

Keling, CONTROL1 dasturini barpo etish jarayonini ko'rib chiqamiz, u Delphi da usullar yozish texnikasini o'rganishga yordam beradi.

CONTROL1 dasturini barpo etish, sichqon yordamida Edit komponentini (Komponentlar qatoridagi "Standart" betida joylashgan) 1.35-

shaklga joylashtiring. Shundan keyin sizning shaklingiz 1.35. da ko'rsatilgandek bo'ladi.

Endi Object Inspector iga o'ting, "Events" betini tanlang va

OnClick hodisaning to'g'risidagi bo'sh satr bo'yicha 1.36-shaklda ko'rsatilgandek ikki marotaba sichqon bilan shiqillating. Shundan keyin faollashgan Redaktor derazasida EditOnClick usulning generatsiyalashgan "skeletini" OnClick hodisasiga javob sifatida chiqishini ko'rasiz.

```
procedure TForm1.EditOnClick(Sender: TObject);
begin
end;
```

Protsedura generatsiyalashgandan keyin siz uning nomini, Delphi "o'rnatganday", qoldirishingiz yoki ixtiyoriy boshqaga uni o'zgartishingiz mumkin ( buning uchun oddiygina yangi nomni Obyektlar Inspektorining yuqorida ko'rsatgan satriga kiriting va Enter ni bosib qo'ying).

Endi kodni Redaktorining derasida usulning ma'noli qismini kiriting:

```
procedure TForm1.EditOnClick(Sender: TObject);
begin
  Edit1.Text := 'Siz muharrir satrida ikki marotaba shiqillatingiz';
end;
```

Dasturni saqlab qoling. Ishlash vaqtida muharrir satrida ikki marotaba shiqillating. Matn bu satrda EditOnClick usulida biz yozganga mos o'zgaradi: 1.37-shaklga qarang.

1.37-shakl. TEdit elementning boshqaruvchisi nisbatan ikki marotaba shiqillatilsa, uning ichidagi o'zgaradi.

A - listingi: CONTROL1 dasturi qanday qilib Delphi da usullarni barpo etish va ular bilan foydalanishni namoyish etadi.

```
program Control1;
uses Forms, Main in 'MAIN.PAS' {Form1} ;
begin
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.
```

V - listingi: CONTROL1 dasturning bosh moduli.

```
unit Main;
interface
```



```

uses WinTypes, WinProcs, Classes, Graphics, Controls,
Printers, Menus, Forms, StdCtrls ;
type
TForm1 = class(TForm)
Edit1: TEdit;
procedure EditDbClick(Sender : TObject) ;
end;
var Form1 : TForm1;
    implementation
{$R *.DFM}
Procedure TForm1.Edit1DbClick(Sender : TObject );
begin
Edit1.Text:= 'Siz muharrir satrida ikki marotaba shiqillatingiz';
end;
end.

```

Sizning dasturingiz tezkor xotiraga yuklangandan keyin CONTROL1.DPR dagi kompilyator tomonidan avtomatik ravishda generatsiyalashtirilgan ikki qator kod bajariladi :

```

Application.CreateForm(Tform1, Form1);
Application.Run ;

```

Birinchi qator operatsion tizimdan tezkor xotirani so'raydi va u yerda TForm1 sinfining nusxasi bo'lgan Form1 obyektini barpo etadi. Ikkinchi qator "aytib o'tmaslik prinsipi bo'yicha" Delphida e'lon qilingan Application obyektiga yondoshning bosh shaklini bajarilishiga qo'yishni ko'rsatma beradi. Bu yerda biz TApplication sinfi va avtomatik ravishda barpo etiladigan Application uning nusxasi to'g'risida batafsil to'xtalib qolmaymiz. Eng muhimi shuni tushunib olish kerakki, uni tayinlashni asosiysi — sizning dasturingizni boshqaruvchi biror yadrosi bo'lib xizmat qiladi.

Odatda, biz siz bilan ko'radigan misollarimizda, .DPR loyihalarning fayllari deyarli bir xil. Shuning uchun bundan keyin ular bir-biridan tubdan farq qilishmasa, biz ularning matnini keltirmaymiz. Bundan tashqari, Delphi tomonidan avtomatik ravishda generatsiyalashadigan .DPR fayliga ko'p holatlarda qarashning hojati yo'q, chunki u bajaradigan hamma amallar standart hisoblanadi.

Demak, biz ko'rib turibmizki, Delphida kodlarning ko'pchiligi avtomatik ravishda generatsiyalashadi. Yondoshlarning ko'pchiligida siz



bajaradigan ishlar — bu bitta yoki bir necha kod satrlarini kiritishingiz kerak.

Edit1Db1Click usulidagi singari:

Edit1.Text :q 'Siz muharrir satrida ikki marotaba shiqillatingiz' ;

CONTROL1 dasturning tashqi interfeysi yetarli darajada oddiy bo'lsa ham, u (dastur) qat'iy ichki strukturaga ega. Delphidagi har bir dastur .DPR kengaytmali loyihaaning fayllaridan va bitta yoki bir necha .Pas kengaytmali modullardan iborat bo'ladi. Loyihaaning bosh shakliga ega bo'lgan modulni bosh modul deb nomlanadi. Modullar orasidagi bog'lanishni kompilyatorga ko'rsatma beruvchisi Uses gapi hisoblanadi, u modullar qaramligini aniqlaydi.

Siz tomondan Redaktorda barpo etilgan va Delphida avtomatik ravishda generatsiyalashgan modullar orasida hech qanday funksional farq yo'q. Har bir holatda modul uchta seksiyaga bo'linadi:

- Sarlavha
- Interface seksiyasi
- Implementation seksiyasi

Shunday qilib, modulning "skeleti" quyidagicha ko'rinishli bo'ladi:

unit Main ; (modulning sarlavhasi)

interface (Interface seksiyasi)

implementation (Implementation seksiyasi)

end.

Interfeysli (interface) seksiyasida boshqa modullarga ma'lum bo'lishi kerak bo'lgan barcha ma'lumotlar ta'riflanadi (tiplar, o'zgaruvchilar, sinflar, konstantalar, protseduralar, funksiyalar). Implementation seksiyasida sinflar, protsedura yoki funksiyalarni amalga oshiruvchi kodlar joylashtiriladi.

### 1.6.2. PARAMETRLARNI UZATISH

Delphida protsedura va funksiyalarga ( demak, sinflar usuliga ham) ularning ishlashiga kerakli bo'lgan axborot bilan ta'minlash uchun parametrlar uzatilishi mumkin. PARAMS dasturi Delphi usullariga qanday qilib parametrlar uzatilishidan foydalanishni namoyish etadi. Bundan tashqari, biz bilib olamizki, qanday qilib:

- o'z xususiy protseduralarni barpo etish;
- sinf usuli barpo etish uchun protseduralarni sinflarga qo'shish;
- bir potsedurani ikkinchi protseduradan chaqirish.

PARAMS dasturi sizga tahrir qilish satriga jumlarlar kiritish imkoniyat beradi.

"Вызов процедуры WriteALL" tugmachasi bosilgandan keyin boshqaruvchi element EditSource dagi satr oltita boshqaruvchi elementlarga — tahrirlash satrlariga 10.38-shaklda ko'rsatilgandek ko'chiriladi.

Bundan keyin biz komponentlarni qanday qilib shaklga joylashtirish kerakligi to'g'risida batafsil to'xtalib o'tirmaymiz, chunki siz buni allaqachon bilasiz. Shaklga yettita Edit komponentlarini joylashtirganizdan keyin, Obyektlar Inspektori yordamida yettinchi (Edit7) komponentning nomini EditSource ga o'zgartiring. Komponent Buttonni shaklga qo'ying va Object Inspectorida uni sarlavhasini (Caption xususiyatini) "Вызов программы WriteALL"ga o'zgartiring (albatta siz uning shrifti, rangi va h.k. larni o'zgartirishingiz mumkin).

Shaklni loyihalash tugatilgandan keyin TForm1 quyidagi ko'rinishda bo'ladi:

```
TForm1 = class(TForm);  
  Edit1 : TEdit ;  
  Edit2 : TEdit ;  
  Edit3 : TEdit ;  
  Edit4 : TEdit ;  
  Edit5 : TEdit ;  
  Edit6 : TEdit ;  
  EditSource : TEdit ;  
end;
```

Navbatdagi qadam foydalanuvchi tomonidan Button1 tugmachasini bosish orqali usulni qo'shishdan iborat. Buni, eslatib o'tamiz, ikki xil yo'l bajarish mumkin :

- Obyektlar Inspektorida "Events" betiga o'tib (dastavval shaklda Button1 komponentini tanlab), OnClick so'zini tanlab va sichqon bilan undan o'ng tarafdagi bo'sh satr bo'yicha shiqillatish kerak.

- Oddiy ravishda shakldagi Button1 komponenti bo'yicha ikki marotaba shiqillatish lozim.

```
Delphi quyidagi "tayyorlovni" generatsiyalashtiradi :  
procedure TForm1.Button1Click(Sender: TObject);  
begin  
end;
```

PARAMS dasturning maqsadi — sizni protseduralarni yozish va ularda parametrlar uzatishni o'rgatadi. Xususan, PARAMS dasturi Button1 tugmachasini bosilishiga WriteAll protsedurasini chaqirish va unga parametr sifatida tahrirlash satridagi EditSource (EditSource.Text)ni uzatish bilan javob qaytaradi.

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
  WriteAll(EditSource.Text);  
end;
```

Shuni tushunish muhimki, EditSource obykti Tdit sinfining nusxasi hisoblanadi va, demak, Text xususiyatli bo'lib, tahrirlash satrida terilgan matnga egadir. Ehtimol, siz payqab olishga ulgurgan bo'lsangiz kerak, aytib o'tmaslik prinsipi bo'yicha Text xususiyati ega bo'lgan qiymat Name komponentining nomiga to'g'ri keladi — bizning holatda "EditSource"G' Text xususiyatini siz shubhasiz, loyihalash rejimida hamda bajarish paytida ham tahrirlashingiz mumkin bo'ladi.

Oltita tahrirlash satrlarida aks etadigan matn WriteAll protsedurasiga parametr sifatida uzatiladi. Protseduraga parametr uzatish uchun, oddiy ravishda shu protseduraning nomini yozing va uzatiladigan parametr (parametrlar)ni qavs ichiga oling — mana bunday :

```
WriteAll (EditSource.Text);
```

Bu protseduraning sarlavhasi quyidagi ko'rinishli bo'ladi :

```
procedure TForm1.WriteAll(NewString : String);
```

Bu yerda protseduraga uzatiladigan NewString parametri String tipli bo'lishi shart.

Eslaymiz, WriteAll protseduraning vazifasi shundan iboratki, u EditSource tahrirlash satridagilarni oltita Edit1-Edit6

```
procedure TForm1.WriteAll(NewString : String);
```

```
begin
```

```
Edit1.Text :q NewString ;
```

```
Edit2.Text :q NewString ;
```

```
Edit3.Text :q NewString ;
```

```
Edit4.Text :q NewString ;
```

```
Edit5.Text :q NewString ;
```

```
Edit6.Text :q NewString ;
```

```
end;
```

WriteAll protsedurasi Delphida biror hodisaga javob beruvchi bo'lmagani sababli, uni to'la "qo'lda" yozish mumkin. Buni eng oddiy ravishda bajarish uchun biror mavjud bo'lgan protseduraning sarlavhasini ko'chirib turib, uni to'g'rilash, keyin esa kerakli kodni kiritish kerak bo'ladi.

Protseduraning sarlavhasiga yana bir qaytamiz. Sarlavha besh qismdan iborat :

- Birinchi qism — rezervlangan "procedure" so'zi ; beshinchi qismi — oxirgi vergulli nuqta ";". Bu ikkala qism muqarrar sintaktik maqsadlarga xizmat qiladi, chunonchi: birinchisi kompilyatorga sintaktik blok "procedure" aniqlangan to'g'risida xabar beradigan bo'lsa, ikkinchisi esa sarlavha tugaganini bildiradi (haqiqatni olganda, Delphida hamma operatorlar vergulli nuqta bilan tugashi kerak).

- Sarlavhaning ikkinchi qismi — "TForm1" so'zi bo'lib, u mazkur protsedura TForm1ni sinfning usuli deb hisoblanish holatini belgilaydi.

- Sarlavhaning uchinchi qismi — protseduraning nomi. Siz uni xohlaganizcha olishingiz mumkin, o'z ixtiyoringiz bo'yicha. Bizning holatda protsedura "WriteAll" deb nomlangan.

- Sarlavhaning to'rtinchi qismi — parametr. Parametr qavslar ichiga olgan holda beriladi va, o'z navbatida, ikki qismdan iborat. Birinchi qism — parametrning nomi, ikkinchisi — uning tipi. Bu qismlarni ikkinuqta bilan ajratiladi. Agar protsedurada bittadan ortiq parametrlar ko'rsatmoqchi bo'lsangiz, ularni nuqta vergul bilan ajratish kerak, masalan:

```
procedure Example(Param1 : String ; Param2 : String) ;
```

Sinf usuli bo'lgan protseduraning sarlavhasini "qo'lda" barpo etganingizdan keyin, siz uni sinfning deklaratsiyasiga kiritishingiz kerak, masalan, nusxalash yo'li orqali (yana bir marotaba shu to'g'rida eslatib qo'yamizki, delfiyli hodisalarga javoban bo'lgan usullar uchun bunday kiritilish avtomatik ravishda o'tadi):

```
TForm1 q class(TForm);  
Edit1 : TEdit ;  
Edit2 : TEdit ;  
Edit3 : TEdit ;  
Edit4 : TEdit ;  
Edit5 : TEdit ;  
Edit6 : TEdit ;
```

```

EditSource : TEdit ;
procedure Button1Click(Sender : Tobject) ;
procedure WriteAll(NewString: String) ;
end;

```

Bu yerda usulning sarlavhasida "TForm1" so'zini qoldirishning hojati yo'q, chunki u sinfning ta'riflanishida allaqachon qatnashayapti.

Listing S: PARAMS dasturini bosh modulining dastlabki kodi qanday qilib tahrirlash satrlar bilan foydalanish va parametrlar uzatishni ko'rsatadi.

```

Unit Main;
interface
uses WinTypes, WinProcs, Classes, Graphics, Controls,
Printer, Forms, StdCtrls ;
type
TForm1 = class(TForm);
Edit1 : TEdit ;
Edit2 : TEdit ;
Edit3 : TEdit ;
Edit4 : TEdit ;
Edit5 : TEdit ;
Edit6 : TEdit ;
EditSource : TEdit ;
procedure Button1Click(Sender : Tobject) ;
procedure WriteAll(NewString: String) ;
end;
var Form1 : TForm ;
implementation
{$R *.DFM}
procedure TForm1.WriteAll(NewString : String);
begin
Edit1.Text := NewString ;
Edit2.Text := NewString ;
Edit3.Text := NewString ;
Edit4.Text := NewString ;
Edit5.Text := NewString ;
Edit6.Text := NewString ;
end;

```

```

procedure TForm1.Button1Click(Sender : TObject);
begin
WriteAll(EditSource.Text) ;
end ;
end.

```

PARAMS dasturi bilan tajriba o'tkazayotganingizda siz protseduralar va parametrlarning nomlarini o'zgartirib ko'rishingiz mumkin. Biroq, shuni esda tutish kerakki, Delphidagi bir qator so'zlar rezervlangan qatorga kiradilar va ularni identifikatorlar sifatida (protsedura, funsiya, o'zgaruvchilar, tiplar, konstantalar nomlarida) ishlatish ruxsat etilmaydi — kompilyator darhol xatoni ko'rsatadi. Bunday so'zlarga "procedure", "String", "begin", "end" va h.k. kabilar kiradi; ularning to'la ro'yxati Delphi bo'yicha ma'lumotnomalar to'plamining on-line qismida keltirilgan.

Rezervlangan so'zlarning hammasini birdan eslab qolishga urinmang — kompilyator har doim ularni noto'g'ri ishlayotganingiz to'g'risida "Identifier expected" (Identifikator kutilgandi, rezervlangan so'zligi ma'lum bo'lib qoldi) tipli xabarnoma chiqarish orqali eslatib turadi.

### 1.6.3. MURAKKABROQ BO'LGAN USULLAR VA BOSHQARUVCHI ELEMENTLAR

Endi siz Delphi dasturlash tizimidagi asosiy tushunchalarga ega bo'ldingiz, komponent va ularning usullarini barpo etish yo'llarini o'rganishni davom ettirish mumkin. Oldinroq sizga tanishtirib o'tilgan, CONTROL1 dasturida Edit1 tahrirlash satrning OnClick hodisasiga javobnoma bo'lgan usulni generatsiya qilingan edi. Xuddi shuningdek, OnDblClick hodisasiga javob bo'ladigan usulning generatsiyalashtirishimiz mumkin. Diskda mavjud bo'lgan CONTROL2 dasturida komponentlar shaklida joylashgan ro'yxat kengaytirilgan va ularning ko'piga OnClick va OnDblClick hodisalari aniqlangan. Tekshirish uchun, siz CONTROL1 loyihaaning fayllarini CONTROL2 yangi direktoriyaga oddiy ravishda ko'chirishingiz mumkin, loyihaaning nomini CONTROL2.DPR ga o'zgartirib (bu faylda "program" kalitli so'zidan keyin "CONTROL2" nom turishi kerak), shaklga Label, GroupBox, CheckBox, RadioButton, Button komponentlarni ko'shib qo'yish kerak (bu

komponentlar Komponentlar qatoridagi "Standart" betida joylashgan. Sizning shaklingiz taxminan quyidagi ko'rinishli bo'ladi, (1.39-shakl).

Shuni yodda tutmoq kerakki, komponent GroupBox ni shakl ustiga "qo'yish" komponentlar CheckBox va RadioButton larni qo'shishdan ilgari bajarish kerak, ular, bizning misolimizda elementlar guruhining "ichida" bo'lish-lari kerak.

Boshqacha qilib aytiladigan bo'lsa, CheckBox1, CheckBox2, RadioButton1 va RadioButton2 Obyektlar "o'ylashlari" kerakki, ularning "ota-onasi" Form1 shakli hisoblanadi va shakl bo'yicha GroupBox1 harakatlanganda ular joyida qoladi. Shunday qilib, muammoning oldini olish uchun, boshqa komponentlarning (Panel, GroupBox, Notebook, StringGrid, ScrollBox va boshqalar) "ota-onasi" bo'ladigan komponentning shaklga ularning "bolalari" qo'yilmasdan oldinroq o'rnatish kerak. Agar shunday bo'lsa ham bu to'g'rida unutib qo'yib "ota-ona" (masalan, GroupBox) ni ularning "avlodlari" (masalan, SheckBox va RadioButton) keyinroq shaklga qo'yilgan bo'lsa — umidsizlanmang ! Hamma kerakli Obyektlarni belgilang va menyuning dit|Cut buyrug'i yordamida almashlash buferiga ko'chiring (chiqarib yuborish bilan). Shundan keyin shaklda kerakli obyekt (GroupBox1ni) belgilang va menyuning Edit|Paste buyrug'ini bajaring. Shunda oldinroq siz belgilagan hamma obyektlar shaklga joylashgan bo'ladilar va ularning "ota-onasi" GroupBox1 obyekt bo'ladi. Ta'riflangan mexanizm standartli hisoblanadi va uni barcha ko'rinarli komponentlar uchun ishlatish mumkin.

Labell obyektini tanlang. Unga OnDbClick hodisasiga javobnoma bo'lgan usulni barpo eting (bu to'g'rida oldingi betlarda ko'rsatilgan). Usulga bir satrni kiriting, masalan :

```
procedure TForm1.Label1DbClick(Sender : TObject);  
begin  
  Edit1.Text := 'ikki marotaba shiqillatish Labell bo'yicha';  
end;
```

Dasturni ishga tushiring va sichqon bilan Labellning belgisiga shiqillating. Siz ko'rasizki, tahrirlash satri o'zgaradi va unda 'ikki marotaba shiqillatish Labell bo'yicha' matn paydo bo'ladi.

Endi yondoshni berkiting va loyihalash rejimiga qayting. Shaklda mavjud bo'lgan har bir obyekt uchun onClick va onDbClick hodisalalarning ishlovchilarini qo'shib qo'ying. Sizni bosh modulingizning matni quyidagicha ko'rinishli bo'ladi :



Listing - D: CONTROL2 dasturning bosh moduli

```
Unit Main;
interface
uses WinTypes, WinProcs, Classes, Graphics, Controls,
Printers, Forms, StdCtrls, Menus ;
type
TForm1 = class(TForm);
Label1 : TLabel ;
Edit1 : TEdit ;
Button1 : TButton ;
GroupBox1: TGroupBox ;
CheckBox1 : TCheckBox ;
CheckBox2 : TCheckBox ;
RadioButton1 : TRadioButton ;
RadioButton2 : TRadioButton ;
procedure Edit1DbClick (Sender : TObject) ;
procedure Label1DbClick (Sender : TObject) ;
procedure CheckBox1Click (Sender : TObject) ;
procedure CheckBox2Click (Sender : TObject) ;
procedure RadioButton1Click (Sender : TObject) ;
procedure RadioButton2Click (Sender : TObject) ;
procedure Button1Click(Sender : TObject) ;
end;
var Form1 : TForm1 ;
implementation
{$R *.DFM}
procedure TForm1.Edit1DbClick(Sender : TObject);
begin
Edit1.Text := 'Edit1 bo'yicha ikki marotaba shiqillatish';
end;
procedure TForm1.Label1DbClick(Sender : TObject);
begin
Edit1.Text := 'Label1 bo'yicha ikki marotaba shiqillatish';
end;
procedure TForm1.CheckBox1Click (Sender : TObject) ;
begin
Edit1.Text := 'CheckBox1 bo'yicha shiqillatish';
```

```

procedure TForm1.CheckBox2Click (Sender : TObject) ;
begin
Edit1.Text :q 'CheckBox2 bo'yicha shiqillatish' ;
procedure TForm1.RadioButton1Click (Sender : TObject) ;
begin
Edit1.Text :q 'Radiobutton1 bo'yicha shiqillatish' ;
procedure TForm1.RadioButton2Click (Sender : TObject) ;
begin
Edit1.Text :q 'Radiobutton2 bo'yicha shiqillatish' ;
procedure TForm1.Button1Click(Sender : TObject);
begin
Edit1.Text :q 'Button1 bo'yicha shiqillatish' ;
end ;
end.

```

Bu dastur ikki maqsadga xizmat qiladi :

- Qanday qilib protseduralarni (usullarni) barpo etish va qanday ularning ichini ma'noli narsa bilan to'ldirishni ko'rsatadi.
- Windowsning boshqarish elementlari bilan ishlash texnikasi namoyish etadi.

#### 1.6.4. BAJARISH DAVRNING AXBOROTI. CONTROL3 DASTURI

Ehtimol, siz e'tibor bergan bo'lsangiz kerak, CONTROL2 dasturning usullari OnClick va OnDbClick hodisalariga javobnoma bo'lib, ko'p jihatdan bir-biriga o'xshab ketadi.

Delphi muhitining ochiqqligi sababli muhim turli axborotni olish va uning ustida ishlash imkoniyatini beradi. Bu axborotni ishlash davrdagi axborot deyiladi (RTTI - run-time type information) va u bir necha darajali ko'rinishda tashkil qilinadi.

RTTIning yuqori darajasi tiplarni tekshirish va keltirish vositasi sifatida is hamda as kalitli so'zlarni ishlatish orqali amalga oshiriladi.

Kalitli is so'zi dasturchiga mazkur obyekt kerakli tipga egami yoki shu tipning bir avlodi hisoblanadimi, bu haqda ma'lumot beradi, masalan, mana bunday :

```
if MyObject is TSomeObj then ...
```

Obyektli tipni keltirish jarayoni uchun kalitli as so'zini ishlatish orqali RTTI dan foydalanish imkoniyati bor:

```
If MyObject is TSomeObj then
```

(MyObject as TSomeObj).MyField :q ...

quyidagiga ekvivalent :

TSomeObj(MyObject) .MyField :q ...

RTTI ning o'rtancha darajasi kompilyatsiyalash davrida as va is operatsiyalarni almashtirish uchun Obyektlar usullarini ishlatadi. Asosan, bu hamma usullar asosli Tobject sinfiga kiritilgan, ulardan VCL komponentlar kutubxonasining sinflari meros bo'lib chiqadi. Tobject ni ixtiyoriy avlodi uchun, boshqalar qatori, quyidagi axborotli usullarga ega bo'lishi mumkin:

- ClassName — sinf nomini qaytaradi, obyektning nusxasi hisoblanadi
- ClassInfo — ko'rsatkichni RTTI dan jadvalga qaytaradi. Bu jadval obyektlar tipi, ularning ota-onasi hamda e'lon qilinadigan barcha xususiyati, usullari va hodisalari to'g'risidagi axborotga ega.
- ClassParent — obyekt ota-onasining tipi qaytariladi
- InheritsFrom — obyekt ko'rsatilgan sinf avlodi ekanligi to'g'risida mantiqiy qiymat sifatida qaytaradi
- InstanceSize — obyekt o'lchamini bayt hisobida qaytaradi.

Bu usullarni siz o'z kodlaringizda to'g'ridan-to'g'ri ishlatishingiz mumkin.

RTTI ning pastki darajasi delfiyli TypInfo moduli bilan aniqlanadi va komponentlar ishlab chiqaruvchilar uchun alohida qiziqish tug'diradi. Ular orqali Delphining ichki strukturalariga kira olish mumkin, ular qatori, shakllar resursiga, Obyektlar inspektoriga va h.k.

Shunday qilib, Delphida bajarish davrdagi axborotga kira olish shaklda joylashgan obyektning nomini dinamik ravishda olish bilan birgalikda, unga tegishli sinf nomiga ham ega bo'lish imkoniyati bor. Buning uchun Name xususiyati ishlatiladi. Bunday xususiyatlarga ixtiyoriy sinf — Tcomponent avlodi (VCL delfiyli kutubxonasiga kirgan barcha komponentlar ana shunday) va ClassName usuli, asosli Tobject sinfining ixtiyoriy avlodlari uchun kira olish mumkin bo'lganlar kiradi.

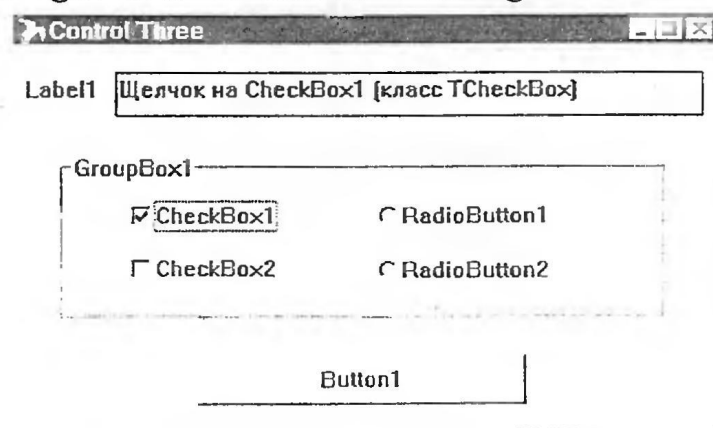
Ko'rgan misollarimizga qaytib, biz to'la usullar "to'plamini" ikkita bilan almashtirishimiz mumkin, ular barcha obyektlar uchun darhol OnClick va OnDbClick hodisalarni amalga oshirishi mumkin. Buning uchun CONTROL2dagi barcha fayllarni yangi CONTROL3 direktoriyasiga ko'chirish kerak yoki ishlash uchun diskda mavjud dasturdan foydalanish lozim. Standart ravishda biror obyekt uchun

(masalan Label1 uchun) ControlDbClick SontrolClick usullarni barpo etamiz. Quyidagi satrlarni kiritamiz :

```
procedure TForm1.ControlDbClick(Sender:TObject);
begin
  Edit1.Text :q 'ikki marotaba shiqillatish' Q ga
  (Sender as TCoponent).Name Q
  ' (sinf' Q Sender. ClassName Q ' )' ;
end;
procedure TForm1.ControlClick(Sender : TObject);
begin
  Edit1.Text :q ',bir marotaba shiqillatish' Q ga
  (Sender as TCoponent).Name Q
  ' (sinf' Q Sender. ClassName Q ' )' ;
end;
```

Endi, obyektlar shaklida joylashgan OnClick va OnDbClick barcha hodisalariga shu usullarni qo'llaymiz. Ko'ramizki, dastur hajmi deyarli qisqardi, funksional darajasi esa anchagina oshdi. Bajarish rejimida, CheckBox1 obyekti bo'yicha shiqillatishdan keyin, yondoshni ko'rinishi 10.40-shaklda ko'rsatilgandek bo'ladi

Shunday qilib ko'rib turganimizdek, bajarish davridagi axborotni ishlatib, dasturni eguvchan va universal holatga keltirdik.



1.40-shakl.

## 1.7. DYRAZALI YONDOSHLAR

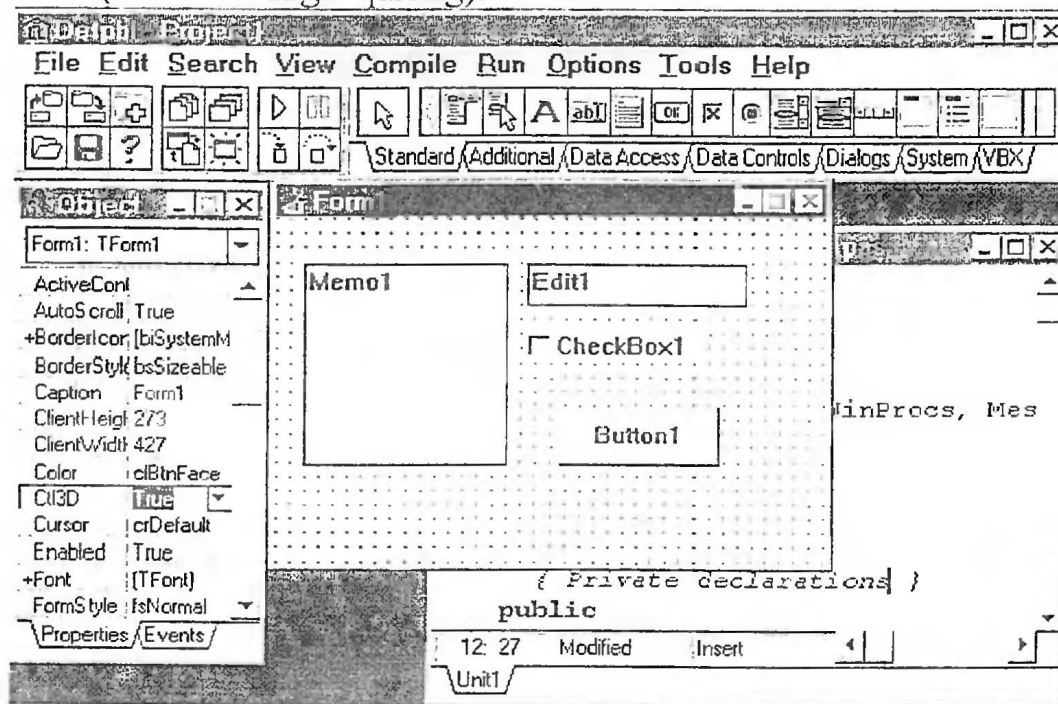
Bu bobda, yondoshlarning asosiy turi bo'lgan — derazali yondoshlarni barpo etish texnologiyasi ko'rib chiqiladi, ular Windows tizimlarida ishlatiladi. Derazali yondoshlarga Windowsda ishlash uchun barpo etilgan amaliy dasturlarning ko'pchiligi kiradi, masalan, Provodnik,

matnli protsessori Word, jadvalli protsessor Excel, grafikli Paint muharriri va boshqa dasturlar. Bundan farq qiluvchi, konsolli yondoshlar derazali hisoblanmaydilar, chunki konsolli yondoshlarning bajarishlarini operatsionli MS-DOS tizimi ta'minlaydi. Konsolli yondoshlar to'g'risidagi olingan bilim Object Pascal tilida dasturlash asoslarini o'rganishga imkoniyat berdi. Hozir esa, Object Pascalni o'rganishda davom etamiz va asosiy diqqatni sinf, obyekt va komponent degan tushunchalarga qaratamiz, ular bevosita derazali yondoshlarni barpo etish jarayoni bilan bog'langanlar. Oldinroq olingan hamma bilimlarimizni deyarli barchasi hozir ham kerak bo'ladi. Faqat endi kiritish va chiqarish — read, readln, write, writeln protseduralari derazali yondoshlarda ishlatilmaydi. Ularning o'rniga kiritish va matnli axborotni aks ettirish uchun maxsus komponentlar ishlatiladi.

### 1.7.1. DERAZALI YONDOSHLARNI BARPO ETISH BOSQICHLARI

Bundan oldinroq, Delphi dasturlash muhitida qanday asosiy derazalar bilan foydalanish mumkinligi to'g'risida aytib o'tilgan edi. Ular bilan batafsilroq tanishish vaqti keldi.

Delphi 6 ishga solingandan keyin kompyuter ekranida ko'rishingiz mumkin (10.41-shaklga qarang).



1.41-shakl. Delphi 6 muhitining deraza va panellari

1 — ekranning yuqori qismida asosiy menyu joylashgan, u dasturlash muhitining buyruqlariga kira olishni ta'minlaydi; asosiy menyu asosiy deraza deb nomlanadigan va Delphi 6-Project sarlavhaga ega bo'lganda joylashadi.

2 — asosiy derazada instrumentlar panellari mavjud, ular asosiy menyuning asosan ko'proq ishlatiladigan buyruqlarini tezda bajarish imkonini beradilar.

3 — asosiy derazaning o'ng pastki qismida obyektlarning katta to'plamiga ega bo'lgan Komponentlar qatori joylashgan, ularni shaklga qo'shish mumkin; xuddi shu komponentlar Delphi muhitida yondoshlarni barpo etish uchun vosita bo'lib xizmat qiladilar.

1 — ekranning o'rtasida Form 1 sarlavhasi bilan shakl derazasi joylashgan, u Delphi ning komponentlarini joylashtirish uchun ishlatiladi.

2 — kod muharririning derazasi, u dastur kodlariga ega, aytib o'tmaslik prinsipi bo'yicha Units.pas sarlavhaga ega va u dasturning kodlarini barpo etish hamda tahrirlashga mo'ljallangan.

3 — Obyektlar Inspektorining (Object Inspector) derazasi komponentlar xususiyatini (xarakteristikasini) o'zgartirish imkonini beradi.

4 — Obyektlar Daraxtining derazasi (Object TreeView) Delphi 6 da paydo bo'lgan yangilik ; bu derazada shaklda joylashtirilgan komponentlar aks ettiriladi hamda ularning orasidagi bog'lanishlar beriladi.

Vizualli dasturlash terminologiyasida loyiha obyektlari (komponentlari) deb dialogli derazalarni va dialogli derazada joylashtirilgan boshqarish elementlari — buyruq tugmachalar, matnli axborotni kiritish uchun maydonlar, o'zgartiruvchilar, menyu va h.k. larga aytiladi.

Obyekt xususiyatiga , masalan, uning o'lchami (balandligi va eni), ekrandagi yoki shakldagi holati, sarlavha matnni yoki buyruq tugmachasidagi matn kiradi.

Hodisa — bu yondoshni ishlash vaqtida bo'ladiganlar. Delphida har bir hodisaning nomi bor. Masalan, sichqon klavishasini shiqillatish - bu OnClick hodisasi, sichqon bilan ikki marotaba shiqillatish - OnDblClick, klavisha bosilgan holatdagi hodisa - OnKeyDown, bosilgan klavisha qo'yib yuboriladigan bo'lsa - OnKeyUp va h.k.



Hodisaga javob sifatida biror amal (harakat) bo'lishi mumkin, masalan, dialogli derazada buyruq tugmachasi bo'yicha sichqon klavishasi bilan shiqillatish natijasida hosil bo'lgan OnClick hodisasiga javob sifatida berilgan formula bo'yicha hisobni bajarish va natijani chiqarish ishlari bo'lishi mumkin. Delphida hodisaga bo'lgan javobni hodisaning ishlov protsedurasi deb olinadi. Shunday qilib, dasturchining vazifasi hodisalarning kerakli ishlovini olib boruvchi protseduralarni ko'rsatishdan iborat bo'ladi.

Oddiy derazali yondoshni barpo etishdagi yiriklashtirilgan jarayonni ikki bosqichga bo'lish mumkin: shaklni konstruksiyalash bosqichi va dasturlash bosqichi. Bu bosqichlarni batafsilroq ko'rib chiqamiz.

Shaklni konstruksiyalash bosqichida bajariladigan amallarni keltiramiz:

1. Eng avval Komponentlar qatoridan kerakli komponentlarni tanlab olish kerak va ularni shaklga joylashtirish lozim. Komponentlarni shaklga joylashtirish uchun Komponentlar qatoridagi komponent bo'yicha sichqonni chap klavishasi bilan shiqillatish kerak, keyin esa, komponentni shaklga qo'yiladigan joyiga sichqon bilan shiqillatish lozim.

2. Shaklga joylashgan komponentni shakl bo'yicha sichqon yordamida ko'chirish mumkin bo'ladi va uning o'lchamini o'zgartirish uchun uning ichiga olgan ramkaning markerlari bilan foydalanish kerak.

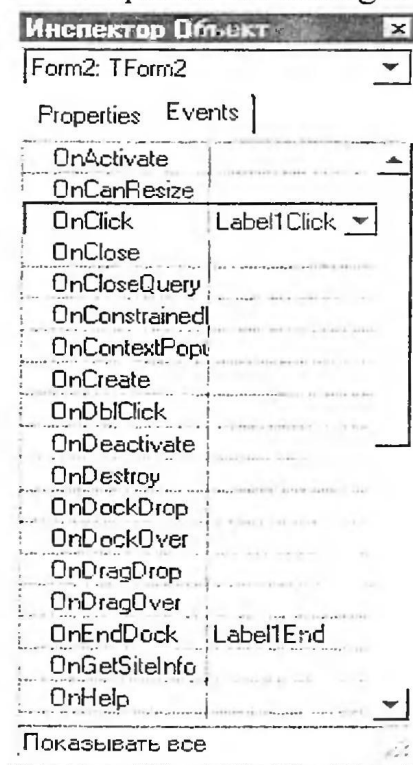
Komponentga kerakli xususiyatga ega qilish uchun esa, Obyektlar Inspektorining derazasidagi Properties betidan foydalanish lozim bo'ladi. Bu bet ikkita ustunlardan iborat: chap ustun xususiyatni nomiga ega, o'nggi esa — uning muayyan qiymatini ko'rsatadi. Obyektlar Inspektorining derazasi sichqon bilan ajratilgan komponentning axborotini aks ettiradi. Bu derazaning bet satrlarini sichqon bilan shiqillatish orqali tanlab olinadi va u yerda oddiy yoki murakkab xususiyatlar aks ettirilishi mumkin. Oddiy xususiyatlarga bir qiymat bilan aniqlanadiganlar kiradi — son, simvollar satri, False yoki True qiymat va h.k. Murakkab xususiyatlar qiymatlar to'dasi bilan aniqlanadi. Bunday xususiyatlarning nomidan chap tomonida "Q" belgisi ko'rsatiladi.

Ikki marotabali shiqillatish uning nomi bo'yicha murakkab xususiyatning qiymatlar ro'yxatining ochilishiga olib keladi. Ochilgan ro'yxatni yopish uchun ham murakkab xususiyatning nomi bo'yicha



qaytadan ikki marotaba shiqillatish kerak bo'ladi. Obyektlar Inspektoriga kengaytirilgan ichki komponentli havolalar deb nomlanadiganlarni (expanded inline component references), qisqasi, ichki komponentlarning qo'shilishi Delphi 6 dagi juda ham qiziqarli yangilik bo'ldi. Bu atama orqasida shuni nazarda tutiladiki, komponentlarning ayrim xususiyatlarining qiymati sifatida boshqa komponentlarning nomi olinadi (ya'ni boshqa komponentlarga havolalar). Masalan, komponentlarni ko'pida PopurMenu xususiyat bor, u kontekstli menyusining nomidan iborat. Ichki komponentlarga havola bo'lgan xususiyatlar Obyektlar Inspektorida qizil rang bilan aks ettiriladi (aytib o'tmaslik prinsipi bo'yicha). Qachonki, bunday xususiyatga qiymat beriladigan bo'lsa, uning oldida "Q" belgisi paydo bo'ladi. Agar ichki komponentning nomiga ega bo'lgan xususiyatga ikki marotaba yoki oddiy ravishda "Q" belgisiga shiqillatilsa, ichki komponentning xususiyatlar ro'yxati ochiladi. Ichki komponentning xususiyatlari aytib o'tmaslik prinsipi bo'yicha yashil rang bilan aks ettiriladi.

Yuqorida ko'rsatilgan amallarni bajarish natijasida kelajak dasturning derazasini tashqi ko'rinishi barpo etiladi. Dastur biror foydali ishni bajarishi uchun, ikkinchi bosqich — dasturlash bosqichiga o'tish kerak bo'ladi. Bu bosqichning asosiy xosiyatlarini ko'rib chiqamiz.



1.42-shakl. Obyektlar Inspektorining derazasi

1. Har bir komponent ma'lum hodisalar to'plamiga ta'sir qiladi. Aniq qaysi bir komponentga javob berishini bilish uchun, uni shaklda sichqonni shiqillatish orqali ajratib, keyin Obyektlar Inspektorining Events betiga o'tish kerak (1.42-shaklga qarang).

Bet ikki ustunga ega. Birinchi ustunda hodisalarning nomlari keltirilgan, ikkinchisi dasturchi tomonidan to'lg'aziladi va barpo etilgan hodisalar ishlovchisining nomlaridan iborat. Nazariy tomondan barcha hodisalarning ishlovchisini barpo etish mumkin, ayrim hodisalar uchun umuman ularning ishlovini olib boruvchisini barpo etmasa ham bo'ladi. Oxirgi holatda komponent boshqa kompo-

nentlarga barpo etilgan hodisalarning ishlovini olib boruvchilari bilan foydalanishi mumkin. Bironta bitta hodisaning ishlovini olib boruvchisini barpo etish mumkin, keyin esa Obyektlar Inspektorining Events betida xuddi shunday tipga ega bo'lgan boshqa hodisalarning nomlarini ko'rsatish kerak. Bunday holatda bitta protsedura har xil hodisalar paydo bo'lganda bir o'zi ishlaveradi. Shunday qilib, dasturlash bosqichida, eng avval biz u yoki bu komponent qaysi bir hodisaga javob berishi mumkinligini aniqlab olishimiz kerak va Obyektlar Inspektorining Events betida barpo etiladigan ishlovchining nomini kiritishimiz kerak.

2.Hodisalar ishlovchisi protseduradan iborat bo'lib, bir-biridan nuqta belgisi bilan ajralgan ikki qismli nomga ega. Birinchi qism barpo qilinayotgan shaklning sinf nomini beradi. Sinf tushunchasi to'g'risida keyingi boblarda batafsilroq ko'rib chiqamiz. Nomning ikkinchi qismi, yuqorida aytib o'tilgandek, dasturchi tomonidan beriladi, yoki Delphi muhiti uni barpo etadi, agarda dasturchi o'z nomini kiritmagan bo'lsa. Agar ishlov beruvchi nomining ikkinchi qismi avtomatik ravishda Delphi tomonidan tuzilayotgan bo'lsa, u holda bu komponentning nomi va hodisaning On old ko'makchisiz nomidan tashkil topadi, masalan:

```
procedure TForm1.Label1Click(Sender:TObject);
```

Bu yerda TForm1 — barpo etilayotgan shakl sinfning nomi, Label1Click — Delphi tomonidan avtomatik ravishda barpo qilingan hodisa ishlovchisi nomining ikkinchi qismi, u protsedura bajarilishi uchun Label1 nomli komponent bo'yicha sichqonni chap klavishasi bilan shiqillatish kerak, degan ma'noni bildiradi. TObject tipli Sender parametri OnClick (ya'ni Label1) hodisani barpo etgan komponent havolaga ega bo'ladi.

3. Obyektlar Inspektorining Events betida hodisa ishlov beruvchisining nomini ko'rsatganingizdan so'ng yoki uni Delphi avtomatik ravishda tashkil qilganda ko'rsatmasangiz, hodisa ishlov beruvchisining tayyorlovini yaratish zarur. Buning uchun Obyektlar Inspektorida barpo qilinayotgan ishlov beruvchining nomi qo'yiladigan maydoni bo'yicha ikki marotaba shiqillatishni bajarimiz. Natijada kod muharrirning derazasiga ishlov beruvchining tayyorlovi qo'shiladi, masalan :

```
procedure TForm1.Label1Click(Sender: TObject);  
begin
```

end;

Protseduraning tanasi rezervlangan begin... end so'zlari bilan cheklangan, ularning orasiga dasturchi o'z operatorlarini joylashtirishi kerak.

4. Dasturchi tomonidan kiritiladigan operatorlar ichida shundaylari ham bo'lishi mumkinki, qaysilar shaklda joylashgan komponent xususiyatining qiymatini o'zgartirishi mumkin. Bu shuni bildiradiki, komponentlar xususiyatini nafaqat loyihalash bosqichida Obyektlar Inspektori yordamida, balki dasturni bajarish jarayonida ham o'zgartirish mumkin bo'ladi. Komponent xususiyatlarini dasturni bajarish jarayonidagi o'zgarishini xususiyatlarni dinamikli o'zgarishi deb ataladi.

Qayd qilamizki, aytib o'tmaslik prinsipi bo'yicha Delphining har bir komponenti faqat bitta hodisaga egadir. Odat bo'yicha, bu hodisaga ko'proq ishlov beruvchilar tuziladi. Masalan, Delphining ko'p komponentlari uchun bunday hodisa sifatida OnClick hisoblanadi. Aytib o'tmaslik prinsipi bo'yicha berilgan hodisa ishlov beruvchisining tayyorlovini barpo etish uchun komponent bo'yicha sichqon bilan ikki marotaba shiqillatish yetarli bo'ladi.

Shakl va muharrir derazalari ichidagilar orasida ajralmas bog'lanish mavjuddir, uni qat'iy ravishda Delphi nazorat qilib turadi. Masalan, komponentning shaklga joylashtirilishi dastur kodini avtomatik ravishda o'zgarishiga olib keladi. Yuqorida aytib o'tilgandek, avtomatik ravishda hodisaga ishlov beruvchining tayyorlovi ham barpo etiladi. Shu bilan birga dasturchi tayyorlovlarni muayyan ma'no bilan to'ldirishi mumkin - operatorlar kiritishi, xususiy o'zgaruvchilar, tiplar, konstantalar va h.k. larning ta'rifini qo'shishi. Shu bilan birga dasturchi shuni ham esda tutishi kerakki, dastur matnidan Delphi muhiti kiritgan satrlarni chiqarib yuborish mumkin emas.

### 1.7.2. DELPHI LOYIHASINING STRUKTURASI

Delphi dasturi — bu bir necha o'zaro bir-biri bilan bog'langan fayllar. Shunday, har qanaqa dastur bizga oldindan tanish bo'lgan loyiha fayllaridan (bunday fayllar .dpr kengaytmayli bo'ladilar) va bitta yoki bir necha modullardan (.pas kengaytmali fayllar) iborat bo'ladi. Loyiha fayli foydalanuvchi tomonidan tahrirlashga mo'ljallangan emas va ularni Delphi dasturlash muhiti avtomatik ravishda barpo etiladi. Loyiha

faylining ichidagini ko'rish uchun Project| View Source buyrug'ini bajarish kerak. Loyiha faylining ichi, masalan, quyidagicha bo'lishi mumkin:

```
Program Project ;
uses
Forms,
Unit in 'Unit1.pas' {Form1} ;
{$R * .RES}
begin
Application.Initialize ;
Application.CreatForm(TForm1, Form1) ;
Application.Run ;
end.
```

Loyiha fayli (bosh modul) program so'zi bilan boshlanadi, undan keyin loyiha nomi bilan mos keladigan dasturning nomi keltiriladi. Loyiha nomi dasturchi tomonidan loyiha faylini saqlash paytida beriladi ( .exe kengaytmali fayl). Shundan keyin uses so'zi ketidan foydalanadigan modullarning nomlari beriladi : standartli Forms va shaklning Unit1 modullarning. Sharhlovchiga o'xshash bo'lgan {\$R \* .RES} direktivasi kompilyatorga shuni ko'rsatadiki, resurslardan foydalaish kerakligi, qaysil yondoshlarning resurslari ta'rifga ega, masalan piktogrammalar. Yulduzcha (\*) resurslar fayllarining nomi xuddi loyiha faylinikiday, faqat .res kengaytmali.

Bosh modulning bajariladigan qismi begin ... end. operatorli qavslar ichida joylashgan bo'ladi. Bosh modulning bajariluvchi qismi yondoshlarni initsializatsiyalashtirilishiga va ekranga startli derazaning chiqarilishiga imkon yaratadi.

Bosh moduldan tashqari har bir dastur kamida bitta shaklning modulini o'z ichiga oladi, ular yondosh startli shakllarning ta'rifiga ega va ularning protsedurasi ishlashini qo'llab turadi. Delphida har bir shaklni o'zining tegishli moduli bor. Shakldan kod muharririning derazasiga o'tish uchun bosh menyuning View | Toggle | Form G' Unit buyrug'ini bajarish lozim, funksional F12 klavishasini bosish kerak, yoki instrumentlar View panelida sichqon bilan oo tugmachasiga qaratib shiqillatish lozim.

Modullar — bu dasturli birliklari, dastur fragmentlarini joylashtirishga xizmat qiladi. Ularning ichidagi dastur matnlari (dastur

kodlari) yordamida foydalanuvchining yechiladigan masalasi amalga oshiriladi.

Modullar Object Pascal dasturlash tili tomonidan imkoniyatli bo'lgan standart konstruksiyaga (bo'limlarning ketma-ketligi va sang'iga) ega. Modulning strukturasini umumiy ko'rinishga keltiramiz:

```
unit <modulning nomi> ;  
interface  
.....  
implementation  
.....  
initialization  
.....  
finalization  
.....  
end.
```

Misol sifatida modulning ichini xuddi Delphi muhiti yuklangandan keyin ko'rinishga ega bo'lgan holatini keltiramiz :

```
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 : TForm ;  
implementation  
{SR *.dfm}  
end.
```

Modul unit so'zi bilan boshlanadi, uning ketidan modulning nomi beriladi. Xuddi shu nom yondoshning bosh moduldagi uses operatorining foydalanadigan modullar ro'yxatida ko'rsatib o'tiladi.

Modul to'rtta bo'limlardan iborat bo'lishi mumkin: interfeys, amalga oshiruvchi, initsializatsiyalashtiruvchi va tugatuvchi qismidan.

Interfeys bo'limi (interface so'zi bilan boshlanadi) kompilyatorga shu to'g'rida xabar qiladi, modulda joylashgan qanday berilganlar dasturning boshqa modullariga ega bo'lish imkoniyati bor. Bu bo'limda (uses so'zidan keyin) ushbu modulda foydalanadigan modullar keltirilgan, hamda type so'zi ketidan Delphi tomonidan barpo etilgan shakl tipining ta'rif joylashgan (1.8-bobga qarang).

Amalga oshiruvchi bo'limi implementation so'zidan boshlanadi va shaklning ishini quvatlaydigan lokal o'zgaruvchilar, protsedura va funksiyalarga ega bo'ladi. Bo'lim boshida {\$R \*.dfm} direktivasi joylashgan, u kompilyatorga amalga oshirish bo'limiga shakl xususiyatining qiymatini o'rnatuvchi buyruqlar kiritish kerakligini xabar qiladi, ular .dfm kengaytinali faylda joylashgan bo'lib, nomlari modulning nomi bilan bir xilda bo'ladi. Format dfm bo'lgan fayllar shaklning tashqi ko'rinishiga asosan Delphi tomonidan generatsiyalashtiriladi.

Direktiva {\$R \*.dfm}dan keyin shaklning hodisalar ishlovchisining protseduralarini ta'rif joylashtiriladi. Shu yerga ham dasturchi hodisa ishlovchisining protsedurasidan chaqiriladigan o'z protsedura va funksiyalarning ta'rifini o'rnatiradi. Initsialashtiruvchi va tugatuvchi qismlarni berish shart emas.

Initsialashtiruvchi qism initialization so'zi bilan boshlanadi yoki begin ... end operatorli qavslar ichiga olinadi. Bu qismdagi operatorlar asosiy dasturga boshqaruvni berishdan ilgari bajariladi va u odatda tayyorgarlik ishlarini bajarishda foydalaniladi.

Tugatuvchi qism finalization so'zi bilan boshlanadi va u dasturning ishi tugash paytida bajariladigan operatorlardan iborat. Yuqorida keltirilgan modul misolida initsialashtiruvchi va tugatuvchi qismlar yo'q.

Modulni avtomatik ravishda Delphi tomonidan barpo etiladigan loyiha faylidan farqi shundan iboratki, uni dasturchi o'zgartirishi (tahrirlashi) mumkin. Foydalanuvchi yangi shakl barpo etganda, avtomatik ravishda yangi modul ham barpo etiladi. Dastur tarkibiga bir necha o'nlab shakllar kirishi mumkin. Shu bilan birga modulning matni foydalanuvchiga va Delphi muhitining o'ziga ham kira olish imkonini beradi. Delphi muhiti avtomatik ravishda modulning matniga har qanday shaklga qo'shilgan komponentning ta'rifini kiritishi hamda hodisa



ishlovchilariga tayyorlovlar (kod satrlarini) barpo etishi mumkin. Ular qatori dasturchi o'z usullarini oldinroq e'lon qilingan sinflarga qo'shishi, hodisa ishlovchilarini muayyan ma'no bilan to'ldirishi, xususiy o'zgaruvchilar, tiplar, konstantalar va h.k. larni kiritishi mumkin. Lekin, bundan oldinroq aytilgandek, dasturchiga Delphi muhiti tomonidan modulga kiritilgan satrlarni chiqarib yuborish man etiladi. Dasturni kompilyatsiyalash paytida Delphi har bir modul uchun .dcu kengaytmali fayllar barpo etadi.

Shunday qilib, pas kengaytmali fayl modulning dasturli kodiga ega, u kod muharririning derazasida dasturchi va Delphi muhitining birgalikdagi harakati tufayli hosil bo'ladi. Kengaytmasi .dfm bo'lgan fayllarda shakl deraza ichidagilarning ta'rif saqlanadi, dcu fayllarda esa, ikkala fayllarning matnini mashina instruksiyalariga qayta o'tkazishdan chiqqan natija joylashadi. Delphi muhitidagi tarkib etuvchisi (kompanovshik) dcu-fayllarni yagona yuklanuvchi (bajariluvchi) exe-fayllarga o'zgartiradi. Bajariluvchi fayl dasturni avtonomli yondosh sifatida ishga tushirishga imkon beradi.

### **1.7.3. ODDIY DERAZALI YONDOSHNI BARPO ETISH MISOLI**

Yuqorida barcha aytilganlarni oddiy derazali yondoshni barpo etish misolida namoyish etamiz

**Misol 1.7.1.** Ixtiyoriy radiusli doiraning yuzasini aniqlovchi dastur tuzilsin.

Yechim. Derazali yondoshni barpo etish jarayonini qadamma-qadam bajarishni ta'riflab o'tamiz.

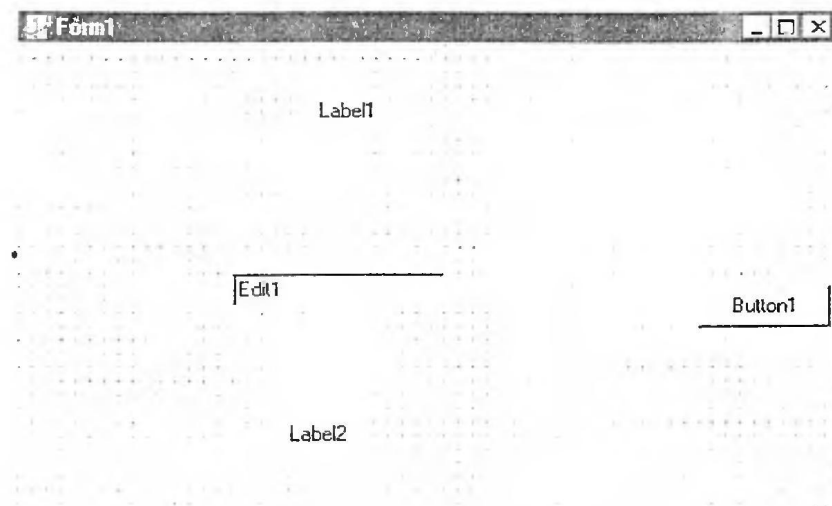
1. Eng avval, loyihaga kiruvchi barcha fayllarni joylashtirish uchun yangi papka barpo etish kerak. Keyinchalik har doim har bir yangi loyihaga alohida papkani barpo etamiz. Qayd qilamizki, yangi papkani loyihaga kirgan fayllarni saqlab qolish paytida ham barpo etish mumkin, buning uchun fayllarni saqlash dialogli derazasida mavjud bo'lgan "Создание новой папки" tugmachasi bilan foydalanish mumkin.

2. Yangi loyihani barpo qilish uchun bosh menyuning File | New | Application buyrug'i bilan foydalanish mumkin yoki, agar siz Delphi muhitini endigina yuklagan bo'lsangiz, yangi loyihani barpo etishga kirishingiz mumkin.

3. Komponentlar qatorining Standart betidan Form1 shaklga to'rtta komponentlarni joylashtiramiz: Label1, Label2, Edit1 va Button1 (1.43 shaklga qarang).



Komponentlarni shaklga joylashtirish paytida bir yo'la ularning o'lchamlarini ham beramiz. Buning uchun Komponentlar qatoridan komponentni tanlab olingandan keyin shaklda to'g'ri to'rtburchakli sohani ko'rsatish lozim, qayerni komponent egallaydi. Sohaning chap yuqori tomoni sichqonni chap klavishasini shiqillatish orqali aniqlanadi, keyin, sichqon klavishasini qo'yib yubormasdan, sichqon ko'rsatkichini sohaning pastki o'ng burchagiga siljitish kerak.



1.43-shakl 1. Shaklda Label1, Label2, Edit1 va Button1 komponentlarning joylashishi.

Foydalanadigan komponentlar to'g'risida batafsilroq ma'lumotni keyingi boblarga qoldiramiz, hozircha ular to'g'risida qisqacha axborot berish bilan cheklanamiz.

Label1, Label2-belgilar-matnli axborotni aks ettirish uchun mo'ljallangan komponentlar. Ular tomonidan aks ettiruvchi matn shaklni konstruksiyalash bosqichida hamda dasturni bajarish paytida ham ularda joylashi mumkin. Label1 belgiga "doiraning radiusini kiriting" matnini joylashtiramiz, Label2 belgiga esa, dasturni bajarish bosqichida hisoblangan doiraning yuzasini qo'yamiz.

Edit1 — kiritish satri-simvolli satrni kiritish uchun mo'ljallangan komponent. Bu komponent yordamida radiusning qiymatini kiritamiz. Berilganlar simvolli satr ko'rinishda kiritilishi sababli, dasturda uni haqiqiy songa aylantirishni ko'zda tutmoq kerak.

Button1 — tugmacha — uni bosilganda hodisani vujudga keltirish uchun mo'ljallangan komponent. Bizning dasturimizda tugmacha bosilgan holda doiraning yuzasini hisoblab chiqiladi.

4. Bo'lajak dasturimizning tashqi ko'rinishini tuzishga o'tamiz. Buning uchun, Obyektlar Inspektorini ishlatib, shakl xususiyati va undagi komponentlarni o'rnatamiz. Shakldan boshlaymiz.

Shaklni faollashtiramiz, sichqon bilan boshqa komponentlar bilan band bo'lmagan joyga shiqillatib. Bundan keyin har doim Obyektlar Inspektorida u yoki bu komponentning xususiyatini o'zgartirayotganda avval uni sichqon bilan ajratildi deb ko'zda tutamiz.

Obyektlar Inspektorida Caption (Sarlavha) xususiyatini tanlaymiz va "Form1" matn o'rniga "Doiraning yuzini hisoblash" matn kiritamiz. Shunday qilib biz shaklimizning sarlavhasini o'zgartiramiz.

Shuni aytib o'tamizki, Capiton xususiyatning qiymatlarini simvolli satrlar tashkil qiladi, ya'ni string tipli berilganlar. Shuning uchun keyinchalik Capiton xususiyatning qiymatini hamda String tipli boshqa xususiyatlarni Object Pascalda qabul qilingandek bir tirnoqli apostroflar ichiga olib yozamiz. Masalan: 'Form1', 'doiraning yuzini aniqlash'.

Shakl o'lchamlarini va, demak, bo'lajak dasturning derazasini o'zgartiramiz. Buning uchun sichqon kursorini shaklning biror chetiga keltirish kerak (shunda kursor ikki yo'nalishli strelka ko'rinishga kiradi) va, sichqonning chap klavishasini qo'yib yubormasdan, shaklning o'lchamini o'zgartirish mumkin. Shu bilan birga avtomatik ravishda Obyektlar Inspektoridagi Height (balandligi) va Width (eni) xususiyatlar o'zgaradi. Bu demak, shaklning o'lchamlarini to'g'ridan-to'g'ri Obyektlar Inspektorida Height va Width xususiyatlarga qiymat berish orqali o'zgartirish mumkin edi. Lekin, boshqa tomondan, vizualli dasturlashning afzalligi xuddi shundan iboratki, shaklning loyihalash jarayonini bevosita kuzatib boriladi.

Ko'rsatilgan usullardan birini qo'llab, shaklning Height va Width xususiyatlarini mos ravishda 350 va 400 (piksel) teng qilib olamiz.

Left va Top xususiyatlar display ekranining chap yuqori burchagidan shaklning chap yuqori burchagigacha mos ravishda gorizontaal va vertikal masofalarni tayinlaydi. Ularning qiymatini mos ravishda 300 va 200 (piksel) teng qilib o'rnatamiz. Buning uchun Obyektlar Inspektoridan foydalanish mumkin yoki sichqon bilan uning sarlavhasiga ilintirib olib, shaklni ko'chirish kerak. Natijada bizning shakl ekranning taxminan o'rtasida joylashadi.

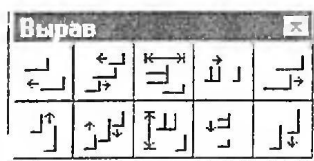
Komponentlarning Label1 va Label2 xususiyatlarini o'rnatamiz. Label1 va Label2 komponentlari ham Height, Width, Left va Top

xususiyatlarga egadirlar. Lekin, shakldan farqi shundaki, Left va Top xususiyatlari komponent joylashgan shaklning chap burchagidan komponentning o'zini yuqori chap burchagicha bo'lgan masofani ko'rsatadi. Label1 va Label2 komponentlar uchun shu xususiyat qiymatlarini keltiramiz.

|        | Label1 | Label2        |
|--------|--------|---------------|
| Height | 57     | ixtiyoriy son |
| Width  | 129    | ixtiyoriy son |
| Left   | 131    | 100           |
| Top    | 34     | 218           |

Label2 komponent uchun Height va Width xususiyatlarning qiymatlarini ixtiyoriy qilib olsak bo'ladi, nima uchun bunday qilishni keyinroq tushuntiramiz.

Bu qiymatlarni o'rnatish uchun, shaklga berilgan singari, komponentni shakl bo'yicha siljitib sichqon bilan uning o'lchamlarini o'zgartirish yoki Obyektlar Inspektorining yordamiga murojaat qilish mumkin Bundan tashqari Align (to'g'rilamoq) instrumentlar paneli bilan foydalanish mumkin (1.44-shakl).



1.44-shakl. Align instrumentlar paneli.

Bu panelni ekranga chiqarish uchun Delphining bosh menyusidagi View | Alignment Palette buyrug'ini bajarish kerak. Bu panelda ko'rsatilgan tugmachalar shaklning ichida ajratilgan komponentlarni to'g'rilash imkonini beradi. Masalan:

- shaklning vertikal simmetriya o'qiga nisbatan markazlashtiradi;
- shaklning gorizontal simmetriya o'qiga nisbatan markazlashtiradi.

Nafaqat bir komponentni, balki bir yo'la bir necha komponentlarni to'g'rilash mumkin. Komponentlar guruhini ajratish uchun, Shift klavishasini bosib turgan holda ajratishni bajarish kerak. Komponentlarni to'g'rilash natijasida Obyektlar Inspektorida Left va Top xususiyatlarni avtomatik ravishda o'zgarishiga olib keladi.

Caption xususiyati belgi uchun asosiy hisoblanadi va u aks ettiriluvchi matnga ega. Label1 belgi uchun Caption xususiyatini

'Doiraning radiusini kiriting va Schet tugmachasini bosing' qilib o'rnatamiz. Label2 belgi uchun Caption xususiyatini dasturni bajarish bosqichida aniqlanadi.

Belgining AutoSize xususiyati aniqlaydi: belgining o'lchamini u yerga joylashgan simvolli satrning uzunligiga qarab avtomatik ravishda o'rnatish kerakmi yoki yo'qmi? Agar "ha" bo'lsa, u holda AutoSize xususiyatning qiymatini True qilib olish kerak, aks holda — False.

WordWrap xususiyati (so'zlarni ko'chirish) so'zni ko'chirishga ruxsat beradi yoki man etadi, agar satr belgiga sig'masa va AutoSize xususiyati False teng bo'lsa. Birinchi holatda WordWrap xususiyati True teng qilib o'rnatilishi kerak, ikkinchisida — False.

Alignment xususiyati (to'g'rilash) belgi ichida matnni to'g'rilanishini aniqlaydi: chap tomon bo'yicha, markazga yoki o'ng tomon bo'yicha.

Label1 va Label2 belgilar uchun bu xususiyatlarning qiymatlarini keltiramiz :

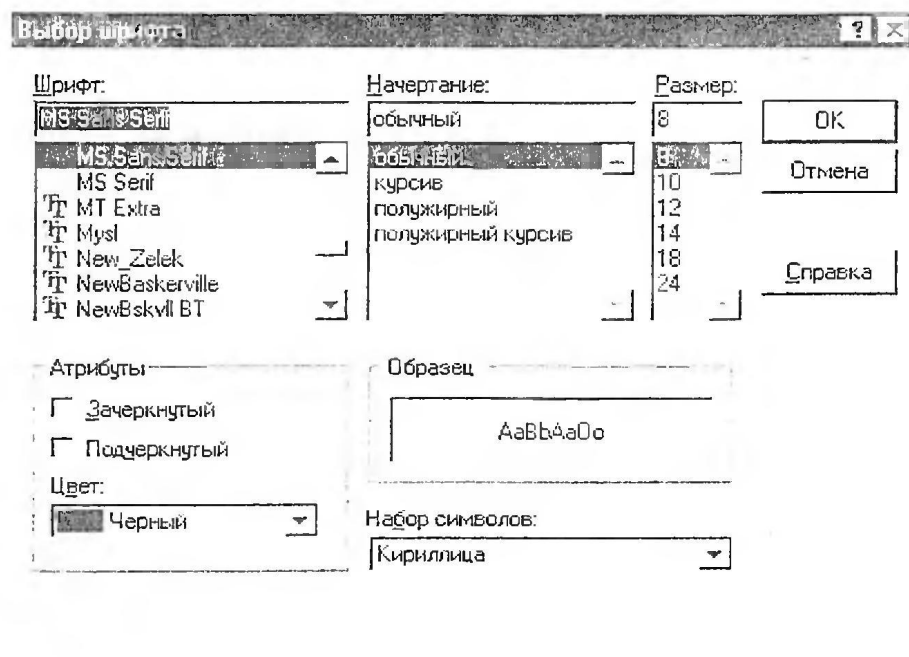
|           |                        |
|-----------|------------------------|
| Label1    | Label2                 |
| AutoSize  | False True             |
| WordWrap  | True False             |
| Alignment | taCenter taLeftJustify |

Qayd qilib o'tamizki, faqat Label2 belgi uchun AutoSize xususiyatning qiymatini True teng qilib qo'yildi, komponentlarning o'lchamlari, ya'ni Height va Width xususiyatlarning qiymatlari unga joylashtirilgan simvolli satrning uzunligiga qarab belgilanadi. Yuqorida aytib o'tilgandek, simvolli satr Label2 belgining Caption xususiyatiga qiymatni dasturni bajarish bosqichida beriladi, hozir esa, bo'lajak dasturning tashqi ko'rinishini yaxshilash uchun, Caption xususiyatidan 'Label2' qiymatni chiqarib yuborish mumkin va qiymat sifatida bir necha bo'sh simvollardan (' ') iborat bo'lgan simvolli satrni olish kerak, shunda Label2 komponent shaklda yaxshi ko'rinishli bo'ladi.

Font xususiyati murakkab xususiyat hisoblanadi va u matnlarni aks ettirishda foydalanadigan shriftlarni ta'riflaydigan har xil xarakteristikalarini aniqlaydi, masalan, shriftning balandligi, uning nomi, chizilishi va h.k. Shrift xarakteristikasini o'rnatish uchun, sichqon bilan shiqillatib, Obyektlar Inspektorida Font xususiyatini tanlaymiz va o'ng tomondagi ustunda paydo bo'lgan uch nuqtali tugmacha bo'yicha shiqillatamiz . ... . Ekranda shriftni tanlash derazasi paydo bo'ladi (1.45-

shaklga qarang). Shu deraza yordamida Label1 va Label2 belgilar uchun quyidagi shriftning xarakteristikalarini o'rnatamiz:

Label1      Label2  
 Shrift   Times New Roman   Arial  
 Yozilish shakli   Yarim qalinli   Kursivli  
 O'lcham      10      11



1.45-shakl. "Shriftni tanlash" dialogli deraza.

Ko'ramizki, Label1 va Label2 belgilari uchun shrift xarakteristikalarini o'zgartirilgandan keyin, undagi ParentFont xususiyati o'z qiymatini True dan False ga almashtirdi. Parent xususiyati berilgan komponent uchun bosh komponent-konteynerdagi shriftlar (Font xususiyati) ishlatilishini aniqlaydi. Agar ParentFont xususiyati True ga teng bo'lsa, u holda komponentda xarakteristikalar bosh komponentda berilgan shrift ishlatiladi. Aks holda bosh va o'nga bog'langan komponentlar o'z shriftlari uchun har xil xarakteristikaga ega bo'ladilar.

6. Edit1 va Button1 komponentlari uchun Height, Width, Left va Top xususiyatlar xuddi Label1 va Label2 komponentlardagi ma'noga egadirlar. Shu xususiyatlarning qiymatlarini Edit1 va Button1 komponentlar uchun keltiramiz:

Edit1 Button1  
 Height 21    25

Width 193 75  
Left 31 285  
Top 146 146

Edit1 komponentda mavjud bo'lgan AutoSize xususiyati, agar komponentdagi matnning shrifti o'zgaradigan bo'lsa, kiritish satri o'z balandligi o'zgarishini aniqlaydi. Agar kiritish satrining avtomatik ravishda kattalashini xohlasak, xususiyatning qiymatini True teng qilib qo'yish kerak, aks holda — False. Bizning misolimizda bu qiymatni aytib o'tmaslik prinsipi bo'yicha belgilang.

True qiymatni qoldiramiz.

Text xususiyati Edit1 komponenti uchun asosiy hisoblanadi va u simvolli satrlarni kiritishga (kamroq chiqarishga) mo'ljallangan. Aytib o'tmaslik prinsipi bo'yicha bu xususiyatlarning qiymati sifatida komponentning nomi 'Edit1' bo'ladi, lekin dasturni yaxshi ko'rinishli bo'lishi uchun uning qiymatini bo'sh satr ' ' qilib olamiz. Button komponenti uchun Caption xususiyatining qiymati sifatida 'Schet' simvolli satrni qo'yamiz.

7. Misolda ishlatiladigan komponentlarning barchasi — shakl, belgi, kiritish satri va tugmacha nomlarga egadir, u nomlarni ulardagi mavjud Name (Nom) xususiyati tomonidan beriladi: Form1, Label1, Label2, Edit1, Button1. Komponentlar nomlari Delphi ning integrallashtirilgan muhiti tomonidan komponent barpo etilayotganda, ya'ni, masalan, komponentni shaklga joylashtirayotganda avtomatik ravishda generatsiyalashtiriladi. Komponentning nomini tashkil qilish uchun birinchi T harfini tashlab sinfning nomidan foydalaniladi. Nomning oxirida raqam qo'shiladi, bu raqam komponent o'z sinfida qaysi bir tartib raqamli bo'lib paydo bo'lganini ko'rsatib turadi. Masalan, Edit1 komponenti TEdit sinfning nusxasi hisoblanadi, Label1 va Label2 komponentlari esa, TLabel sinfning nusxalari bo'ladi.

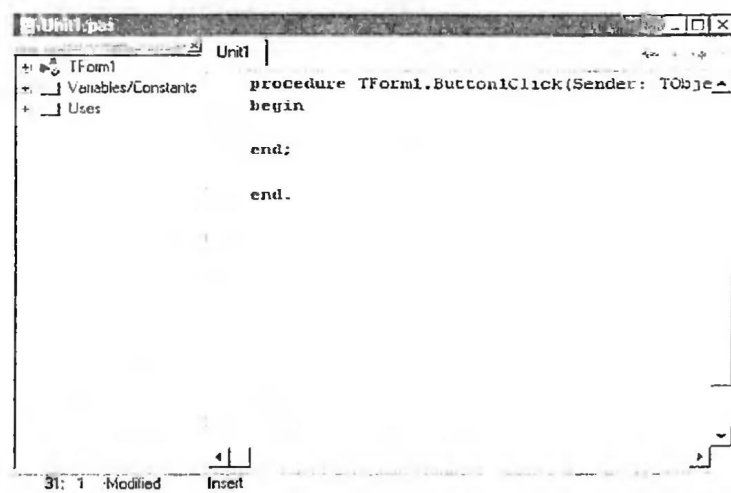
Dasturchi o'z xohishiga ko'ra Delphi tomonidan generatsiyalashtirilgan nomlarni qoldirishi yoki komponentlarga o'z nomlarini berishi mumkin. Katta yondoshlar uchun, ehtimol, qandaydir ma'noli nomlar berish maqsadga muvofiq bo'ladi. Ko'rilyotgan misollarda, bizning fikrimizcha, Delphi tomonidan berilgan nomlarni qoldirish o'rganish nuqtai nazaridan ma'qulroq bo'ladi. Shu bilan shaklni konstruksiyalash bosqichi tugatildi desak ham bo'ladi.

8. Dasturning kodini yozish bosqichiga o'tamiz.



Bizning bo'lajak dasturimiz doira yuzasini aniqlashi kerak, qachonki kiritish satriga radiusning qiymatini kiritib, sichqon bilan Schet tugmachasiga shiqillatilsa.

Bu shuni anglatadiki, biz Button1 tugmachasiga OnClick hodisa ishlovchisining tayyorlovini yozishimiz kerak bo'ladi. OnSlick hodisasi tugmacha uchun aytib o'tmaslik prinsipi bo'yicha hodisa hisoblanadi, shu sababdan hodisa ishlovchisining tayyorlovini barpo etish uchun Button1 tugmacha bo'yicha ikki marotabali shiqillatishni bajarish yetarli bo'ladi. Natijada kod muharririning derazasi faol bo'lib qoladi va hodisa ishlovchisining tayyorloviga ega bo'ladi ( 10.42-shaklga qarang).



1.46-shakl. Kod muharririning derazasi, hodisa ishlovchisining tayyorloviga ega.

Tayyorlovga doira yuzasini aniqlovchi kodni qo'shib qo'yamiz.

```
procedure TForm1.Button1Click(Sender : TObject);
var r, s : real ;
begin
r: StrToFloat (Edit. Text);
s :q pi*sqr (r ) ;
Label2.Caption :q 'Doira yuzasi teng' Q
FloatToStrF (s, ffGeneral, 7,2)
end;
```

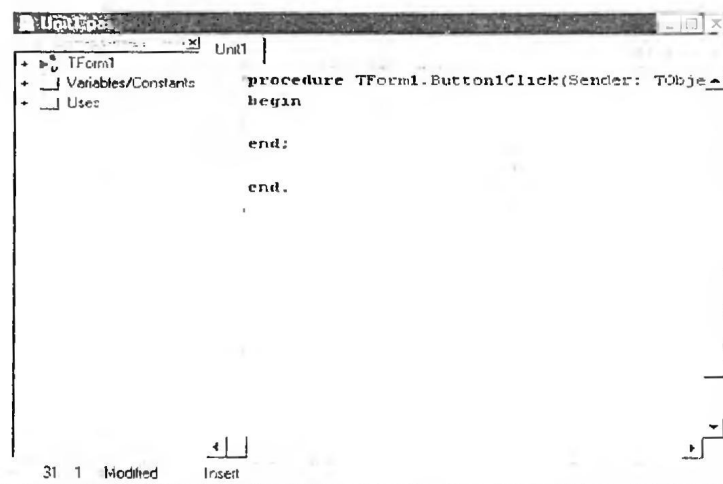
Yulduzchalar satrlarga sharhlovchi sifatida qo'shilgan, ular dasturchi tomonidan kiritilgan satrlarni ko'rsatadi. Ahamiyat bering, qanday qilib komponentning xususiyati dastur matnida foydalanilmoqda. Xususiyatga



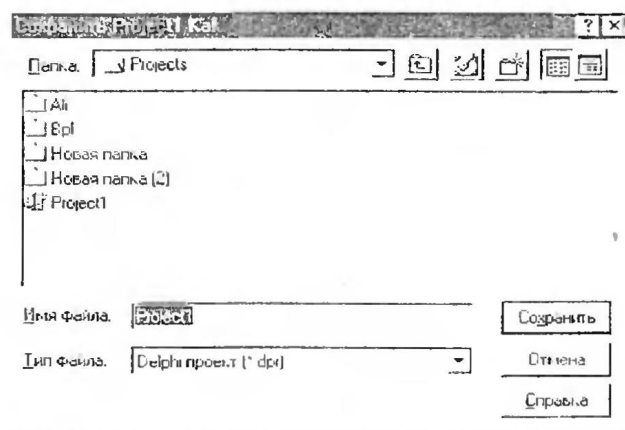
biror qiymatni berish yoki undan qiymatni olish uchun komponent nomi yozilgandan keyin, nuqta qo'yib, xususiyatni ko'rsatish kerak. Shu bilan dasturlash bosqichi tugatildi desak ham bo'ladi.

9. Dasturni ishga solishdan oldin uni har doim saqlab qolish kerak bo'ladi. Sozlanmagan dasturda "задание" hodisasi sodir bo'lsa, bu sizning vaqtingizni tejashga yordam beradi. Loyihani saqlab qolish uchun bosh menyuning File | Save | All buyrug'ini bajarish kerak. Natijada displeyning ekranida modulni saqlovchi dialogli deraza paydo bo'ladi (1.47-shaklga qarang). Shundan keyin ochiladigan Papka ro'yxatidan ishlab chiqarilayotgan loyihaga ilgari barpo etilgan papkani izlab toping va uni oching. Agar bunday papka yaratilmagan bo'lsa, uni \*\* — yangi papka barpo etish tugmachasi yordamida yaratish mumkin. Kerakli papka ochilgandan keyin, "Имя файла" (faylning nomi) deb nomlangan maydonda saqlanadigan modulning nomini ko'rsatish lozim. Aytib o'tmaslik prinsipi bo'yicha modul Unit1 nomga ega bo'ladi. Istasangiz bu nomni nisbatan ma'noliroqliniga almashtirishingiz mumkin, ammo bu misolda uni o'zgartirmasdan qoldiramiz. "Сохранить" deb nomlangan tugmachaga shiqillatib, saqlanishni tugatamiz.

Shundan keyin loyiha faylini saqlash uchun, xuddi avvalgiga o'xshagan, dialogli deraza paydo bo'ladi (1.48 shaklga qarang). Aytib o'tmaslik prinsipi bo'yicha bu faylga Project deb nom beriladi. Xuddi shu nom keyinchalik bajariladigan faylga berilishi sababli, bu yerda iloji bo'lsa ajoyibroq nom o'ylab topilsa, masalan, area\_of\_circle yoki , juda bo'lmaganda, p2\_1 yaxshi bo'lardi.



1.47-shakl. Modulni saqlash uchun dialogli deraza.



1.48-shakl. Loyiha faylini saqlash uchun dialogli deraza.

Ko'rib turibmizki, loyiha va modul fayllarining nomlari har xil bo'lishi kerak ekan.

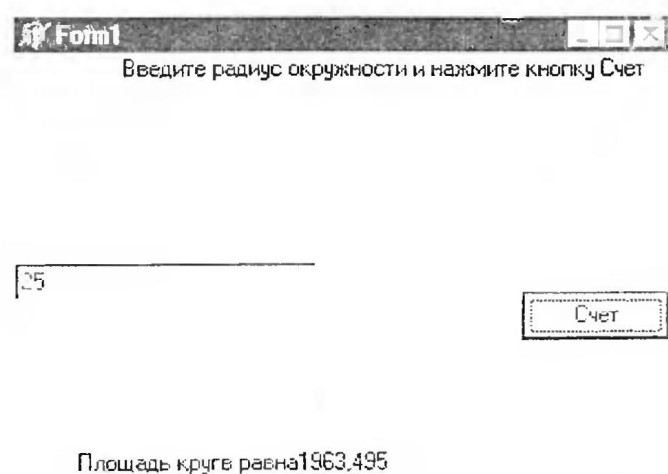
Delphining bosh menyusida saqlanish uchun mo'ljallangan boshqa buyruqlar ham bor:

Save — joriy modulni saqlash :

Save As — joriy modulni yangi nom bilan saqlash ;

Save Project As — joriy loyihani yangi nom bilan saqlash.

Loyihani saqlab turib, uni bajarishga qo'yamiz. Agar xatolar chiqadigan bo'lsa, masalan, sintaktikli, u holda ularni to'g'rilaymiz, xuddi konsolli yondoshlarga qilganimizdek. Xatolar yo'q bo'lgan holatda display ekranida bizning dasturimizning derazasi paydo bo'ladi (1.49-shaklga qarang).



1.49-shakl. Doira yuzasini aniqlovchi dasturning derazasi.

Kiritish satriga ixtiyoriy radiusni kiritamiz va "Чер" tugmachani bosamiz — derazaning pastki qismida doiraning tegishli yuzasi paydo bo'ladi.

Modul Unit1.pas ning matnini keltiramiz, u biz va Delphi tomonlaridan barpo etildi.

```
Unit1.pas modulning matni
unit Unit1 ;
interface
uses
Windows, Messages, SysUtils, Classes, Graphics,
Controls, Forms, Dialogs, StdCtrls ;
type
TForm1 = class(TForm)
Label1 : TLabel;
Label2 : TLabel;
Button1 : TButton ;
Edit1 : TEdit ;
procedure Button1Click(Sender :TObject);
private
{Private declarations}
public
{Public declarations}
end;
var
Form1 : TForm ;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender :TObject);
var r, s : real ;
begin
r : StrToFloat (Edit. Text) ;
s : pi*sqr (r ) ;
Label2.Caption :q 'Doira yuzasi teng' Q
FloatToStrF (s, ffGeneral, 7,2)
end;
end.
```

Dasturchi tomonidan kiritilgan modul satrlari qalin shrift bilan ko'rsatilgan.

Shu misoldan ko'rinib turibdiki, modulning dasturli kodini ko'p qismi avtomatik ravishda Delphi tomonidan barpo etilgan. Keyinchalik ko'riladigan misollarda dasturchi tomonidan kiritiladigan kod satrlarini qalin shriftli qilib yozib boramiz.

Agar loyiha va modul fayllarini saqlab qolgan papkaga ko'z tashlasak, u yerda bir qancha fayllar guruhini ko'ramiz, ularning bir qismi konsolli yondoshlarni barpo etish bo'yicha bizga allaqachon ma'lum, boshqalari to'g'risida 10.7.2-bandda so'zlangan edi. Loyihaga kirgan fayllar ichida biz uchun yangilari:

|            |   |                                                |
|------------|---|------------------------------------------------|
| Unit1.pas  | - | dasturli kodlarni saqlash uchun modul fayli    |
| Unit1.dfm  | - | shaklga ega fayl                               |
| Unit1.dcu  | - | modulning kompilyatsiyalashtirilgan fayli      |
| Unit.~ pas | - | modulning rezervli nusxasi                     |
| Unit.~ dfm | - | shaklning rezervli nusxasi                     |
| p2_1.res   | - | loyiha tomonidan ishlatiladigan piktogrammalar |

va boshqa resurslarni saqlovchi fayl

## 1.8. SINFLAR VA OBYEKTLAR

### 1.8.1. OBYEKT BO'YICHA ORIENTATSIYALANGAN DASTURLASHNING ASOSIY PRINSIPLARI

Object Pascal obyekt bo'yicha oriyentatsiyalangan dasturlash til hisoblanadi. Tilda obyekt bo'yicha oriyentatsiyalangan imkoniyatlar to'plangan, uni dasturlash tilning obyektli modeli deb nomlaydilar. Obyektli modelni ishlatishdan kelib chiqadigan amaliy natijasi sifatida komponentlarni barpo etish va qo'llashni ko'rsatish mumkin. Bu bobda Object Pascalning nazariy asoslarini ko'rib chiqamiz, ular asosida sinf va obyekt tushunchalari yotibdi.

Object Pascalda sinf deb maxsus tiplarga aytiladi, ular maydon, usul va xususiyatlarga ega.

Obyekt — bu sinfning muayyan nusxasi va, boshqa o'zgaruvchilar singari, u dasturning var bo'limida ta'riflanadi.

Sinflar asosida uchta fundamental prinsiplar yotadi — инкапсуляция, наследование и полиморфизм.

Инкапсуляция (birlashtiruv ma'nosini beradi) deb berilganlar va dastur qismlarini ishlov berish uchun sinflarga birlashtirishga aytiladi. Berilganlar sinfning maydoni ichida joylashadi, protsedura va funksiyalar esa ularga ishlov berish uchun usullarni belgilaydilar. Obyekt bo'yicha oriyentatsiyalangan dasturlash qoidalariga asosan sinfning maydonlariga to'g'ridan-to'g'ri kira olish ma'qul emas. Shu sababli Object Pascalda xususiyat deb nomlanadigan maxsus konstruksiyalar mavjud, ular tegishli usullarni chaqirish yordamida maydonlarga o'qish yoki yozish amallarini bajaradilar.

Inkapsulyatsiya aniq funksionallikka ega bir butunlik sifatidagi sinfni barpo etishga imkoniyat yaratadi. Misol sifatida Delphida aniqlangan standart sinflarni keltirish mumkin:

TEdit — kiritish satrlarini barpo etish va boshqarish ishlariga imkon beradi ;

TLabel — belgilarning faoliyatini ta'minlaydi ;

TButton-tugmachaning ishiga barcha kerakliklarni ichiga olgan.

Qabul qilingan umumiy qoidaga asosan sinf nomlari T harfi bilan boshlanishi kerak.

Наследование (nasldan-naslga o'tish) — ma'nosi shundan iboratki, har qanday sinf boshqa sinfdan vujudga kelishi mumkin. Biror urug'dosh sinfdan yangi sinf barpo etilishini quyidagi dasturli kod yordamida amalga oshirish mumkin :

TNewClass q class (ToldClass)

Nasldan-naslga o'tishning ajoyib prinsipi shundan iboratki, yangi vujudga kelgan sinf o'z urug'doshidan avtomatik ravishda uning maydonlar, usullar va xususiyatlarini meros sifatida qabul qilib oladi va ularni yanada yangilar bilan to'lg'azishi mumkin. Natijada dasturchi Delphida mavjud bo'lgan standartli sinflardan foydalanib, xususiy sinflarni va hattoki sinflar kutubxonasini barpo etishi mumkin.

Object Pascalda barcha sinflar TObject sinfning avlodlari hisoblanadi. Bu sinf o'z tarkibiga maydon va xususiyatlarni kiritmagan bo'lsa ham, biroq uning usullari obyektlarning faoliyatini ushlab turadi va obyektlarni chiqarib yuboradi. Agar dasturchi TObject sinfning bevosita avlodi bo'lgan sinfni aytmoqchi bo'lsa, u holda urug'dosh sinfning nomini ko'rsatmasa ham bo'ladi, ya'ni quyidagi satrlar bir-biriga ekvivalent:

TSecondClass q class (TObject)

TSeconClass q class

Delphida nasldan-naslga o'tish prinsipidan foydalanish natijasi tarmoqlangan sinflar daraxtini barpo etilishiga olib keldi. Shu daraxtning yuqori qismida abstraktli deb nomlanadigan sinflar joylashgan, ular uchun to'la ishlaydigan obyektlar barpo etila olmaydi. Shu bilan birga abstraktli sinflar katta sinflar guruhi uchun urug' boshlig'i hisoblanadilar, ular esa real obyektlarni barpo etadilar.

Полиморфизм (ko'p shakllik xususiyat) har xil sinflarga kirgan usullarga bir xil nom ishlatish imkonini beradi. Polimorfizm prinsipi bir xil nomlangan usullarga murojaat bo'lgan holatda, ularni ichidan obyekt sinfiga tegishli bo'lganini bajarishni ta'minlaydi.

Masalan, biz yangi sinf barpo etishga qaror qildik, u urug'dosh bo'lgan sinfdan biror usulida o'zgargan algoritmi bilan farqlanadi. Bunday holatda avlod-sinfida tegishli usulni qaytadan qoplamoq lozim bo'ladi, ya'ni avlod-sinfida bir xil nomlangan usulni e'lon qilish va u yerda kerakli algoritmi yozish kerak bo'ladi. Natijada biz bir xil nomlangan, lekin turlicha bajariladigan ikkita usul olamiz.

Sinflar Type bo'limida quyidagi umumiy ko'rinishda e'lon qilinadi:

Type

<sinf nomi>q class(<urug'dosh sinf nomi>)

public

<ommabop elementlarning ta'rifi>

published

<Obyektlar Inspektoriga foydalanish huquqiga ega bo'lgan elementlarning ta'rifi>

Protected

<avlod-sinflarga kira olish imkoni bo'lgan elementlarning ta'rifi>

private

<faqat modullarga kira olish imkoni bo'lgan elementlarning ta'rifi>

end;

Public, published, protected, private seksiyalari maydonlar, usullar, xususiyatlar va hodisalarning ta'rifini o'z ichlariga olishlari mumkin.

### 1.8.2. MAYDONLAR

Maydonlar deb sinfdan inkapsulyatsiyalashtirilgan berilganlarga aytiladi. Maydonlar, yozuvlar maydonlariga o'xshash, lekin ulardan farqlanishi shundan iboratki, ular ixtiyoriy tipli bo'lishlari mumkin, shu qatori sinflar, masalan:

```

Type
TChildClass q class
FOne : Integer ;
FTwo : String ;
FThree : Tobject ;
end;

```

Inkapsulyatsiyadan kelib chiqqan holda, maydonlarga bo'ladigan murojaatni sinfning usul va xususiyatlari yordamida amalga oshirish mumkin. Shu bilan birga, Object Pascalda maydonlarga bevosita murojaat qilishga ruxsat etiladi. Maydonga murojaat etish uchun sinf va maydon nomlari nuqta belgisi bilan ajratilgan tuzma nomni yozish kerak bo'ladi, masalan:

```

var
MyObject : TChildClass ;
begin
MyObject.FOne :q 16;
MyObject.Two :q 'Biror satrli qiymat' ;
end;

```

Shuni nazarda tutish kerakki, odatda maydon nomi mos xossa nomi bilan o'xshash bo'ladi, ammo maydonni nomlashda birinchi harf sifatida F qo'shiladi.

### 1.8.3. USULLAR

Usullar deb sinfda inkapsulyatsiyalashtirilgan protsedura va funksiyalarga aytiladi. Masalan:

```

Type
TChildClass q class
FOne : Integer ;
FTwo : String ;
FThree : TObject ;
function FirstFunc(x:real):real ;
procedure SeconProc ;
end;

```

Usullarga murojaat qilish uchun, maydonlar singari, tuzma nomdan foydalanish kerak :

```

var
MyObject : TchildClass ;
y :real ;

```



```
begin
```

```
.....  
MyObject.SecondProc ;  
.....
```

```
end ;
```

Sinfda aniqlangan usullar statik, virtual, dinamik yoki abstraktli bo'lishlari mumkin. Usulning tipi uni avlodlarda qayta qoplash mexanizmi bilan aniqlanadi.

Statik usullar uchun qayta qoplashni kompilyator amalga oshiradi. Masalan, bizda Tbase urug'dosh sinfini va bir xil nomlangan MyJoy usuliga ega uning avlodi Tdescendantning ta'rif bor deylik:

```
Type
```

```
Tbase q Class
```

```
    procedure MyJoy ;
```

```
end;
```

```
TDescendant q class(TBase)
```

```
    procedure MyJoy ;
```

```
end;
```

```
var
```

```
FirstObject : TBase ;
```

```
SecondObject : TDescendant ;
```

```
begin
```

```
.....  
FirstObject.MyJoy ;
```

```
SecondObject.MyJoy ;  
.....
```

```
end;
```

Polimorfizm prinsipiga asosan FirstObject.MyJoy operatorida TBase sinfida ta'riflangan usul chaqirilmoqda, operator SecondObject.MyJoy da esa Tdescendant sinfida ta'riflangan usulga murojaat qilinmoqda.

Aytib o'tmaslik prinsipi bo'yicha sinfda ta'riflangan hamma usullar statikli hisoblanadi.

Dinamik va virtuallik usullar statikli usuldan shu bilan farqlanadiki, avlodli usullarni urug'doshliklar bilan o'rnini bosish dasturni bajarish bosqichida amalga oshiriladi. Virtuallik usulni urug'dosh sinfida e'lon qilish uchun rezervlangan virtual so'zi, dinamik usuli uchun esa -

rezervlangan dynamic so'zi ishlatiladi. Avlod-sinfda o'rnini bosuvchi usulning sarlavhasida rezervlangan override so'zini ishlatmoq lozim. Masalan:

```
Type
Tbase q class
    procedure MyJoy ; virtual ;
end;
TDescendant q class(TBase)
    procedure MyJoy; override ;
end;
var
FirstObject : TBase ;
SecondObject : TDescendant ;
begin
.....
FirstObject.MyJoy ;
SecondObject.MyJoy ;
.....
end ;
```

Agar MyJoy usuli Tbase sinfida dinamikli bo'lishini istasak, protseduraning sarlavhasidagi virtual so'zini dynamic so'ziga almashtirish lozim bo'ladi. Virtual va dinamik usullari orasida farq katta emas va ularni chaqirishni amalga oshirish xususiyatlariga bog'liqdir. Shunday aytish mumkin, vaqt sarflanishi nuqtai nazaridan virtualli usullar natijaliroq, dinamikli usullar esa tezkor xotirani ratsional ravishda foydalanishga imkoniyat yaratadi.

Abstraktli deb virtual yoki dinamik usullarga aytiladi, ular sinfda aniqlanadi, lekin hech qanday amallarga ega emas, hech qachon chaqirilmaydi va albatta avlod-sinflarida qaytadan aniqlanishi lozim. Abstraktli usul rezervlangan abstract so'zi yordamida e'lon qilinadi, qachon virtual yoki dynamic so'zlarning ketidan yozish kerak, masalan:

```
procedure Mymetod ; virtual ; abstract;
```

Abstraktli usulni ishlatishdan asosiy maqsad — sinf avlodlarida iyerarxik urug'dosh boshlig'i bo'lish.

Har qanday sinfda ikkita maxsus usul bor — konstruktor va destruktor. Bu usullarga barcha qolgan sinflarning sinf-urug' boshlovchisi — Tobject ega va, demak, avlodlarga nasldan-naslga o'tadi. Boshqa usullar

singari, ular avlod-sinflarida o'zgarishi mumkin, ya'ni bir-birini qoplashi. TObject sinfida va uning avlodlarini ko'pida konstruktor va destruktorni mos ravishda Greate va Destroy deb nomlanadi.

Konstruktorlar obyektни barpo etish va initsialashtirish uchun ishlatiladi. Gap shundaki, Object Pascal tilida obyekt dinamikli struktura hisoblanadi va o'zgaruvchi-obyekt berilganlarning o'zini emas, ularga bo'lgan havolaga ega. Konstruktor obyektни dinamik xotirada taqsimlaydi va obyekt maydonlariga boshlang'ich qiymatlar beradi. Shunda tartibli tipli maydonlar boshlang'ich qiymat sifatida 0 ni oladi, satrlilar — bo'sh satrni, maydon ko'rsatkchlari — nil qiymatni, maydon-variantlar — Unassigned. Bundan tashqari, konstruktor havolani barpo etilgan obyektда avtomatik ravishda e'lon qilinadigan Self o'zgaruvchisiga qo'yadi. Aytilganlardan shu kelib chiqadiki, obyektning maydonlari, xususiyatlari va usullariga murojaat qilish konstruktor chaqirilganidan keyin bajariladi.

Destruktor dinamik xotirani bo'shatadi va obyektни buzadi.

Konstruktor va destruktorglarni e'lon qilish uchun rezervlangan mos ravishda constructor va destructor so'zlari ishlatiladi. Masalan:

type

Tsample q class

Text : string ;

constructor Greate ;

destructor Destroy ;

end ;

Obyekt barpo etish obyekt sinfiga usul-konstruktorни qo'llash kerak:

var

MyObject : TSample ;

begin

.....

MyObject :q TSample.Greate ;

.....

end;

Agar sinf-avlod barpo etilayotgan bo'lsa va uni barpo etilayotganda ayrim qo'shimcha amallar bajarish rejalashtirilgan bo'lsa, qaysilar sinf-urug'doshida yo'q bo'lsa, u holda sinf-avlod konstruktorida o'z urug'dosh konstruktorini chaqirish kerak, shundan keyingina qo'shimcha amallarni bajarish kerak bo'ladi. Har qanday qayta qoplangan urug'dosh

sinfidagi usulni chaqirish uchun rezervlangan inherited (naslga o'tgan) so'zi yordamida amalga oshiriladi. Masalan, agar TDescedant sinfida o'z xususiy konstruktoringiz bor bo'lsa,

Type

TDescedant q class(Tbase)

Fmark : Boolean ;

constructor Greate(Mark :Boolean) ;

end;

u holda uni amalga oshirish mana bunday bo'lishi ham mumkin :

constructor TDescedant.Greate(Mark:Boolean);

begin

inherited Greate ;

Fmark :q Mark ;

end;

bu yerda urug'dosh konstruktorini chaqirish

inherited Greate; operatori yordamida amalga oshirilmoqda

Fmark :q Mark ; operatori esa qo'shimchali amallarni bajarmoqda.

Destroy destruktordan tashqari, asosli TObject sinfida Free usuli aniqlangan, u avval tekshirib ko'radi — haqiqatan obyekt amalga oshirildimi va faqat shundan keyingina Destroy usulni chaqiradi. Agar obyekt konstruktor tomonidan barpo etilmagan bo'lsa, u holda destruktorga bo'lgan murojaat maxsus vaziyatni generatsiyalashtirilishiga olib keladi. Demak, kerakmas obyektни yo'qotish uchun Free usuli bilan foydalanish qulayroq bo'ladi, masalan:

MyObject.Free ;

Sinfda shunday usullar aniqlanishi mumkinki, qaysi obyektни barpo etmasdan va initsializatsiyasini o'tkazmasdan ularni chaqirtirish mumkin bo'ladi. Bu usullarni sinf usullari deyiladi va ularni e'lon qilish uchun rezervlangan class so'zi ishlatiladi. Masalan :

Type

TChildClass q class(Tobject)

class function ChildClassInfo : string ;

end;

var

y : string ;

begin

.....

```
y :q ChildClassInfo ;
```

```
.....
```

```
end;
```

Odat bo'yicha, sinf usullari sinf to'g'risidagi ma'lumotnomali axborotni olish uchun mo'ljallanganlar — sinf nomi, sinf ajdodlari, sinf o'lchamlari va h.k.

#### 1.8.4. XUSUSIYATLAR

Tashqi ko'rinishi bo'yicha xususiyatlar sinf maydonini eslatadi, haqiqatda esa ular maydonga bo'lgan kira olishning sozlovchi mexanizmi vazifasini bajaradilar. Odatda, xususiyat qandaydir sinf maydoni bilan bog'langan va u shunday sinf usullarini ko'rsatib turadiki, qaysilarni shu maydondan o'qish yoki yozish paytida ulardan foydalanish kerak. Agar o'qish uchun kira olish aniqlangan bo'lsa, u holda u parametrsiz funksiya bo'lishi kerak, bu funksiya qiymatni xuddi xususiyat tipinikiday qilib qaytaradi. O'qish uchun mo'ljallangan funksiyaning nomini old qo'shimcha Get so'zi bilan boshlashga kelishilgan, nimaning ketidan xususiyat nomi beriladi. Yozish uchun foydalanadigan usul bir parametrlil protsedura bo'lishi kerak. Bu parametrning tipi xuddi xususiyatnikiday bo'lishi lozim. Yozuvga mo'ljallangan protseduraning nomi old qo'shimchaci Set so'zi bilan boshlanib, keyin xususiyat nomi beriladi.

Xususiyatni e'lon qilish uchun rezervlangan property, read va write so'zlari ishlatiladi. So'zlar read va write tegishli ravishda o'qish va yozish uchun mo'ljallangan usullariga ega bo'limlar boshini bildiradi. Masalan:

```
type
```

```
Tstudent q class
```

```
Fage : integer ;
```

```
function GetAge : integer ;
```

```
procedure SetAge (Value : integer) ;
```

```
property Age : integer read GetAge write SetAge
```

```
end;
```

Bu yerda Age — xususiyat, Fage maydoni bilan bog'langan ;

GetAge va SetAge — usullar, ular Fage maydoniga mos ravishda o'qish va yozish uchun mo'ljallanganlar.

Dastur matnida xususiyatga murojaat qilish uchun, shu qatori maydon va usullarga, obyekt nomi, nuqta belgisi va xususiyat nomidan iborat

bo'lgan tuzma nomni ishlatish kerak, masalan:

```
var
  GoodStudent : Tstudent ;
  HisAge : integer ;
begin
  GoodStudent :q Tstudent.Greate ;
  GoodStudent.Age :q 19 ;
  .....
  HisAge :q GoodStudent.Age ;
  .....
  GoodStudent.Free ;
end;
```

Xususiyatlardan foydalanish, maydonlardan bevosita foydalanishdan shu bilan farqlanadiki, u har xil qo'shimcha amallarni bajarish imkonini beradi. Masalan, siz FAage maydoniga kerakli qiymatni bevosita joylashtirishingiz mumkin edi:

```
GoodStudent.FAage :q 19 ;
```

Lekin, Age xususiyatidan foydalanib, o'qish va yozish uchun mo'ljallangan GetAGE va SetAge usullarida har xil qo'shimcha amallar bajarishimiz mumkin: kiritilgan qiymatni berilgan diapazonga tegishligini tekshirish, ekranga xabarnoma chiqarish, ekranda obyektning tashqi ko'rinishini o'zgartirish va h.k.

Maydonga o'qish yoki yozish holatlarida qo'shimcha amallar bajarish kerak bo'lmasa, tegishli usulning nomi o'rniga maydon nomini ko'rsatish mumkin:

```
type
  TStudent q class
  FAge : integer ;
  procedure SetAge (Value : integer ) ;
  property Age : integer read FAge write SetAge ;
end;
```

Maydonlar faqat o'qishga yoki faqat yozishga kira olish imkoniyatiga ega bo'lishlari mumkin. Bunday hollarda xususiyatning ta'rifida read yoki write bo'limlarni tashlab o'tiladi. Masalan, agar biz Age xususiyati faqat o'qish uchun kira olishini xohlasak, u holda write bo'limini olib tashlash kerak:

```

type
TStudent q class
FAge : integer ;
function GetAge : integer ;
property Age : integer read  GetAge ;
end;

```

Ko'rib turibmizki, xususiyat muayyan maydon bilan bog'lanmasligi ham mumkin. Bunday holda xususiyatlarda oddiy ikki usul aniqlanadi, ular berilganlar bilan xususiyat bajaradigan ayrim amallarni bajaradilar.

### 1.8.5. HODISALAR

Ilgari 1.7.1-bandda aytib o'tilgan ediki, hodisa — bu dasturning ish jaryonida vujudga keladi. Delphida bir necha o'nlab namunali hodisalar aniqlangan. Object Pascal tili nuqtayi nazarida hodisa qanday ifodalanishini ko'rib chiqamiz.

Hodisa — bu protsedura tipli xususiyat va uning qiymati esa biror usulga ko'rsatkich hisoblanadi. Bunday xususiyatga qiymat berish deganda usulning adresini ko'rsatish ma'nosini bildiradi. Bu usul hodisa boshlanishi bilan bajarilishiga o'tadi. Bunday usullar, ilgari aytilgandek, hodisalarning ishlovchisi deb ataladi. Misol tariqasida, OnDblClick (sichqonni chap klavishasi ikki marotaba shiqillatishdan kelib chiqadigan), OnMouseDown (sichqon klavishasi osilganda kelib chiqadigan) va OnMouseMove (sichqonni siljirilgan kelib chiqadigan) standartli hodisalarni qanday qilib TSontrol sinfida ta'riflanishini ko'rib chiqamiz:

```

TControl q class (TComponent)

```

```

    private

```

```

    .....
    FOnDblClick : TNotifyEvent ;

```

```

    FOnMouseDown : TMouseEvent ;

```

```

    FOnMouseMove : TMouseMoveEvent ;
    .....

```

```

protected

```

```

    .....
    property OnDblClick : TNotifyEvent read FOnDblClick write
    FOnDblClick ;

```



```

property OnMouseDown : TMouseEvent read FOnMouseDown write
FOnMouseDown ;
property OnMouseMove : TMouseMoveEvent read FOnMouseMove
write
FOnMouseMove ;
.....
end;

```

Rezervlangan private va protected so'zlarining xizmatini keyinroq ko'rib chiqamiz, hozircha esa quyidagini qayd qilamiz.

Xususiyat-hodisani ta'riflanishida ishlatiladigan FOnDbClick, FOnMouseDown va FOnMouseMove maydonlari usulga bo'lgan ko'rsatkichlarni saqlash uchun mo'ljallanganlar, ular tegishli hodisaning ishlovchisi hisoblanadi. Maydonlarni ta'riflashda ishlatiladigan protsedurali tiplar quyidagicha aniqlanadi:

type

TNnotifyEvent q procedure (Sender : TObject) of object ;

TMouseEvent q procedure (Sender : TObject ; Button : TMouseButton ;

Shift : TShiftState ; x, y : Integer) of object ;

Barcha protsedurali tiplar uchun umumlik obyektga bo'lgan ko'rsatkichga ega Sender parametrini ishlatish — hodisa manbayi. Button parametri qaysi bir sichqon klavishi bosilgani to'g'risidagi axborotga ega. Shift parametri klaviaturada bosilgan klavishaning kodiga ega. X va Y — komponent atrofida sichqonning joriy koordinatalari.

Usulning oddiy protseduradan farqi shundan iboratki, usulga har doim aniq ta'riflangan parametrlar ko'rsatishdan tashqari, yana noaniq ravishda uni chaqirgan obyektga ko'rsatkich ham uzatiladi, qaysi sinfda avtomatik ravishda ta'riflangan Self o'zgaruvchisiga joylashtiriladi. Yuqorida ta'riflangan protsedura tiplari aynan usullar bilan ishlash uchun mo'ljallangan, ya'ni usullar tiplari hisoblanishini ta'kidlab o'tish uchun rezervlangan of object so'zlari qo'llaniladi.

Obyektlar Inspektoridagi Events betida faqat usul tipi, ya'ni hodisaga ega bo'lgan komponent xususiyatlari ifodalanadi. Hodisa ishlovchining tipi, qaysini biror hodisaga mos ravishda ko'riladi, xuddi hodisa tipidek bo'lishi kerak, shuning uchun usulning tipi boshqachaaytganda hodisa ishlovchisining tipi deb ham aytiladi.

### 1.8.6. SINF ELEMENTLARINING KO'RINISH SOHALARI

Har bir sinf to'rtta seksiyaga ega bo'lishi mumkin, ular rezervlangan published (deklaratsiya qilingan), private (shaxsiy), protected (himoyalangan) va public (kira olishli) so'zlari orqali aniqlanadi. Seksiyalar sinfga kirgan elementlarni ko'rinish sohasini aniqlaydi. Har bir seksiya ichida avval maydonlar ta'riflanadi, keyin esa — usullar, xususiyatlar va hodisalar.

Public seksiyasi e'lon qilinayotgan sinfni o'zida mujassamlashtirgan ma'lum modulga kirishga ega bo'lgan, har qanday dastur yoki modulda ko'rinadigan elementlardan tarkib topgan. Nisbatan qat'iy seksiya private hisoblanadi, u sinf joylashgan modul ko'rinishini chegaralaydi, Private seksiyasi faqat haqiqatda sinfga bog'liq bo'lgan va barcha hosila sinflardan yashirinishi zarur bo'lgan maydon va usullarni o'zida mujassamlashtirishi kerak.

Protected seksiyasida joylashgan elementlar modul ichida public va private seksiyalari qoidalariga bo'ysunadi. Ammo modul tashqarisida himoyalangan elementlarga kirishga faqatgina sinf-avlodlar usullari ega.

Published seksiyasida ko'rinish qoidasi public seksiyasi qoidalari bilan o'xshash. Farqi shundaki, published seksiyasi komponentlarni e'lon qilish uchun maxsus ishlab chiqilgan. Bu seksiyaga kiritiluvchi xususiyatlar dasturni Obyektlar Inspektori derazasida tuzish bosqichida tushunarli bo'ladi. Shuning uchun published seksiyasida e'lon qilingan maydonlar sinf tipiga ega bo'lishi zarur. Shaklning sinfini e'lon qilish boshida joylashgan va nomga ega bo'lmagan seksiya aytib o'tmaslik prinsipi bo'yicha published deb hisoblanadi. Har qanday seksiya sinfni ta'riflashda ixtiyoriy barpo etilishi mumkin. Seksiyalarning birin-ketin kelishi ham ixtiyoriy.

1.7.3-bandagi 1.7.1-misolga qaytib va obyekt bo'yicha oriyentatsiyalangan dasturlash atamasini ishlatib, quyidagini aytish mumkin.

Shaklni konstruksiyalash jarayonida TForm1 sinfi barpo etildi. Bu sinf Delphida aniqlangan standartli TForm sinfining avlodi hisoblanadi. TForm1 sinfida Label1, Label2, Button1, Edit1 maydonlari aniqlangan. Bu hamma maydonlar sinf tipiga ega. Bundan tashqari, TForm1 sinfida Button1Click usul ta'riflangan, u Button1 tugmachasi uchun OnClick hodisa ishlovchisi hisoblanadi. TForm1 sinf maydonlari va usulning

ta'riflari published seksiyasida joylashtirilgan va ular Obyektlar Inspektorining derazasiga kira olishga imkoniyatli.

## 2. KOMPYUTYERDA SONLI HISOBLASH USULI BILAN MASALALARNI YECHISH

### 2.1. ALGEBRAIK VA TRANSENDENT TENGLAMALAR

Algebraik va transsendent tenglamalarni yechish riyoziy analizning muhim masalalariga kiradi. Fizika, mexanika, texnika va umuman texnikada xilma-xil masalalarning hal qilinishida bunday tenglamalarni yechishga to'g'ri keladi.

Faraz qilaylik, algebraik yoki transsendent tenglamani yechish talab qilingan. Uning umumiy ko'rinishda yozilishi quyidagicha:

$$f(x) = 0, \quad (21)$$

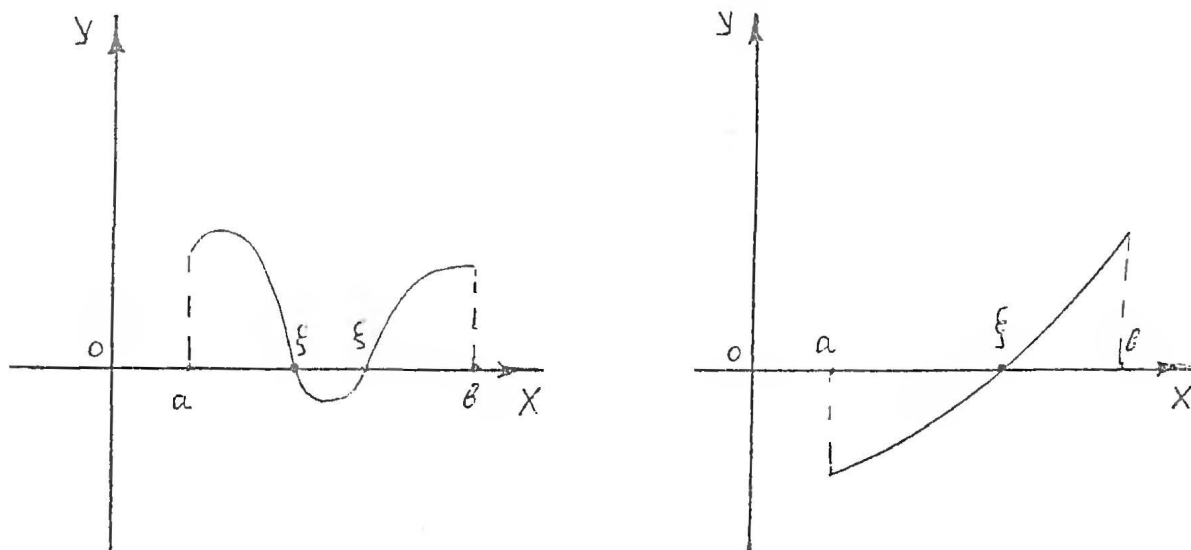
Bu yerda  $f(x)$  funksiyasi berilgan biror  $a < x < b$  oraliqda aniq va uzluksiz bo'ladi.

Ko'pgina taqribiy hisoblash usullarida tenglamaning ildizlari joylashgan oraliqni aniqlash kerak bo'ladi. Quyidagi teorema tenglamaning ildizlari yotgan oraliqni ajratishga yordam beradi.

1. Arap uzluksiz bo'lgan  $f(x)$  funksiyasi biror  $[a, b]$  oraliqni chetki nuqtalarida har xil ishorali qiymatlarga ega bo'lsa (2.1-shakl), u vaqtda mazkur oraliqda (21) tenglamaning hech bo'lmaganda bitta ildizi mavjuddir. Shu bilan birga, agar funksiyaning birinchi va ikkinchi tartibli hosilalari  $f'(x)$ ,  $f''(x)$  lar mavjud bo'lib ular o'z ishorasini saqlasa, u holda bu oraliqda ildiz yagonadir.

Odatda, asosan  $a$  va  $b$  nuqtalar berilgan bo'ladi, lekin, ba'zan ularni belgilash zarurati paydo bo'ladi.

$a, a_1, \dots, a_n$  ( $a_0 > 0$ ) haqiqiy koeffitsiyentlariga ega bo'lgan algebraik tenglamalarning haqiqiy ildizlari chegaralarini aniqlash uchun Lagranj formulasi, o'zgaruvchi ishorali yig'indilar usuli va Nyuton usulidan foydalanish mumkin.



2.1-shakl.

### 2.1.1. LAGRANJ FORMULASI

Lagranj formulasi quyidagi ko'rinishga ega:

$$R = 1 + \sqrt[k]{B/a_0} \quad (2.2)$$

bunda  $R$  — haqiqiy ildizlar chegarasi;

$K$  ( $K \geq 1$ );  $f(x) = a_0x_n + \dots + a_n$  polinomi birinchi manfiy koeffitsiyentini tartib raqami;

$V$  —  $f(x)$ ning absolut qiymati bo'yicha eng katta miqdor bo'lgan manfiy koeffitsiyenti.

$R_{yu}^+$ ;  $R_p^+$ ;  $R_{yu}^-$ ;  $R_p^-$  — orqali tenglamaning ildizlarini mos ravishda yo'qori va pastki chegaralarini belgilaymiz. Uning barcha  $X_g^+$  musbat ildizlari  $X + < R_{yu}^+$  tengsizlikni qoniqtiradi, bu yerda:

$$R_{yo}^+ = 1 + \sqrt[k]{B/a} \quad (2.3.)$$

$f(x)=0$  tenglamasining  $R_p^+$  haqiqiy musbat ildizlari hamda yordamchi tenglamalarning  $R_{yu}^-$ ,  $R_p^-$  manfiy ildizlarini chegarali (2.3.) formula bo'yicha aniqlash mumkin

$$\begin{aligned}
f_1(x) &= X^n f(1/X) = 0 \\
f_2(x) &= f(-x) = 0 \\
f_3(x) &= X^n f(-1/X) = 0 \\
(1/R_1) &\leq X^+ \leq R_0^+ \\
-R_2 &\leq X^- \leq (1/R_3)
\end{aligned}
\tag{2.4}$$

Misol:

$2X^5 - 100X^2 + 2X - 1 = 0$ , bu misolda  $a_3 = -100$

$$R_0^+ = 1 + \sqrt[3]{100/2} = 4,7$$

$$b = 4,7$$

Pastki chegarani topish uchun tenglamani (2.4) ga ko'ra o'zgartiramiz.

$$\begin{aligned}
\frac{2}{X^5} - \frac{100}{X^2} + \frac{2}{X} - 1 &= 0 \\
-X^3 + 2X^4 - 100X^3 + 2 &= 0 \\
X^5 - 2X^4 + 100X^3 - 2 &= 0 \\
R_0^+ &= 1 + \frac{2}{1} = 3 \\
a &= \frac{1}{R_0^+} = \frac{1}{3} \approx 0,3
\end{aligned}$$

Shunday qilib, (2.2) ga asosan tenglamaning ildizlari  $0,3 \leq X \leq 4,7$  oraliqda mavjuddir.

### 2.1.2. O'ZGARUVCHI ISHORALI YIG'INDILAR USULI

Lagranj usulining g'oyasi quyidagi tarzda umumlashtirilishi mumkin:

$R(x)$  polinomidagi hadlar o'zgaruvchi  $x$  ning darajasi kamayishi bo'yicha joylashgan; shu bilan birga, eng katta koeffitsiyenti  $a_0 > 0$  bo'lsin. Polinom  $R(x)$  ning ishorasini o'zgaruvchi yig'indilar orqali ifodalaymiz:

$$P(x) = Q_1(x) - Q_2(x) + Q_3(x) - Q_4(x) + \dots + Q_{2m-1}(x) - Q_{2m}(x)$$

bu yerda:  $Q_1(x) - a_0 x_n$  hadidan boshlab musbat ishorali qo'shiluvchilarning yig'indisining ketma-ketligi;

$Q_2(x)$  manfiy ishorali ko'shiluvchilarni ketma-ketligi, bevosita musbat ishoraligidan keyin keluvchilardan tashkil topadi va hokazo. Shu bilan birga, yoki aynan nolga teng oxirgi qo'shiluvchi manfiy ishorali bo'lishi mumkin.

$C_j$  ( $j = 1, 2, \dots, m$ ) orqali musbat sonlarni belgilaymiz, qaysilar quyidagi shartga bo'ysunishsa:

$$Q_{2j-1}(C_j) - Q_{2j}(C_j) \geq 0 \tag{2.5}$$

Bunday holda tenglamaning musbat ildizlari joylashgan oraliqning yo'qori chegarasi sifatida quyidagi sonni qabul qilishimiz mumkin:

$$R_{yu}^+ = \max(C1, S2, \dots, Cm) \quad (2.6)$$

$$\begin{aligned} \text{Tenglama uchun } 2x^5 - 100x^2 + 2x - 1 &= 0 \\ 2x^5 - 100x^2 + 2x^2(x^3 - 50) &\geq 0 \\ x^3 - 50 &\geq 0; x = 50^{1/3}; x \geq 3,7 \\ 2x - 1 &= 2(x - 1/2) > 0; x > 0,5 \end{aligned}$$

$$\text{Binobarin } R_{yu}^+ = \max(3,7; 0,5) = 3,7$$

Pastki chegarani topish uchun (2.4) ga muvofiq tenglamani o'zgartiramiz.

$$\begin{aligned} x^5 - 2x^4 + 100x^3 - 2 &= 0 \\ x^4(x - 2) &> 0; x > 2 \\ 100x^3 - 2 &> 0; x = 0,3 \\ R_1 &= \max(2; 0,3) = 2 \end{aligned}$$

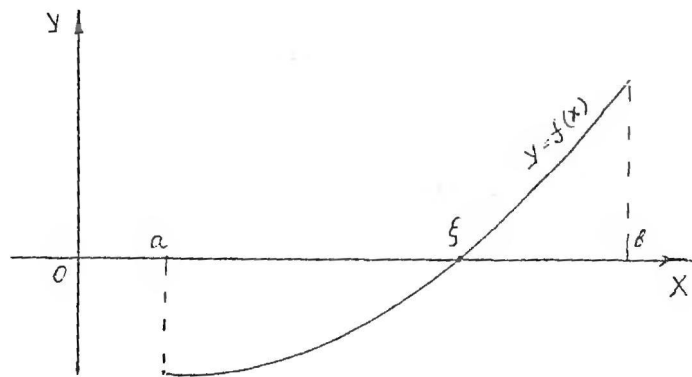
$$\text{va } r = \frac{1}{R_1} = \frac{1}{2} = 0,5$$

Bundan  $0,5 \leq X \leq 3,7$ , ya'ni bu usulda avvalgi usulga nisbatan aniqroq oraliq topiladi.

Ildizlar mavjud bo'lgan oraliq hisoblab topilgandan keyin ular aniqliklarini topishga kirishish mumkin. Buning uchun grafik, ketma-ket tanlash, yarim bo'lish, vatar, Nyuton (nisbatlar) va iteratsiya usullardan birini qo'llash mumkin.

### 2.1.3. KETMA-KET TANLASH USULI

Bu usulning mohiyati shundan iboratki, hisob boshi sifatida, masalan, ildizlar joylashgan oraliqning pastki chegarasi — a nuqta olinadi va ildizlarni qaysi aniqlik bilan topilishini  $\epsilon$  orqali belgilanadi. Shundan keyin  $f(x)=0$  tenglamaning  $x=a$  dagi miqdori aniqlanadi.  $Y=f(a)$  — manfiy yoki musbat son bo'lishi mumkin, ya'ni  $f(a)>0$  yoki  $f(a)<0$ . Argument  $x$  ga orttirma berib, funksiyaning miqdorini qaytadan hisoblaymiz. Agar  $f(x+\epsilon)$  ning qiymati oldingi olinganiga nisbatan ishorasini o'zgartirsa, ya'ni  $f(x)^* f(x+\epsilon)<0$  bo'lsa, u holda  $x$ qa yoki  $x=a+\epsilon$  larni birini tenglamaning ildizi deb hisoblash mumkin. Agarda  $f(x)$  va  $f(x+\epsilon)$  bir xil ishorali sonlar bo'lsalar,  $x$  ga orttirma berishni davom ettiramiz, (to funksiya miqdorining ishorasi o'zgarmaguncha), ya'ni  $f[x+(n-1)^* \epsilon]^* f[x+n^* \epsilon]<0$ . Tenglamaning aniq, ildizi shu ikkita  $x+(n-1)^* \epsilon$  va  $x+n^* \epsilon$  miqdorlar orasida joylashgan bo'ladi. Taxminan ikki sondan birini ildiz deb olinadi. Ko'rilayotgan oraliqda ildizlarning soni bittadan ortiq bo'lsa, ularni topish uchun birinchisini aniqlash singari ish bajarish lozim.



2.2-shakl.

Misol. Berilgan tenglamaning  $x^3 - 4x^2 + 2 = 0$  0,0001 aniqlik bilan tenglama ildizi topilsin.

$$X^3 - 4X^2 \geq 0$$

$$X^2(X - 4) \geq 0$$

$$X \geq 0 \quad X \geq 4$$

$R = \text{MAX}(0, 4) = 4$  yo'qori chegara, endi quyi chegarani aniqlaymiz.

$$\frac{1}{x^3} - \frac{4}{x^2} + 2 = 0$$

$$2x^3 - 4x + 1 = 0$$

$$x(2x^2 - 4) \geq 0$$

$$2x^2 - 4 \geq 0$$

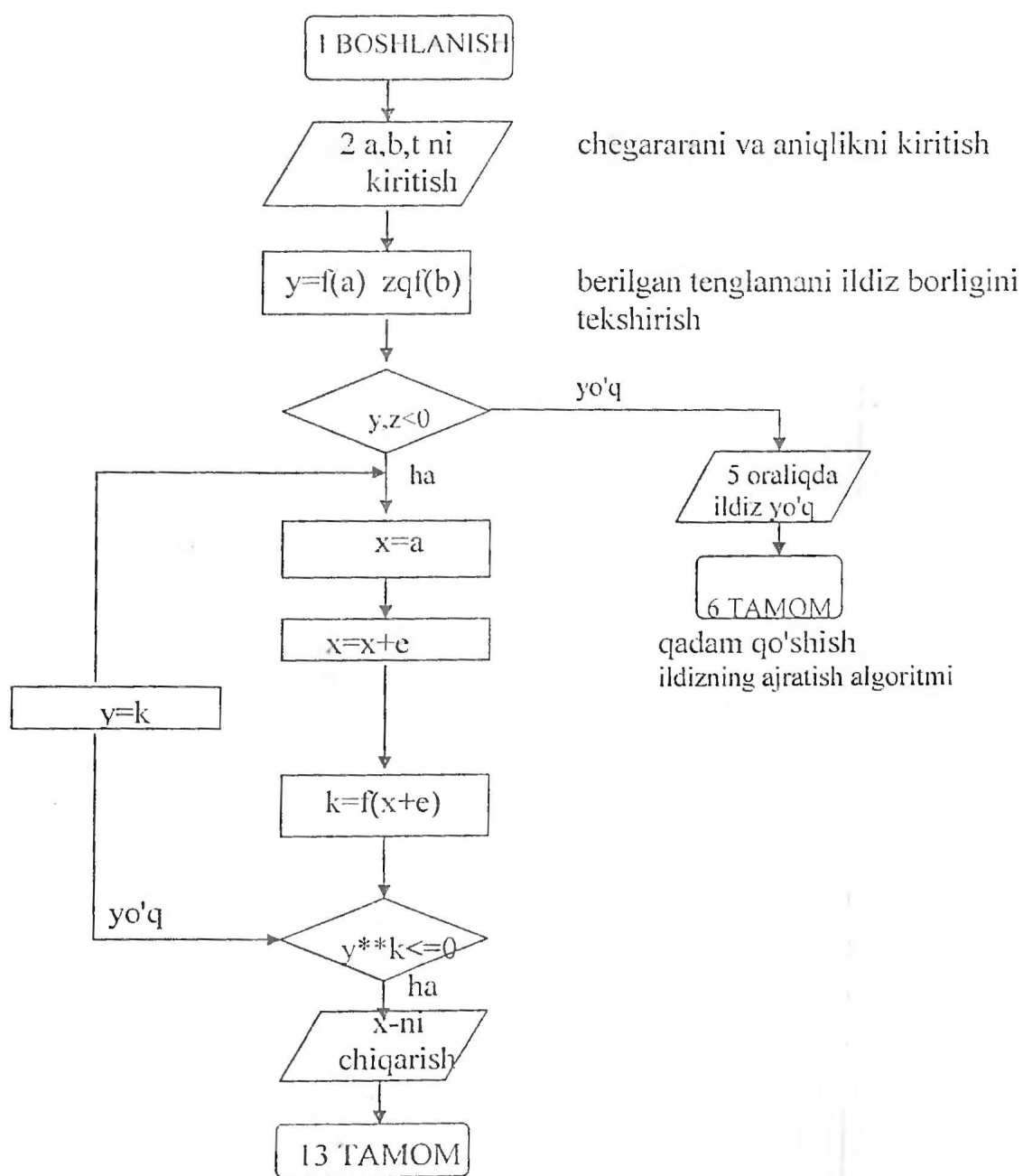
$$x^2 \geq \frac{1}{2} \quad x \geq \frac{1}{\sqrt{2}} \geq \frac{1}{1,4}$$

$$R_1 = \text{MAX}\left(0, \frac{1}{1,41}\right) = \frac{1}{1,41}$$

$$r = \frac{1}{R^1} = -\frac{1}{1/1,4} = 1,41,$$

bundan  $a = 1,41$ ;  $b = 4$ .





2.3 – shakl.

Blok-sxema chizilgandan keyin dasturlashga o'tamiz.  
Quyida ildiz ajratish dasturi keltirilgan.

```

1. Program ILDIZ AJRATISH
2. LABEL 1, 2, 3,4;
3. VAR A, B, E, X, K, Y, Z: REAL: o'zgaruvchilarni ta'riflash
4. BEGIN
5. READLN(A,B,E); oraliqlarni va aniqlikni kiritish
6. Y1 = .....;
7. Z: =.....;
8. IF Y*Z<0 THEN GOTO 3 } ildiz borligini tekshirish
9. X: =A;
10. 1: X: =X+E;
11. K: =.....;
12. IF Y*K<=0 THEN GOTO 2 } ildizni ajratish
13. Y:=K; GOTO1;
14. 2: WRITTELN ('X=', X);
15. GOTO4;
16. 3: WRITTELN ('oraliqda ildiz yo'q')
17. 4: END.

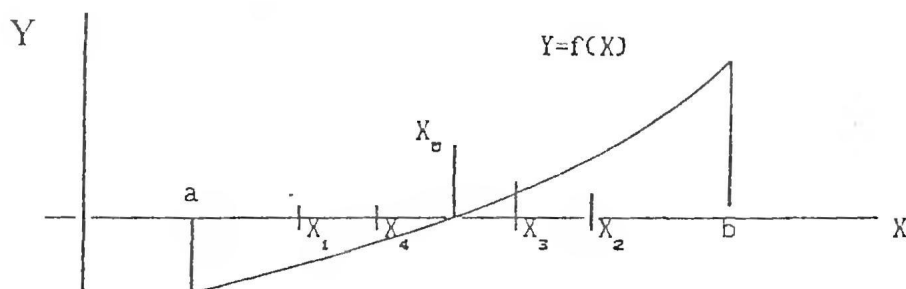
```

#### Dasturga izoh:

2,3-satrlarda o'zgaruvchilarni belgilaydi va ta'riflaydi;  
5-satrd kompyuter xotirasiga muayyan ildiz aniqligi va chegaralari qiymatini kiritadi;  
6 - satrda quyi chegarada "a" funksiyasini hisoblab chiqadi;  
7 - satrda yo'qori chegarada "b" funksiyasini hisoblab chiqadi;  
8 - satrda  $a < x < b$  chegaralarida ildiz mavjudligi shartlarini tekshiradi, agar ildiz bu chegarada bo'lmasa, 16 satrga o'tiladi va hisoblash to'xtatiladi. Agar ildiz ko'rsatilgan chegara ichida mavjud bo'lsa, hisoblashlar davom ettiriladi va 9 - satrga o'tiladi.  
10- satrda E aniqligida saralov qadamini belgilab beradi;  
11-satrd orttirmaning navbatdagi nuqtasida funksiyani hisoblaydi;  
12 - satrda "a" nuqtasida aniqlangan funksiyasi  $a+E$  nuqtadagi funksiya bilan solishtiradi va, agar solishtirish natijasi "0" yoki "0"dan kichik chiqsa, ildiz topilgan hisoblanadi; aks holda, 11-satrga o'tib keyin 1-belgiga yoki 10-satrga o'tish lozim va hokazo, toki ildiz topilguncha davom ettiriladi.

### 2.1.4. YARIM BO'LISH USULI

$f(x)=0$  tenglamasi ildizini topish uchun, agar  $[a, b]$  kesmalariga tegishli bo'lsa, bu kesmani ikkitaga bo'lamiz



2.4-shakl.

u holda ajratilgan bo'laklar —  $\left[a, \frac{a+b}{2}\right], \left[\frac{a+b}{2}, b\right]$  dan birini olish kerak, qaysini uchlarida funksiya har xil ishorali bo'lsa, ya'ni  $f(a) * f\left(\frac{a+b}{2}\right) < 0$  yoki  $f(b) * f\left(\frac{a+b}{2}\right) < 0$  shartlar bajarilsa. Yangi olingan bo'lakni o'rta nuqtasi uchun funksiyaning qiymati nolga tengligi tekshiriladi, aks holda toraygan yarim bo'laklarning birini yo'qorida aytilgan belgisi bo'yicha tanlab olamiz va uni bo'lishni davom ettiramiz. Bunday ish natijasida biror bosqichdagi hisoblangan funksiyaning qiymati nolga teng bo'ladi, ya'ni uning aniq ildizi aniqlanadi, yoki olingan kesmaning uzunligi hali juda kichik bo'lgani sababli quyidagi shartlar bajariladi:

$$f(a) * f(b) < 0 \quad (n = 1, 2, 3, \dots) \quad (2.7)$$

$$\text{va } b_n - a_n = \frac{1}{n} (b - a) \quad (2.8)$$

Bundan tashqari,  $a_1, a_2, \dots$  an kesmaning chap uchlarni absissasi monoton kamayuvchi, o'ng uchlari esa  $b_1, b_2, \dots, b_n$  monoton o'smaydigan cheklangan ketma-ketliklarni hosil qilishadi, shu sababli (2.7) tenglik kuchi tufayli umumiy limit mavjuddir.

$$\xi = \lim_{n \rightarrow \infty} a_n + \lim_{n \rightarrow \infty} b_n \quad (2.9)$$

(2.7) tengsizlikda  $n$ -limitiga o'tib  $f(X)$  funksiyaning uzluksizligidan  $[f(\xi)]^2 < 0$  ga ega bo'lamiz. Bundan  $f(\xi) = 0$ , ya'ni tenglama ildizidir hamda ayondirki,

$$0 < \xi - a_n \leq \frac{1}{2^n} (b - a) \quad (2.10)$$

Misol. Kesmada  $[1, 2]$  absolut xato  $E=10^{-4}$  bo'lgan

$$f(x) = \cos(x) - e^{\frac{-x}{2}} + x - 1 = 0$$

tenglamaning ildizini yarim bo'lish usulida topilsin.

Tenglama ildizini aniqlash algoritmining amallarini bajarish ketma-ketligini yozib chiqamiz:

1.  $a=1$ ;  $b=2$  va  $E=0.0001$  deb bilamiz

2.  $F(a) = \cos a - e^{\frac{-a}{2}} + a - 1$  hisoblab chiqamiz

3.  $x=(a+b)/2$  ni hisoblab chiqamiz va shu nuqtada  $f(x) = \cos x - e^{\frac{-x}{2}} + x - 1$  funksiyaning qiymatini belgilab olamiz.

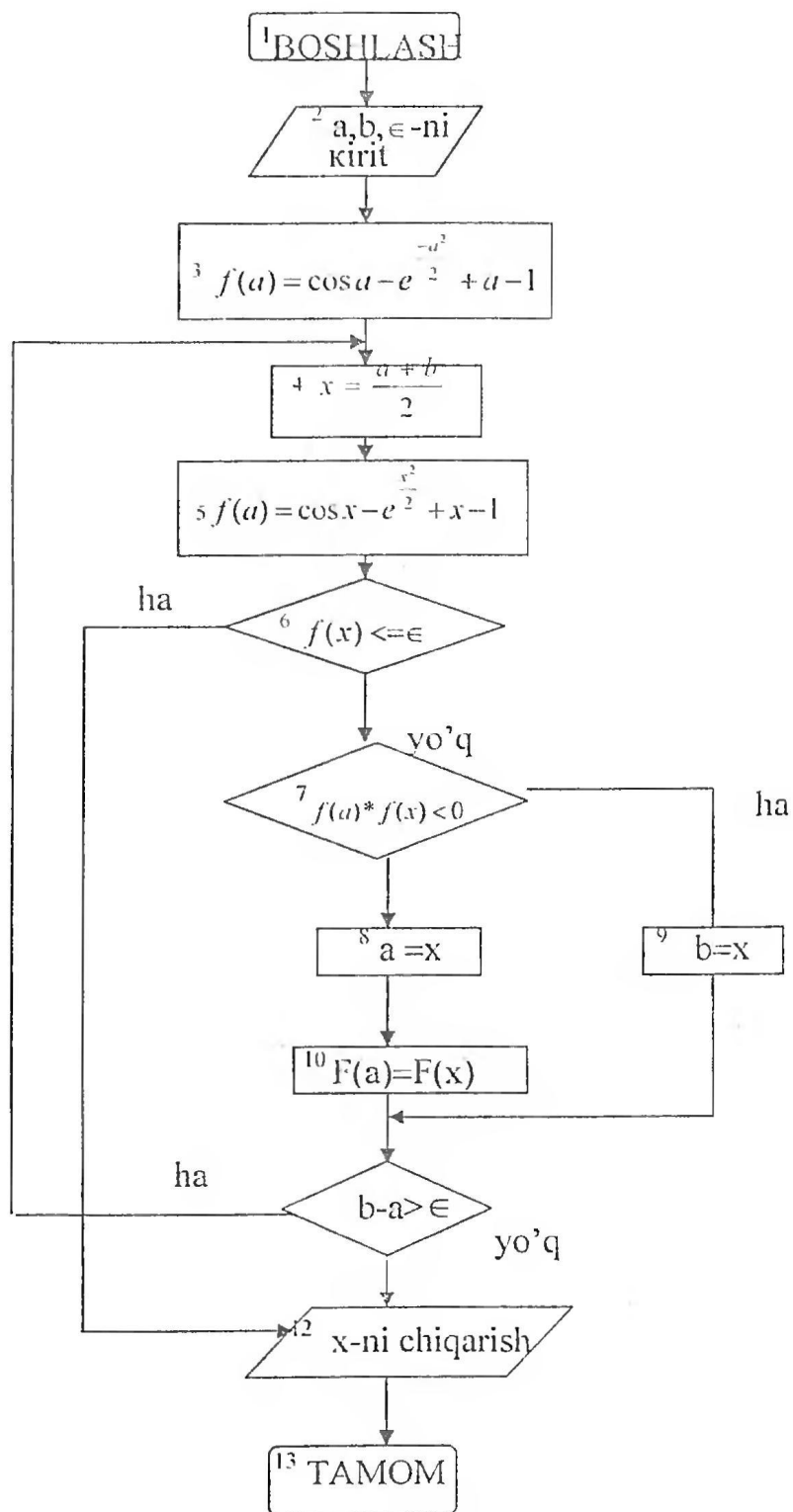
4.  $f(x)=0$  shartlarini tekshiramiz. Agar bu shartlar bajarilgan bo'lsa, unda  $X$  ildiz deb hisoblanadi va hisoblashni to'xtatamiz. Agar shartlar bajarilmagan bo'lsa, kesma tanlashga o'tamiz, bunda tanlangan kesma uchlarida  $f(x)$  funksiya har xil ishorali bo'lishi lozim, ya'ni.

5.  $f(a)*f(x)<0$  shartini tekshiramiz. Agar bu shart bajarilayotgan bo'lsa, unda  $b=x$  deb hisoblaymiz va 6 chi punktga o'tamiz.

Agar shart bajarilmayotgan bo'lsa  $a=x$ ,  $S(a)=f(x)$  hisoblaymiz va 6 chi punktga o'tamiz.

6.  $b-a > \epsilon$  shartini tekshiramiz. Agar bu shart bajarilayotgan bo'lsa, unda kesmalarni ikkitaga ajratish jarayoniga qaytamiz; aks holda  $X$  qiymatini natija deb qabul qilamiz va hisoblashni to'xtatamiz.

Yechim algoritmining blok-sxemasi 2.5-shaklda ko'rsatilgan.



2.5-shakl.

Mazkur algoritimga muvofiq tayyorlangan dastur va uning bajarilgan natijasi quyida ko'rsatilgan.

```
1. Program YARIM BOLISH;  
2. LABEL 1,2,3,4;  
3. VAR A,B,E,X,Y,2: REAL;  
4. BEGIN  
5. READLN (A,B,E);  
6. Y:= COS(A)-EXP(-(A*A/2))+A-1;  
7. 1: X:=(A+B)/2;  
8. Z:=COS(X)-EXP(-(X*X/2))+X-1;  
9. IF Z<=E THEN GOTO 2;  
10. IF Y*Z<0 THEN GOTO 3;  
11. A:=X;  
12. Y:=Z; GOTO 4;  
13. 3: B:=X;  
14. 4: IF (B-A)>E THEN GOTO 1;  
15. 2: WRITELN ('X=', X);  
16. END.
```

#### Dasturga izoh:

5 - satrda boshlang'ich ma'lumotlar kiritilgan;  
6 - satrda "a" nuqtadagi funksiyaning qiymati hisoblanadi;  
7 - satrda [a, b] kesma oralig'i aniqlanadi;  
8 - satrda bu oraliqdagi (x nuqtasida) funksiyaning qiymati hisoblab chiqiladi;  
9-satrda berilgan aniqlikda ildiz chiqarish sharti tekshiriladi. Shart bajarilganda, 15-satrda topilgan ildizning qiymati ekranga chiqariladi, aks holda 10-satrga o'tib, [a, x] kesma chegaralarida funksiya qiymatining har xil ishoraliqi tekshiriladi, ya'ni  $Y*Z<0$  sharti qo'yiladi. Agar oxirgi shart bajarilsa, aniqlanayotgan ildiz [a, b] kesma oraliqda mavjud bo'lgani sababli 13-satrga o'tilib, x ga b ning qiymati beriladi. Shunday qilib 14-ni satr bajarilishi oldidan biz yangi [a, b] kesmaga ega bo'lamiz. Bu kesma avvalgisidan ikki marta kichik va to'liqsiz ildizga ega.  
14-satrda berilgan aniqlikka erishilganligi tekshiriladi va zarurat tug'ilsa, 7-satrga qaytiladi va oraliqni navbatdagi ikkiga bo'lish bajariladi.

Agar 10-satrda  $Y*2<0$  shart bajarilmayotgan bo'lsa, bilinadiki, [a, x] kesma ildizga ega emas. Binobarin, ildiz [x, b] kesmadadir. 11-satrda a ga X ning yangi qiymati beriladi, 12-satrda tegishli ravishda Z ni Y o'zlashtiradi

va 14-satrga o'tiladi.

### 2.1.5. NYUTON USULI

Faraz qilaylik,  $x$   $f(x)=0$  tenglamaning ildizi bo'lib,  $[a,b]$  kesmada cheklangan; shu bilan birga,  $f(x)$ ,  $f'(x)$  uzluksiz va  $a < x < b$  oraliqda o'z ishoralarini o'zgartirmaydilar. Ildizning qandaydir  $n$  chi  $x_n$  "  $x(a \leq x \leq b)$  taxminiy qiymatini topib. Nyuton usuli bo'yicha quyidagicha aniqlash mumkin.

$$X = x_n + h_n \text{ deb qo'yamiz,} \quad (2.11)$$

bu yerda  $h_n$  -ni kichik miqdorli kattalik deb hisoblaymiz.

Bunda Teylor formulasidan foydalanib topamiz:

$$0 = f(X_n + h_n) = f(X_n) + h_n f'(X_n). \quad (2.12)$$

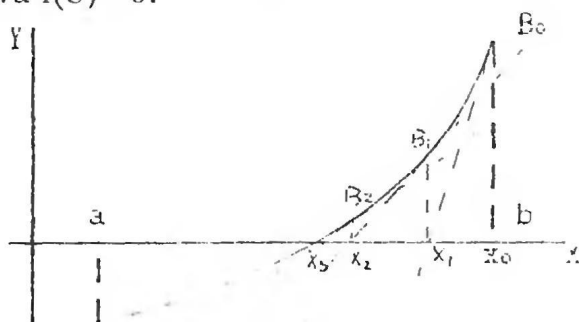
Demak,

$$h_n = -\frac{f(X_n)}{f'(X_n)} \quad (2.13)$$

Mazkur tuzatmani (2.11) formulaga kiritib, keyingi taqribiy ildizni topamiz (tartib bilan).

$$X_{n+1} = X_n - \frac{f(X_n)}{f'(X_n)} \quad (n = 0, 1, 2, \dots). \quad (2.14)$$

Nyuton usulining geometrik ma'nosi shundan iboratki, uzunligi kichik bo'lgan egri yoyni  $y = f(x)$  tenglamaning biror nuqtasida o'tkazilgan urinma bilan almashtiramiz, va o'z navbatida, aniqlik uchun  $f'(x) > 0$  ni qo'yamiz, bunda  $a < x < b$  va  $f(b) > 0$ .



2.6-shakl.

Masalan,  $X_0=b$  ni tanlab olamiz.  $B_0[X_0, f(X_0)]$  nuqtada  $Y=f(X)$  egri chiziqqa urinma o'tkazamiz.

$x$  ildizning birinchi  $X_1$  taxminiy miqdori sifatida urinma bilan  $OX$  o'qi kesishgan nuqtasining absissasini olamiz  $V_1$  nuqtadan yangidan urinma o'tkazamiz va uning  $OX$  bilan kesishgan nuqtaning absissasi  $X_2$  qiymatini beradi, qaysiki  $x$  ildizning ikkinchi taxminiy miqdori hisoblanadi, va hokazo.



```

PROGRAM NYUTONUSULI;
Label 1, 2;
VAR A, B, E, X, F1, F2, Y, D; REAL;
BEGIN
  READLN (A, B, E);
  F1:= EXP (-A)+A*A-2;
  F2:= EXP(-A)+2;
  IF F1*F2>0 THEN GOTO 1;

```

Ravshanki,  $B_n[X_n, f(X_n)]$  nuqtadagi urinmaning tenglamasi.

$Y-f(X_n)=f'(X_n)(X-X_n)$  kurinishida bo'ladi deb faraz qilib,  $Y=0$ ,  $X=X_n+1$ , quyidagilarga ega bo'ladi :

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)};$$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}; \quad x_2 = x_1 - \frac{f(x_1)}{f'(x_1)};$$

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)}$$

va hokazo.

Agar  $X_0=a$  deb va, demak,  $f(x_0)*f'(x_0)<0$ ,  $Y=f(x)$  egri chiziqqa  $A[a, f(a)]$  nuqtada urinma o'tkazsak,  $[a, b]$  kesmadan tashqarida yotuvchi  $X$  nuqtani hosil qilamiz, ya'ni boshlang'ich qiymatni berilayotganda. Nyuton usuli qulay emas ekan. Shuning uchun  $X_0$  sifatida zarur  $f(x_0)*f'(x_0)>0$  tengsizlikni qanoatlantiruvchi qiymatni tanlash zarur.

Taqribiy hisoblash jarayonini

$$|x_{n+1} - x_n| < \sqrt{\frac{2mE}{M}} \quad (2.15)$$

shart bajarilgandan so'ng to'xtatamiz. Bunda  $m[a, b]$  kesmada ning eng kichik qiymati.

$M[a, b]$  kesmada eng katta qiymati.

Bu shart uchun quyidagi tengsizlik bajariladi.

$$|\xi - x_n| \leq E,$$

bunda  $E$  — aniqlayotgan  $x$  ildizni absolut xatosining chekli qiymati.

Misol. Nyuton usuli bilan  $f(x)=e^{-x}+x^2-2$  tenglamaning ildizini  $[1;2]$  kesmada  $=10^{-5}$  aniqlikda toping.

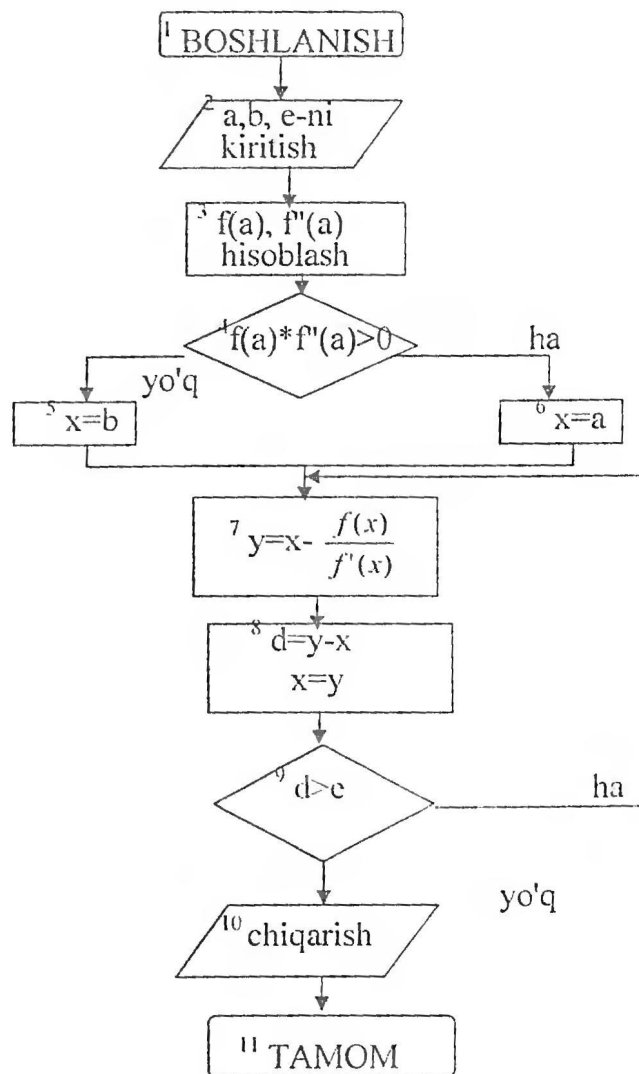
$$F'(x)=-e^{-x}+2x; \quad f'(x)=e^{-x}+2$$

(11.14) formulaga binoan

$$x_{n+1} = x_n - \frac{e^{-x_n} + x_n^2 - 2}{-e^{-x_n} + 2x_n}$$

belgilaymiz  $Y=x_n+1$  va

4.  $x=a$  ni belgilab, 6 punktga o'tish.
  5.  $X=b$  belgilab, 6 punktga o'tish.
  6.  $Y = x - \frac{f(x)}{f'(x)}$  qiymatini hisoblash.
  7.  $D = |Y - X|$ ;  $X = Y$  qiymatini hisoblash.
  8.  $|D| \leq E$  shart bajarilishi (tekshirish)? agar u bajarilsa - 9 punktga, aks holda 6 punktga o'tish kerak.
  9.  $X$  qiymatini bosmaga chiqazish.
  10. Tamom.
- Nyuton usuli masala yechish blok-sxemasi 2.7-shaklda ko'rsatilgan.



2.7-shakl.

Iteratsiya usuli bilan ildizlarni amaliy aniqlashda (2.16)dagi funksiyasini shunday tanlash kerakki, uning hosilasining absolut qiymati iloji boricha 1 dan kichik lozim. Shunda ildiz albatta aniqlanadi va q qanchlik kichik bo'lsa, shunchalik iteratsiya sonini kamaytirish mumkin.

Dastlabki tenglama ishlashni istalgan umumiy usuli mavjuddir Funksiya  $f(x)$  hosilasi  $0 < m \leq f'(x) \leq M$   $a < x < b$  ligida chegaralangan bo'lsin. Hosilaning eng kichik qiymati  $m$ , eng kattasi esa  $M$  ga tengdir.

Berilgan tenglamani  $x = x - \lambda f(x)$ , ( $x > 0$ ) ko'rinishga olib kelimiz.  $\varphi(x) = x - \lambda f(x)$  deb olamiz. Agarda  $\lambda = 1/M$  - parametрни tanlasak, unda

$$\begin{aligned} 0 \leq \varphi'(x) &= 1 - \lambda f'(x) \leq q < 1, \\ q &= 1 - m/M < 1 \end{aligned}$$

**Misol.**  $X - \sin x = 0,25$ ,  $E = 0,001$ ,  $1 < x < 1,3$ .

Tenglama ko'rinishini o'zgartiramiz, ya'ni  $x = 0,25 + \sin x$ , shunda  $\varphi(x) = 0,25 + \sin x$  bo'ladi.

$\varphi'(x) = \cos x$  topamiz, so'ng  $m$  va  $M$  ni aniqlaymiz.

$$M = \cos 1 = 0,5403;$$

$$m = \cos 1,3 = 0,2675;$$

$$q = 1 - (0,2675/0,5403) < 1.$$

Shuning uchun iteratsiya cheklangan jarayon hisoblanadi va boshlang'ich yaqinlashish uchun  $x_0 = 1$  deb qabul qilamiz.

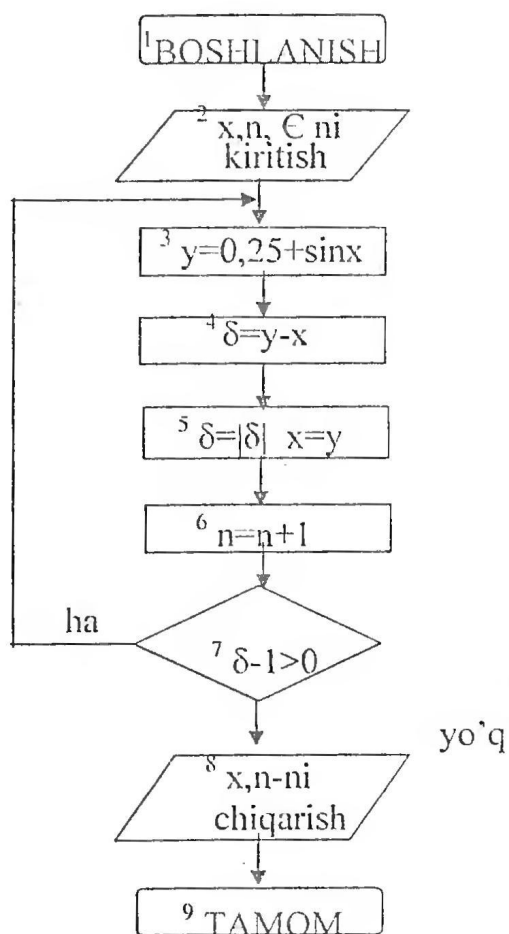
Hisoblash jarayonining algoritmi quyidagicha:

1. Faraz qilamiz  $x_0 = 1$ ,  $E = 0,0001$ ,  $n = 0$ .
2. Keyingi ildiz yaqinlashishini hisoblaymiz:

$$x_{n+1} = 0,25 + \sin x_n$$

3.  $\delta = x_{n+1} - x_n$  farqini hisoblaymiz va  $n$  kattaligini 1 ga oshiramiz.
4.  $\delta > E$  shartni tekshiramiz, agar "ha" bo'lsa, 2ga boramiz; agar yo'q bo'lsa, natijani bosmaga chiqaramiz.

Blok-sxema 2.10-shaklda ko'rsatilgan.



2.10-shakl.

```

PROGRAM ITERACIA USULI;
LABEL 1;
VAR D, N, E, X, Y: REAL;
BEGIN
  READLN (X,N,E);
  1: Y:=0.25+sin(x);
  D:=ABS (Y-X);
  X:=Y;
  N:=N+1;
  IF D>E THEN GOTO 1;
  WRITELN ('x=', x 'N=', N);
  END.
  
```

## TEKSHIRISH UCHUN SAVOLLAR

1. Tenglama yechishning ma'nosi nimada?
2. Algebraik va transsendent tenglamaning sonli usuli bilan zaruriyati yechish nima uchun vujudga keladi?
3. Tenglama ildizini topishda qanday ikki masalani yechishga to'g'ri keladi?
4. Yarim bo'lish va tanlash usulining afzalligi va kamchiligi.
5. Nyuton usuli. Yaqinlashish sharti.
6. Iteratsiya metodi. Yaqinlashish sharti.
7. Qanday operatorlar siklni tashkil etadi?
8. Yarim bo'lish usulida qaysi operatorlar tarmoqlanishni amalga oshiradi?  
Tarmoqlanishni tashkil etishda shartsiz o'tish operatorining ahamiyati.

## MASHQLAR

Keyingi tenglamalarning ildizini aniqlashda blok-sxema va dasturni tuzing.

1.  $3\sin\sqrt{x} + 0.35x - 3.8 = 0$ ;  $2 < x < 3$  iteratsiya usuli bilan
2.  $0.25x^3 + x - 1.2502 = 0$ ;  $0 < x < 2$  Nyuton usuli bilan
3.  $x + \sqrt{x} + \sqrt{x} - 2.5 = 0$ ;  $0.4 < x < 1$  Yarim bo'lish usuli bilan
4.  $x - \frac{1}{3 + \sin 3.6x} = 0$   $0 < x < 0.8$  iteratsiya usuli bilan
5.  $0.1x^2 - x \ln x = 0$   $1 < x < 2$  Nyuton usuli bilan
6.  $\lg x - \frac{1}{3} \lg^3 x + \frac{1}{3} \lg^5 x - \frac{1}{3} = 0$   $0 < x < 0.85$  bo'lish usuli bilan
7.  $\arccos x - \sqrt{1 - 0.3x^3} = 0$   $0 < x < 1$  iteratsiya usuli bilan
8.  $3x - 4 \ln x - 5 = 0$   $2 < x < 4$  Nyuton usuli bilan
9.  $\cos \frac{2}{x} - 2 \sin \frac{1}{x} + \frac{1}{x} = 0$   $1 < x < 2$  yarim bo'lish usuli bilan
10.  $\sqrt{1 - 0.4x^2} - \arcsin x = 0$ ;  $0 < x < 1$  iteratsiya usuli bilan
11.  $e^x - e^{-x} - 2 = 0$   $0 < x < 1$  Nyuton usuli bilan
12.  $\sin(\ln x) - \cos(\ln x) + 2 \ln x = 0$   $1 < x < 3$  yarim bo'lish usuli bilan
13.  $x - 2 + \sin \frac{1}{x} = 0$   $1.2 < x < 2$  iteratsiya usuli bilan
14.  $e^{x+} \ln x - 10x = 0$   $3 < x < 4$  Nyuton usuli bilan
15.  $\cos x - e^{-\frac{x^2}{2}} + x - 1 = 0$   $1 < x < 2$  yarim bo'lish usuli bilan
16.  $1 - x + \sin x - \ln(1+x) = 0$   $0 < x < 1.5$  tanlash usuli bilan
17.  $3x - 14 + e^x - e^{-x} = 0$   $1 < x < 3$  Nyuton usuli bilan

- |                                     |                                         |
|-------------------------------------|-----------------------------------------|
| 18. $\sqrt{1-x} - \lg x = 0$        | $0 < x < 1$ yarim bo'lish usuli bilan   |
| 19. $x - \cos(x^{0.52} + 2) = 0$    | $0.5 < x < 1$ tanlash usuli bilan       |
| 20. $3\ln^2 x + 6\ln x - 5 = 0$     | $1 < x < 3$ iteratsiya usuli bilan      |
| 21. $\sin x^2 + \cos x^2 - 10x = 0$ | $0 < x < 2$ Nyuton usuli bilan          |
| 22. $x^2 - \ln(1+x) - 3 = 0$        | $2 < x < 3$ iteratsiya usuli bilan      |
| 23. $2x \sin x - \cos x = 0$        | $0.4 < x < 1$ yarim bo'lish usuli bilan |
| 24. $e^x + \sqrt{1+e^{2x}} - 2 = 0$ | $-1 < x < 0$ tanlash usuli bilan        |

## 2.2. FUNKSIYALARNI INTERPOLYASIYALASH

Amalda egri chiziq va jadval ko'rinishda berilgan  $y=f(x)$  funksiyalari bilan tez-tez ishlashga to'g'ri keladi.

Masalan, sinov natijasida tajriba berilganlari jadvaliga ega buladilarki, bu jadvaldagi  $x_0, x_1, \dots, x_n$  natijalarning (ularni interpolatsiya tugunchalari deyiladi) argumentlari qiymatiga  $(a, b)$  oralig'ida funksiyalarning ma'lum bir qiymati mos keladi;

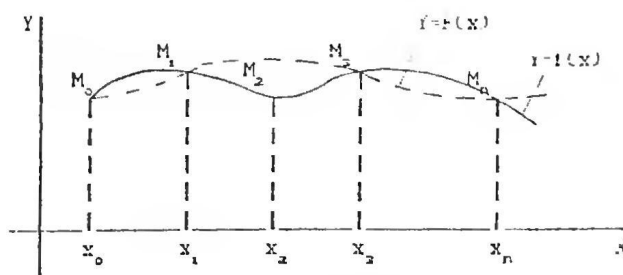
$$Y_0 = f(x_0); y_1 = f(x_1); \dots y_n = f(x_n) \quad (2.18)$$

Bu berilgan ma'lumotlar ishlagan jadvali berilgan funksiyaning taxminiy ifodasini topish, ya'ni  $(a, b)$  oralig'ida  $F(x)$  ning  $x_0, x_1, \dots, x_n$  nuqtalaridan farqli qiymatini aniqlash lozim.

Yoki boshqa bir holni olaylik funksiyaning taxminiy ifodasi ma'lum, lekin bu formula bo'yicha bevosita hisoblash jiddiy qiyinchiliklarga ega, bunday holda  $f(x)$ ni funksiyalarning malum turi bilan almashtiradilar, toki  $x_0, x_1, \dots, x_n$  nuqtalarida berilgan funksiya  $f(x)$  kiymati va biz almashtirayotgan  $F(x)$  funksiya qiymati bilan bir-biriga mos kelguncha, ya'ni:

$$F(x_0) = y_0; F(x_1) = y_1; \dots; F(x_n) = y_n \quad (2.19)$$

Buning geometrik mazmuni, berilgan nuqtalar tizimi orqali utuvchi ma'lum bir turdagi egri chiziqni aniqlash kerak  $(M_i(x_i, y_i), i=0, 1, 2, \dots)$  ligini bildiradi.



2.11-shakl.

Masala bunday umumiy ko'rinishda cheksiz ko'p yechimlarga ega bo'lishi yoki umuman yechimga ega bo'lmasligi mumkin. Lekin, bu masala qachonki ixtiyoriy  $F(x)$  funksiyasi o'rniga (2.19.) tartibiga mos keladigan,  $n$  darajasidan yo'qori bo'lmagan  $Ln(x)$  polinomi qidirilsa, ya'ni

$$Ln(x_0)=y_0; Ln(x_1)=y_1 \dots Ln(x_n)=y_n \quad (2.20)$$

Mazkur interpolyatsiya  $y=F(x)$  formulasi odatda  $X$  argumentining interpolyatsiyalash tugunchalaridan farqli qiymati uchun berilgan  $f(x)$  funksiyasi qiymatlarini yaqinlashishini hisoblab chiqish maqsadlarida ishlatiladi. Bunday amal  $f(x)$  funksiyani interpolyatsiyalash deb ataladi. Interpolyatsiyalash so'zi ayrim paytda tor ma'noda ishlatilganda, qachonki  $x$   $[x_0, x_n]$ , ya'ni  $x$  qiymatini  $x_0$  va  $x_n$  oralig'ida aniqlanadigan bo'lsa va ekspolyatsiyalash ma'nosida ishlatilsa, qachonki  $x$  ( $x_0, x_n$ ), ya'ni  $x$  ning qiymati ko'rsatilgan interval tashqarisida joylashgan bo'ladi. Bundan keyin interpolyatsiyalash atamasi bilan yo'qoridagi amallarning ikkovini ham bir xil ma'noda ishlatamiz.

Interpolyatsiyalash masalalarini yechishda Nyuton, Gauss, Stirling, Bessel interpolyatsion formulalaridan foydalanamiz, lekin ularni interpolyatsion tugunlari teng oraliqda bo'lgan hollari uchungina ishlatish mumkin.

Ixtiyoriy berilgan tugunlar uchun Lagranjni interpolyatsiyalash umumiy formulasidan foydalanish kerak.

$$Ln(z) = \sum_{j=0}^n \frac{(z-x_0)(z-x_1)\dots(z-x_{j-1})(z-x_{j+1})\dots(z-x_n)}{(x_j-x_0)(x_j-x_1)\dots(x_j-x_{j-1})(x_j-x_{j+1})\dots(x_j-x_n)}, \quad (2.21)$$

bu yerda:  $Z$  — interpolyatsiya nuqtalardan farq qiluvchi funksiya aniqlanishi lozim bo'lgan nuqta.

Lagranj formulasiga (2.21) yanada ixchamroq, ko'rinish berish mumkin.

$$Ln(Z) = \sum_{j=0}^n y_j \prod_{i=0, i \neq j}^n \left( \frac{z-x_i}{x_j-x_i} \right) \quad (2.22)$$

Masalan,  $n=3$  uchun formula yoyilmasi bunday kurinishga ega bo'ladi:

$$Ln(z) = y_0 \frac{(z-x_1)(z-x_2)(z-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} + y_1 \frac{(z-x_0)(z-x_2)(z-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} + \\ + y_2 \frac{(z-x_0)(z-x_1)(z-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} + y_3 \frac{(z-x_0)(z-x_1)(z-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)}.$$

Hisoblash algoritmini tuzamiz. Buning uchun interpolyatsiya formulasini shunday tasavvur qilamiz:



$$\ln(z) = \sum_{i=0}^n y_i \frac{a_i}{b_i} \quad (2.23)$$

bunda  $a_i = \prod_{t=0}^i (z - x_t)$ ;  $b_j = \prod_{t=0}^j (x_j - x_t)$

yoki  $a_{i+1} = a_i (z - x_{i+1})$ ;  $a_0 = 1$

$b_{j+1} = b_j (x_j - x_{j+1})$ ;  $b_0 = 1$   $j = 0, 1, 2, \dots$ ;  $i = 0, 1, 2, \dots$ ;  $j \neq i$

Interpolyatsiya formulasi algoritm ko'rinishida shunday yozilishi mumkin:

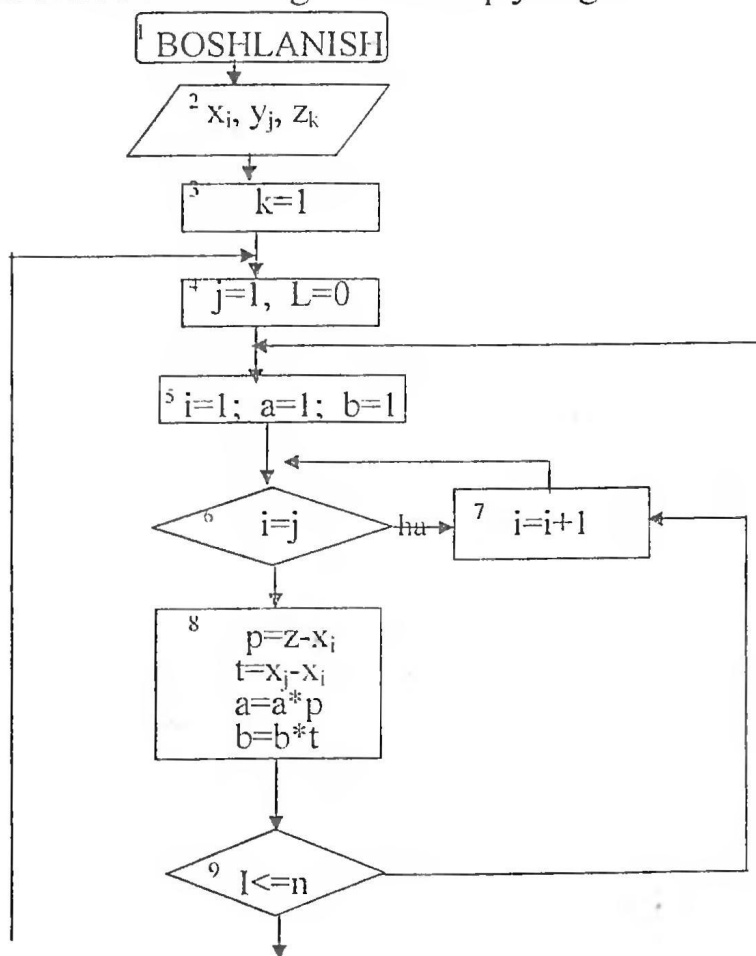
$$L_i = L_{i-1} + y_{i-1} \frac{a_{i-1}}{b_{i-1}}; \quad L_0 = 0$$

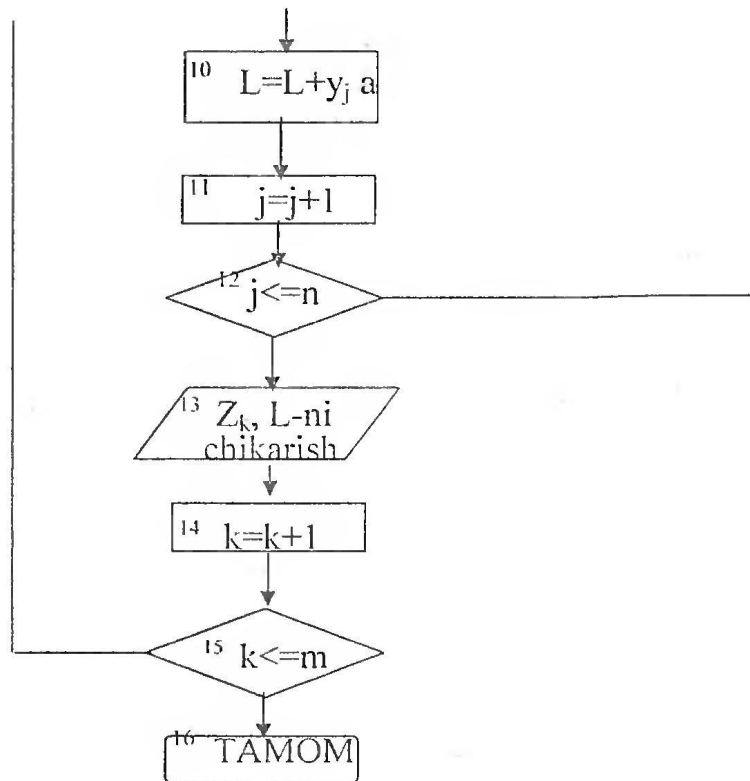
**Misol.** Funksiya jadvalda berilgan.

|   |       |       |       |       |        |
|---|-------|-------|-------|-------|--------|
| X | 3,5   | 3,55  | 3,6   | 3,65  | 3,7    |
| Y | 3,115 | 4,813 | 6,598 | 8,475 | 10,447 |

Argument qiymati  $Z_1=3,52$ ;  $Z_2=3,59$  bo'lgan hisoblash funksiyasining dasturini tayyorlang.

Algoritm blok-sxemasining ko'rinishi quyidagicha:





2.12 – shakl.

```

PROGRAM INTERPOL;
  LABEL 1;
  CONST N= ; M= ;
  VAR X, Y: ARRAY [1..N] OF REAL;
  Z: ARRAY [1..M] OF REAL;
  a, b, p, t, L: REAL;
  I, j, k: INTEGER;
BEGIN
  WRITELN ('Berilgan hamma nuqta uchun x, y koordinatalarini
    kiriting
      vergul orqali')
  FOR I:=1 TO N DO
    READLN (X[I], Y[I]);
  WRITELN ('Z koordinatasiga interpolatsiyalash nuqtasini
    kiriting');
  FOR I:=1 TO M DO
    READLN (Z[I]);
  FOR k:=1 TO M DO
    BEGIN

```

```

        L:=0;
FOR j:=1 TO N DO
    BEGIN
        A:=1.0;
        B:=1.0;
FOR I:=1 TO N DO
    BEGIN
        IF i:=j THEN GOTO 1;
        P:=z[k]-x[I];
        T:=x[j]-x[I];
        A:=A*P;
        B:=b*t;
1: END;
L:=L+Y[j]*a|b;
END;
WRITELN ('Z=', Z[K], 'L=', L);
END;
END.
M - interpolyatsiyalash berilgan nuqtaning soni.

```

## MASHQLAR

1. Funksiya jadvalda berilgan

|              |       |       |       |      |      |      |       |
|--------------|-------|-------|-------|------|------|------|-------|
| $X_i$        | 0.1   | 0.25  | 0.31  | 0.45 | 0.49 | 0.76 | 0.92  |
| $Y_i=f(X_i)$ | -2.31 | -1.34 | -0.56 | 1.28 | 1.00 | 0.46 | -0.21 |

$X_j = X_{j-1} + H$ ;  $X_0 = 0.1$ ;  $H = 0.01$

$F(X)$  qiymatini (0.1-0.92) kesmada  $X_i$  nuqtalarini hisoblab, algoritmi tuzing.

2.  $N(X)$  funksiya uchun jadval holda berilgan  $X=11.7$ ;  $X=12.3$  nuqtalardagi qiymatlarni topish algoritmlarini tuzing.

|       |      |      |      |      |      |      |      |      |       |       |       |
|-------|------|------|------|------|------|------|------|------|-------|-------|-------|
| $X_i$ | 2    | 3    | 4    | 5    | 6    | 7    | 8    | 9    | 10    | 11    | 12    |
| $Y_i$ | 3.32 | 3.74 | 4.24 | 4.87 | 5.68 | 6.72 | 8.07 | 9.85 | 12.34 | 15.65 | 20.04 |

3. Berilgan jadvaldan  $X=0.21$ ;  $0.19$ ;  $0.202$  uchun funksiyaning interpolyatsion algoritmi tuzilsin.

|   |       |      |       |       |
|---|-------|------|-------|-------|
| X | 0.196 | 0.2  | 0.204 | 0.208 |
| Y | 4.75  | 6.31 | 7.96  | 8.88  |

4. Funksiya jadvalda berilgan  $Y=Y(6.42)$ ;  $Y=Y(7.23)$  da hisoblash algoritmi tuzilsin.

|   |      |      |      |      |      |      |
|---|------|------|------|------|------|------|
| X | 1.6  | 2.4  | 3.6  | 4.15 | 6.2  | 7.15 |
| Y | 3.22 | 3.81 | 4.63 | 4.98 | 6.27 | 6.91 |

5. Funksiya jadvalda berilgan

|   |     |      |      |      |     |      |
|---|-----|------|------|------|-----|------|
| X | 9.6 | 10.2 | 10.8 | 11.4 | 12  | 12.6 |
| Y | 97  | 107  | 117  | 127  | 138 | 150  |

$Y=Y(9.42)$ ;  $Y=Y(9.7)$  da hisoblash algoritmi tuzilsin.

6.  $Y=F(X)$  funksiyani  $X=1.7$ ;  $X=2.6$  nuqtalarda hisoblash algoritmi quyidagilar berilgan holda tuzilsin.

|   |      |      |       |      |      |      |
|---|------|------|-------|------|------|------|
| X | 1.1  | 1.4  | 1.6   | 2.1  | 2.2  | 2.5  |
| Y | 42.2 | 20.4 | 13.72 | 6.05 | 5.26 | 3.58 |

7. Agar funksiya jadval ko'rinishida berilgan bo'lsa,  $X=15$ ;  $X=19$  nuqtalardagi funksiyaning qiymatini hisoblash algoritmi ishlab chiqilsin.

|   |    |       |       |       |      |     |     |       |
|---|----|-------|-------|-------|------|-----|-----|-------|
| X | 11 | 12.5  | 14    | 15.5  | 17.5 | 18  | 20  | 21    |
| Y | 69 | 101.6 | 142.2 | 193.9 | 278  | 303 | 416 | 481.5 |

8. Agar funksiya jadvalda berilgan bo'lsa, funksiyaning qiymatini berilgan  $X=15.2$ ,  $X=12$  argument qiymatlarida hisoblash algoritmi ishlab chiqilsin.

|   |       |       |        |       |     |    |      |
|---|-------|-------|--------|-------|-----|----|------|
| X | 12.5  | 14    | 15.5   | 17.5  | 18  | 20 | 21   |
| Y | -54.7 | -53.8 | -46.22 | -28.5 | -21 | 16 | 41.5 |

9. Agar funksiya jadvalda berilgan bo'lsa, funksiya qiymati  $X_1=5.05$   $X_2=5.07$  nuqtalarda hisoblash algoritmi ishlab chiqilsin.

|   |        |       |        |        |        |        |        |
|---|--------|-------|--------|--------|--------|--------|--------|
| X | 5.0    | 5.1   | 5.2    | 5.3    | 5.4    | 5.5    | 5.6    |
| Y | 0.2834 | 0.378 | 0.4685 | 0.5544 | 0.6347 | 0.7087 | 0.7756 |

10. Funksiya jadvalda berilgan.

|   |       |       |       |       |        |
|---|-------|-------|-------|-------|--------|
| X | 3.5   | 3.55  | 3.6   | 3.65  | 3.7    |
| Y | 3.115 | 4.813 | 6.598 | 8.475 | 10.447 |

Funksiyaning  $x_1=3.57$ ;  $x_2=3.59$  argument qiymatlarida hisoblash algoritmi ishlab chiqilsin.

### 2.3. CHIZIQLI TENGLAMALAR SISTEMASINING YECHIMI

Chiziqli tenglamalar sistemasi yechimlarida aniq hisoblangan Kramer formulasi va Gauss usuli hamda iteratsion, ya'ni cheksiz jarayonlar yo'li bilan berilgan aniqlikda sistema ildizlarini topish imkonini beradigan iteratsiya, Zeydel, relaksatsiya va boshqa usullari ishlatiladi.

$n$  ta noma'lum algebraik tenglamalar sistemasini Kramer usuli bilan hisoblashni ko'ramiz.

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + \dots + a_{1n}x_n = b_1;$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 + \dots + a_{2n}x_n = b_2;$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 + \dots + a_{3n}x_n = b_3;$$

$$a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + a_{n4}x_4 + \dots + a_{nn}x_n = b_n.$$

Bu usulda, noma'lum  $x_i = \Delta_i / \Delta$  bo'yicha aniqlanadi.  $\Delta$ -tizimning noma'lumlardan oldidagi koeffitsiyentlardan tashkil qilingan matritsaning determinanti;  $\Delta_i$  esa  $\Delta$  ning  $i$  chi ustunini ozod hadlardan almashtirilgani bilan farq qiladi.

Kompyuterda Kramer usuli kam qo'llaniladi; katta  $n$  lar tufayli aniqlovchilar elementlarining soni, kompyuter xotirasida bu elementlarning qiymatini saqlash uchun ajratiladigan joy va ularni hisoblab chiqishga sarflanadigan vaqt ortadi. Shuning uchun kompyuter hisoblarida noma'lumlarni chiqarishga asoslangan aniq usullar ko'proq ishlatiladi.

$n$  tizimdagi chiziqli algebraik tenglamalar yechimini Gauss usuli asosida topish algoritmi quyidagi asosiy bosqichlardan iborat:

1. Tizimning  $a_{kj}$  koeffitsiyentlari, hamda  $b_k$  ozod hadlari kompyuter xotirasida tug'ri burchakli matritsa shaklida joylashtiriladi, ya'ni:

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & a_{1,n+1} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & a_{2,n+1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} & a_{n,n+1} \end{pmatrix}, \quad (2.25)$$

bunda  $a_{1,n+1} = b_1$ ;  $a_{2,n+1} = b_2$ ;  $a_{n,n+1} = b_n$ .

2. Matritsaning birinchi satridagi elementlari  $a_{1j}$  ga bo'ladi, keyin  $a_{k1}$  ( $k=2,3,\dots,n$ ) ko'paytiriladi va  $k$ -satrlaridan ayiriladi (navbati bilan 2,3,  $p$ -satrdan). Matritsaning birinchi ustuni elementlari  $a_2$  dan boshlab to  $a_{n1}$  gacha nolga aylanadi.

$$A^{[1]} = \begin{bmatrix} 1 & a'_{12} & a'_{13} & \dots & a'_{1n} & a'_{1,n+1} \\ 0 & a'_{22} & a'_{23} & \dots & a'_{2n} & a'_{2,n+1} \\ 0 & a'_{n2} & a'_{n3} & \dots & a'_{nn} & a'_{n,n+1} \end{bmatrix} \quad (2.26)$$

Bunda uzgartirilgan matritsani  $A^1$  orqali belgilash qabul qilingan.

3.  $A^1$  matritsasi ikkinchi satri  $a_2$  ga bo'ladi keyin ga ko'paytiriladi va barcha  $k=3,4,\dots,n$  satrlardan ayriladi.

4. 3 chi punktda berilgan matritsaning o'zgartirish  $(n-1)$  chi satri bilan ham bajarilguncha davom ettiriladi. Ayni paytda matritsa quyidagi shaklga keltiriladi, ya'ni

$$A^{n-1} = \begin{bmatrix} 1 & a^{n-1}_{12} & a^{n-1}_{13} & \dots & a^{n-1}_{1n} & a^{n-1}_{1,n+1} \\ 0 & a^{n-1}_{22} & a^{n-1}_{23} & \dots & a^{n-1}_{2n} & a^{n-1}_{2,n+1} \\ 0 & 0 & 0 & \dots & 1 & a^{n-1}_{n-1,n+1} \\ 0 & 0 & 0 & \dots & a^{n-1}_{n,n} & a^{n-1}_{n,n+1} \end{bmatrix} \quad (2.27)$$

5. Olingan matritsaning oxirgi satri elementlari

$$x_n = \frac{a^{n-1}_{n,n+1}}{a^{n-1}_{n,n}} \quad (2.28)$$

qiymatini hisoblab chiqish imkonini beradi.

6.  $X_n$  ildizning qiymatini  $A^{n-1}$  uchburchakli matritsaning  $n-1$  satriga qo'yib  $x_{n-1}$  topiladi, so'ngra quyidagi formula orqali birin-ketin  $x_{n-1}, \dots, x_1$  ildizlar aniqlanadi:

$$x_k = a^{n-1}_{k(n-1)} - \sum_{j=k+1}^{n-1} a^{n-1}_{kj} * x_j; \quad k=m-1 \text{ formulalar joyida } x_{n-2}, \dots, x_n \text{ ildizlari}$$

hisoblab chiqiladi.

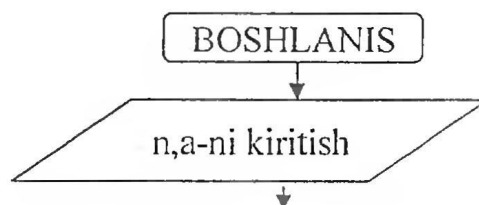
**Misol.** Chiziqli tenglamalar tizimi yechimining blok-sxemasi va dasturi tuzilsin.

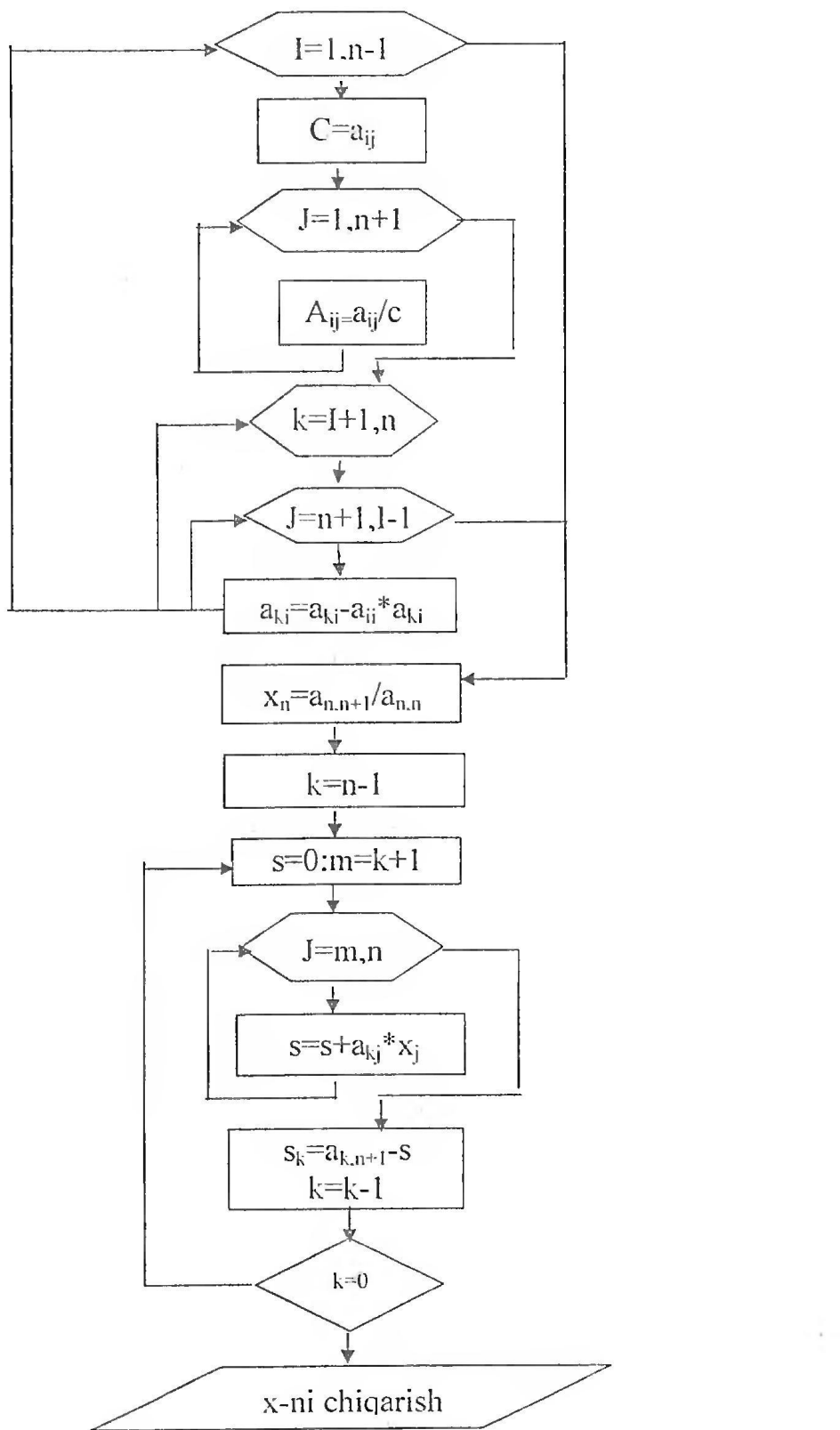
$$-3 * x_1 - x_2 = 5$$

$$-2 * x_1 - x_2 + x_3 = 0$$

$$-2 * x_1 - x_2 - 4 * x_3 = 15$$

Bu tizimda  $n$ -satrlar soni va noma'lumlar miqdori,  $m=n+1$  ustunlar soni blok-sxemasi 2.13-shaklda kursatilgan.





2.13.-shakl.



```

PROGRAM GAUS USULI;
LABEL 1;
CONST N= ;
VAR A: ARRAY [1...N, 1...N=1] OF REAL;
X: ARRAY [1...N] OF REAL;
C, S: REAL;
i, s, k, m: INTEGER;
BEGIN
FOR i:=1 TO N DO
FOR j:=1 TO N=1 DO
READLN (a[i,j]);
FOR i:=1 TO N-1 DO
BEGIN
C:=A[i,j];
FOR j:=1 TO N=1 DO
A[i,j]=A[i,j]/C;
FOR k:=i+1 TO N DO
BEGIN
FOR j:=N+1 DOWNT0 1 DO
A[k,j]:=A[k,j]-A[i,j]*A[l,j]; END; END;
X[N]:=A[N,N+1]/A[N,N]; k:=N-1;
1: S:=0; M:=k+1;
FOR j:=M TO N DO
S:=S+A[k,j]*x[j];
X[k]:=A[k,N+1]-S; k:=k-1;
Ik k>0 THEN GOTO 1;
FOR k:=1 TO N DO
WRITTELN ('x', k, '=', x[k]); END.

```

### Misollar

Chiziqli tenglamalar tizimi

$$\begin{array}{ll}
 x_1 + 2x_2 - 3x_3 = 5 & x_1 = 1.5357 \\
 1. \quad 2x_1 - x_2 = 3 & x_2 = 0.0714 \\
 3x_1 + 4x_2 - x_3 = 6 & x_3 = -1.1071 \\
 3x_1 + x_2 - 4x_3 = 10 & x_1 = 0.5 \\
 2. \quad 3x_2 - x_3 = 5 & x_2 = 1.0455 \\
 2x_1 + 3x_2 - x_3 = 6 & x_3 = -1.8636
 \end{array}$$

$$\begin{aligned}
& 2x_1 - 3x_2 - 2x_3 = -7 \quad x_1 = 3.1429 \\
3. \quad & 5x_1 - 2x_3 = 8 \quad x_2 = 1.8571 \\
& x_1 + 2x_2 - x_3 = 3 \quad x_3 = 3.8571 \\
& -x_1 + 2x_2 + x_3 = 6 \quad x_1 = 0.75 \\
4. \quad & 3x_1 + x_2 + 2x_3 = 10 \quad x_2 = 2.3051 \\
& x_1 - 4x_2 + x_3 = -4 \quad x_3 = 0.4068 \\
& 2x_1 + 3x_2 + 4x_3 = 12 \quad x_1 = 1.7288 \\
5. \quad & -3x_1 - 2x_3 = -6 \quad x_2 = 2.3051 \\
& 4x_1 + x_2 - 3x_3 = 8 \quad x_3 = 0.4068 \\
& x_1 + 2x_2 - 3x_3 = 5 \quad x_1 = 1.5357 \\
6. \quad & 2x_1 - x_2 = 3 \quad x_2 = 0.0714 \\
& 3x_1 + 4x_2 - x_3 = 6 \quad x_3 = -1.1071 \\
& x_1 - 2x_2 - 3x_3 = -9 \quad x_1 = -7 \\
7. \quad & 2x_1 - x_2 + 2x_3 = 5 \quad x_2 = -7.5714 \\
& 4x_2 + 6x_3 = 4 \quad x_3 = 5.7143 \\
& 3x_1 + 3x_3 = 7 \quad x_1 = 5.6667 \\
8. \quad & 2x_1 - x_2 + 2x_3 = 5 \quad x_2 = 3.6667 \\
& 4x_1 + 2x_2 + x_3 = 12 \quad x_3 = -3.3333 \\
& 2x_1 + x_2 - 5x_3 = 8 \quad x_1 = 1.2 \\
9. \quad & 4x_1 + 2x_2 - x_3 = 10 \quad x_2 = 2.2667 \\
& 3x_1 + 4x_2 + x_3 = 12 \quad x_3 = -0.6667 \\
& 3x_1 + x_2 + 2x_3 = 12 \quad x_1 = 1.5126 \\
10. \quad & 2x_1 - 3x_2 + x_3 = 9 \quad x_2 = -0.641 \\
& 4x_1 - x_3 = 2 \quad x_3 = 4.0513 \\
& 2x_1 + 3x_2 + 4x_3 = 12 \quad x_1 = 0.9091 \\
11. \quad & 3x_1 + 4x_2 + x_3 = 10 \quad x_2 = 1.4545 \\
& 9x_1 + x_2 + 2x_3 = 8 \quad x_3 = 1.4545 \\
& 2x_1 - 2x_2 + 3x_3 = 7 \quad x_1 = 1.4444 \\
12. \quad & 5x_1 + x_3 = 10 \quad x_2 = 2.1111 \\
& 3x_1 + x_2 + 2x_3 = 12 \quad x_3 = 2.7778
\end{aligned}$$

$$\begin{aligned}
& 3x_1 + 4x_2 + 5x_3 = 15 \quad x_1 = 8.05 \\
13. \quad & 2x_1 - 3x_2 + x_3 = 8 \quad x_2 = 1.65 \\
& x_1 + 5x_2 + 2x_3 = 10 \quad x_3 = -3.15 \\
& x_1 + 2x_2 + 3x_3 + 4x_4 = 11 \quad x_1 = 2 \\
14. \quad & 2x_1 + 3x_2 + 4x_3 + x_4 = 12 \quad x_2 = 1 \\
& 3x_1 + 4x_2 + x_3 + 2x_4 = 13 \quad x_3 = 1 \\
& 4x_1 + x_2 + 2x_3 + 3x_4 = 14 \quad x_4 = 1
\end{aligned}$$

## 2.4. FUNKSIYALARNI SONLI INTEGRALLASH

Agar  $f(x)$  funksiyasi  $[a, b]$  kesmada uzluksiz hamda uning dastlabki holdagi  $F(x)$  ma'lum bo'lsa, bu funksiyaning  $a$  dan to  $b$  gacha oralig'dagi aniq integrali quyidagicha

$$\int_a^b f(x) dx = F(b) - F(a) \quad (2.29)$$

Nyuton-Leybnits formulasi orqali hisoblab chiqilishi mumkin, bunda  $Y = \int_a^b f(x) dx$  absissa va ordinata o'qlari bilan cheklangan yuzadir.

Lekin, ko'pincha  $f(x)$  ning dastlabki holdagi funksiyasi eng oddiy vositalar yordamida topilmaydi yoki o'ta murakkab hisoblanadi. Buning oqibatida (2.29) formula asosida aniq integralni hisoblab chiqish qiyin bo'lishi mumkin yoki amaliy hisoblash imkonini yo'q.

Bundan tashqari, amalda  $f(x)$  integral ostidagi funksiya ko'pincha jadval shaklida beriladi va uning dastlabki tushunchasi o'z mazmunini yo'qotadi. Shuning uchun aniq integrallarni hisoblab chiqishning qulay usullari sonli hisoblash usullari bo'lib, ular aniq integral qiymatini integral ostidagi funksiyaning bir qancha qiymatlari asosida hisoblab chiqishga asoslangandir.

Aniq integralni taqribiy hisoblash uchun chekli yig'indilarga almashtirishda foydalaniladigan algoritmning keng tarqalgan turlariga quyidagi usullar kiradi: tug'riburchaklar usuli, trapetsiya va Simpson usullari.

Bu usullar bilan aniq integralni taqribiy hisoblash mumkin. Integral ostidagi egri chiziqli funksiya o'rniga, unga mos oddiy formula bilan hisoblanuvchi egri yoki sinq chiziqli qilib olish mumkin.

### 2.4.1. TO'G'RI BURCHAKLILAR USULI

Tug'ri burchaklilar usuli asosida integral ostidagi funksiyaning egri chiziqlarining bo'laklari abssissa o'qiga parallel siniq chiziqlar bilan almashtiriladi.

Integrallash  $[a,b]$  kesma uzunligi  $h$  ni  $n$  teng qismlariga ajratib chiqamiz.

Ajratib olingan  $x_0=a, x_1=a+h, \dots, x_n=b$  nuqtalarida  $y=f(x)$ , ya'ni  $y_i=f(x_i)$ ,  $x_i=a+ih$   $i=0,1,\dots,n$  egri chizig'i bilan kesishguncha  $y_0, y_1, \dots, y_n$  ordinatalarini o'tkazamiz, ordinata uchlarini to'g'ri burchakli bo'laklar bilan birlashtiramiz, bunda  $aABb$  egri chiziqli to'g'ri burchak yuzasini  $aACDN, \dots, Bb$  egri chizig'i bilan chegaralangan shakl yuzasiga teng deyish mumkin.  $S$  deb belgilanadigan bu shakl yuzasi

$$S = h(y_0 + y_1 + y_2 + \dots + y_n)$$

to'g'riburchaklar yuzasining yig'indisiga teng bo'ladi.

Shunday qilib, integralning taqribiy qiymati to'g'ri burchaklilar formulasiga binoan quyidagi ko'rinishda yoziladi:

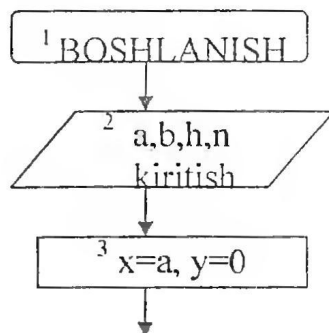
$$y = \int_a^b f(x) dx \approx \frac{b-a}{n} (y_0 + y_1 + y_2 + \dots + y_n) \quad (2.30)$$

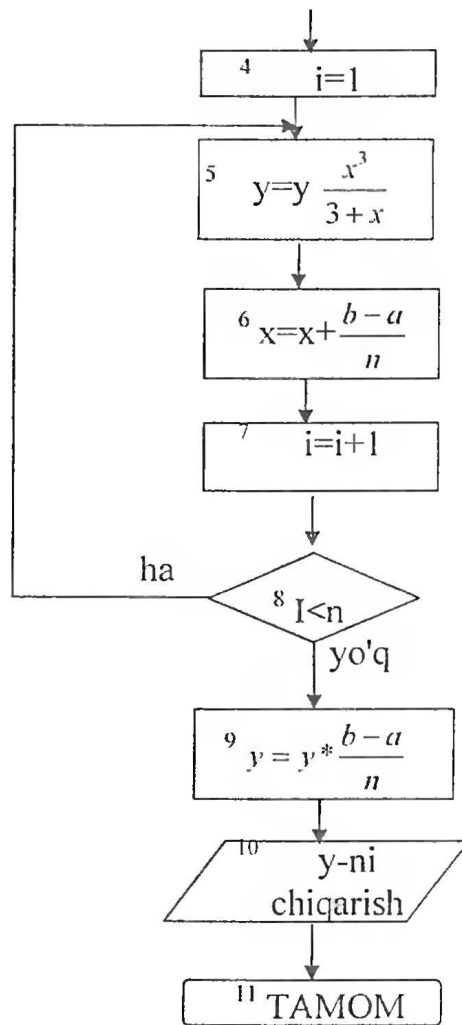
$$y = \int_a^b f(x) dx \approx \frac{b-a}{n} \sum_{i=0}^n f(x_i) \quad (2.31)$$

To'g'ri burchakliklar usulining  $h$  qadami o'zgarmas bulganda integrallash aniqligi  $E=h$  teng,  $h \ll 1$ .

**Misol.** 20 bo'lakli  $[1,2]$  kesmada  $y = \int_a^b \frac{x^3}{3x} dx$  integralni hisoblash algoritmi va dasturi tuzilsin.

Misolning blok-sxemasini 2.14-shaklda ko'rsatilgan.





2.14-shakl.

Yo'qoridagi algoritmgaga muvofiq tuzilgan dastur va uning bajarilgan natijasini keltiramiz.

```

PROGRAM TO'GRIBURCHAK;
VAR a,b,x,y: REAL;
i,j,n: INTEGER;
BEGIN
  READLN (a,b,N);
  X:=a; y=0;
  FOR I:=1 TO N DO

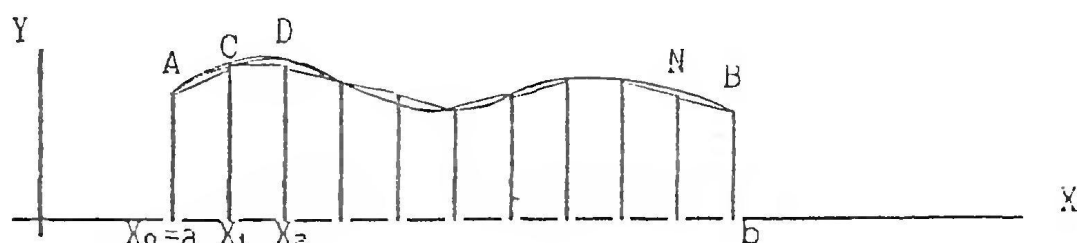
```

```

BEGIN
Y:=y+l(x);
X:=x+(b-a)/n;
END;
Y:=y+(b-a)/n;
WRITELN ('y=', y);
END.

```

## 2.4.2. TRAPETSIYA USULI



2.15-shakl.

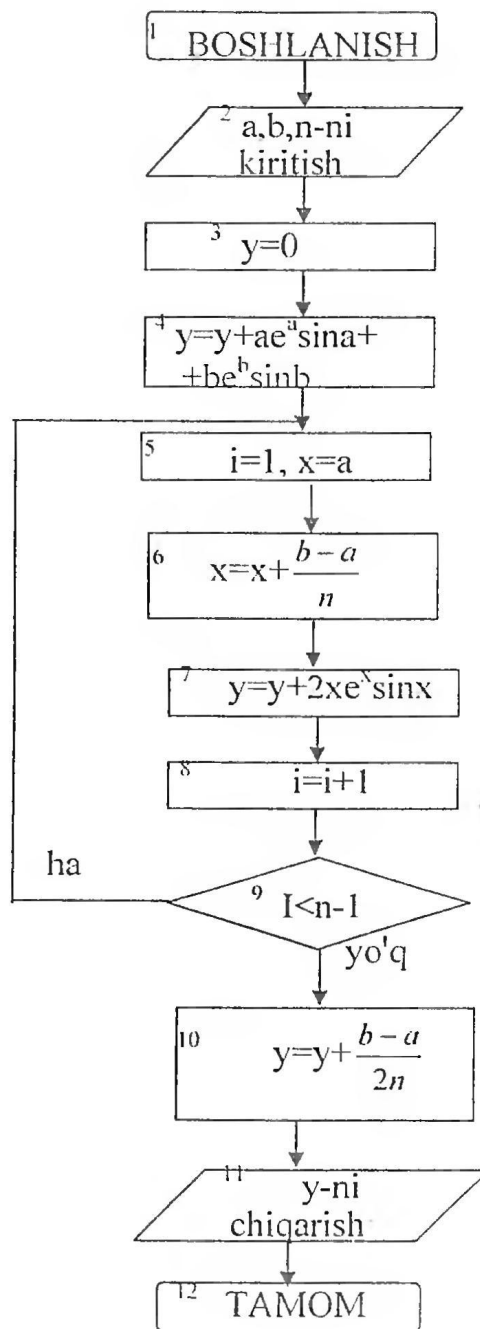
Bu shakl yuzasi  $S$  trapetsiyalar yuzasining yig'indisiga teng, ya'ni

$$S = h \left( \frac{Y_0 + Y_1}{2} + \frac{Y_1 + Y_2}{2} + \dots + \frac{Y_{n-1} + Y_n}{2} \right) = \frac{b-a}{2n} (Y_0 + 2Y_1 + 2Y_2 + \dots + 2Y_{n-1} + Y_n) \quad (2.32)$$

Shunday qilib, trapetsiya formulasiga binoan integralning taqribiy qiymati

$$Y = \int_a^b f(x) dx = \frac{b-a}{2n} (Y_0 + 2Y_1 + 2Y_2 + \dots + 2Y_{n-1} + Y_n)$$

yoki  $Y = \int_a^b f(x) dx = \frac{b-a}{2n} (Y_0 + Y_n + \sum_{i=1}^{n-1} 2Y_i)$  ko'rinishga ega bo'ladi.



2.16 – shakl.

**Misol.**  $[0,1]$  kesmada 40 bo'lakli  $[0,1]$   $Y = \int_a^b x e^x \sin x dx$  integralni hisoblab chiqishning algoritmi va dasturini tuzish trapetsiyalar usuli orqali bajarilsin.



Masala yechish algoritmining blok-sxemasi 2.16-shaklda kursatilgan, keyin keltirilgan algoritmgaga muvofiq tuzilgan dastur va uning bajarilgan natijasi keltirilgan.

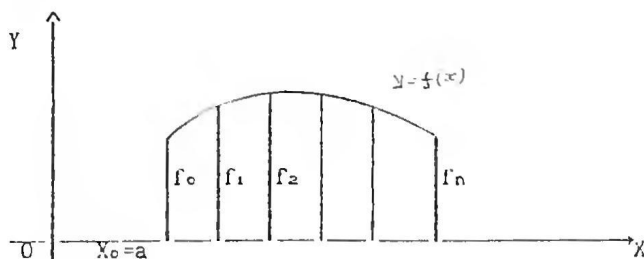
```

PROGRAM TRAPACION USULI;
VAR a,b,,x,y: REAL;
I, A: INTEGER;
BEGIN
READLN (a,b,n);
Y:=0;
Y:=y+f(a)+f(b);
X:=a;
FOR I:=1 TO N-1 DO
BEGIN
X:=x+(b-a)/n;
Y:=y+2*f(x);
END;
Y=y*(b-a)/(2*M);
WRITELN ('y=', y);
END.

```

### 2.4.3. SIMPSON USULI

Simpson usuli  $h$  qadamni oshirmasdan aniq integralni hisoblab chiqishning yo'qori aniqligini ta'min qiladi. Bu usul quyidagilardan iborat: integral ostidagi egri chiziqli funksiya kvadratli parabolalar ulushlaridan iborat bo'lakli-uzluksiz chiziq bilan almashtiriladi. Bunda integrallash oraligi  $2n$  ta, har birini uzunligi  $h=(b-a)/2n$  ga teng bo'laklarga ajratiladi.



2.17-shakl.

Bunda sonli integrallash uchun Simpson formulasi quyidagicha ko'rinishga ega bo'ladi:

$$y = \int_a^b f(x) dx \cong \frac{b-a}{6n} (y_0 + 4y_1 + 2y_2 + 4y_3 + 2y_4 + \dots + 4y_{2n-1} + y_{2n})$$

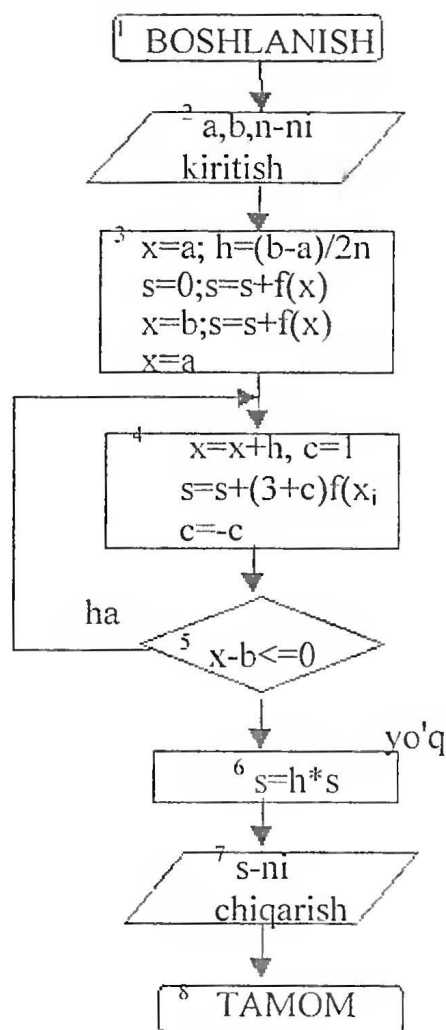
$$\text{yoki} \quad y = \int_a^b f(x) dx \cong \frac{h}{3} [y_0 + y_n + \sum_{i=1}^{n-1} (3 + c_i) y_i] \quad (2.33)$$

bunda  $C^i = +1$  toq  $i$  holda;  $-1$  juft  $i$  holda

**Misol.** [0.2] kesmani 50 bo'lakka teng bo'lish

$$y = \int_a^b x^2 \sqrt{4-x^2}$$

Integralni hisoblash algoritmi va dasturi Simpson usulida tuzilsin. Misol yechish algoritmining blok-sxemasi 2.18-shaklda ko'rsatilgan. Keyin dastur va uning bajarilgan natijasi keltirilgan.



2.18-shakl.

```

PROGRAM SIMPSON USULI;
VAR a,b,x,y: REAL;
I,e,n: INTEGER;
BEGIN
READLN (a,b,n);
Y:=0;
Y:=y+f(a)+f(b);
X:=a; c:=1;
FOR I:=1 TO N-1 DO
BEGIN
X:=x+(b-a)/n;
Y:=y+(3+c)*f(x);
C:=c;
END;
Y:=y*(b-a)/(3*n);
WRITELN ('Y=', y)
END.

```

### TEKSHIRISH UCHUN SAVOLLAR

1. Boshlang'ich ta'rif.
2. Aniq integralning ta'ri.
3. To'g'ri to'rtburchakli usulining afzalligi va kamchiligi
4. Trapetsiya usuli.
5. Qaysi maqsad bilan Simpson formulasida "S" o'zgaruvchi kiritilgan?
6. Nima uchun 90-satrd S o'zgaruvchi o'z ishorasini o'zgartirdi.

### MASHQLAR

Aniq integralni hisoblashda blok-sxema va dastur tuzing.

1. 
$$Y = \int_1^{3.5} \frac{\ln x}{x\sqrt{1+\ln x}} dx; \quad n = 30 \text{ Simpson usulida}$$
2. 
$$Y = \int (tg^2 x + ctg^2 x) dx; \quad n = 54 \text{ Trapetsiya usulida}$$
3. 
$$Y = \int_2^3 \frac{1}{x \lg x}; \quad n = 36 \quad \text{To'g'ri burchakli usulida}$$

4.  $Y = \int_1^4 \frac{\ln^2 x}{x} dx; \quad n = 32 \quad \text{To'g'ri burchakli usulida}$
5.  $Y = \int_0^2 \sqrt{e^x - 1} dx; \quad n = 106 \quad \text{Simpson usulida}$
6.  $Y = \int_0^1 x e^x \sin x dx; \quad n = 46 \quad \text{To'g'ri burchakli usulida}$
7.  $Y = \int_0^e x \sinh x dx; \quad n = 48 \quad \text{Simpson usulida}$
8.  $Y = \int_0^2 \frac{1}{\sqrt{9 + x^2}} dx; \quad n = 206 \quad \text{Trapetsiya usulida}$
9.  $Y = \int_1^{25} \frac{1}{x^2} \sin \frac{1}{x} dx; \quad n = 44 \quad \text{To'g'ri burchakli usulida}$
10.  $Y = \int_0^{\sqrt{3}} x \arctg x dx; \quad n = 48 \quad \text{Simpson usulida}$
11.  $Y = \int_0^3 \arcsin \sqrt{\frac{x}{1+x}} dx; \quad n = 36 \quad \text{Trapetsiya usulida}$
12.  $Y = \int_1^3 x^x (1 + \ln x) dx; \quad n = 40 \quad \text{To'g'ri burchakli usulida}$
13.  $Y = \int_0^1 \frac{dx}{\sqrt{1 + 3x + 2x^2}}; \quad n = 44 \quad \text{Simpson usulida}$
14.  $Y = \int_1^2 \frac{\sqrt{x^2 - 0.16}}{x} dx; \quad n = 160 \quad \text{Trapetsiya usulida}$
15.  $Y = \int_0^1 \frac{x \arctg x}{\sqrt{1 + x^2}} dx; \quad n = 40 \quad \text{To'g'ri burchakli usulida}$
16.  $Y = \int_0^2 \frac{e^{3x+1}}{e^{x+1}} dx; \quad n = 50 \quad \text{Simpson usulida}$

17.  $Y = \int_0^1 \sin^2 x dx;$   $n = 30$  Trapetsiya usulida
18.  $Y = \int_0^2 x^2 \sqrt{4-x^2} dx$   $n = 100$  To'g'ri burchakli usulida
19.  $Y = \int_0^1 e^x \cos^2 x dx;$   $n = 60$  Simpson usulida
20.  $Y = \int_0^3 \arcsin \sqrt{\frac{x}{1+x}} dx;$   $n = 50$  Trapetsiya usulida
21.  $Y = \int_0^1 \frac{x^2-1}{(x^2+1)\sqrt{x^4+1}} dx;$   $n = 30$  To'g'ri burchakli usulida
22.  $Y = \int_1^{1.5} \sin x \ln(x) dx;$   $n = 40$  Simpson usulida
23.  $Y = \int_1^2 \frac{x}{x^4+3x^2+2} dx;$   $n = 70$  Trapetsiya usulida

## 2.5. DIFFERENSIAL TENGLAMALARNING TARQIBIY ECHIMLARI

### 2.5.1. EYLER USULI

Differensial tenglamalarni yechishning oddiy sonli usullaridan biri Eyler usuli.

Boshlang'ich shartlari  $x_0, y(x_0)=y_0$  bo'lgan birinchi tartibdagi

$$\frac{dy}{dx} = f(x, y) \quad (2.34)$$

differensial tenglamasi sonli hisoblash usuli bilan yechish uchun yechimni  $h$  atrofida Teylor qatorlariga yoyamiz.

$$y_1 = y(x_0 + h) = y(x_0) + hy'(x_0) + \frac{h^2}{2!} y''(x_0) + \dots$$

Qatorning ikkinchi va yo'qori tartibdagi hosilalar mavjud bo'lgan barcha qator hadlarini tashlab yuborganda,  $y' = y(x_0 + h) = y_0 + hf(x_0, y_0)$  ga ega bo'lamiz.

Bunda  $f(x, y)$  - differensial tenglamaning o'ng qismi (2.34)

$Y_1$  qiymati  $h$  atrofida  $x_1 = x_0 + h$  tengligini hisobga olib  $y(x)$  yoyilmasini yozamiz.

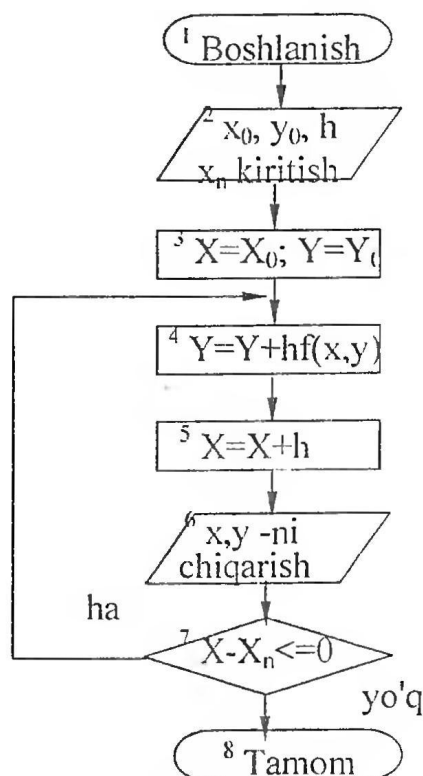
$$\begin{aligned} y_2 &= y(x_0 + 2h) = y_1 + hf(x_1, y_1) \\ y_3 &= y_2 + hf(x_2, y_2) \\ y_4 &= y_3 + hf(x_3, y_3) \\ y_{i+1} &= y_i + hf(x_i, y_i) \end{aligned} \quad i = 1, 2, 3 \dots n \quad (2.35)$$

ega bo'lamiz.

$h$  qadami etarli kichik bo'lsa, Euler usuli yechimni yo'qori aniqlikda beradi; so'nggi yechim xatoligi integrallashning har qadamida  $h^2$  ( $h \ll 1$ ) ga yaqin.

Tenglamani integrallash uchun boshlang'ich ma'lumotlar:  $x_0, y_0$  boshlang'ich qiymatlari, integrallash qadami  $h$  va integrallash oralig'i oxirgi  $x_n$  abssissa qiymati beriladi.

Differensial tenglamalarning Euler usulidagi taqribiy yechimlari bilan algoritmi va blok-sxemasi 2.19-shaklda keltirilgan.



2.19 – shakl.

```

PROGRAM EILER USULI;
VAR X0, Y0,XN,X,Y,H: REAL
BEGIN
READLN (X0,Y0,XN,H);
X:=X0; Y:=Y0;
WHILE (X-XN)<=0 DO
BEGIN
Y:=Y+H*f(x,y);
X:=X+H;
WRITELN ('X=', x, 'Y=', y);
END;
END.

```

Eyler usulining kamchiligi shuki, yechimga aniqlik beruvchi  $h$  qadami kichik qiymatga ega bo'lgan taqdirda hisoblab chiqishni sekinlashtiradi.

### 2.5.2. RUNGE-KUTTA USULI

Differensial tenglamalarni sonli integrallashning juda keng tarqalgani Runge-Kutta usuli bo'lib, u har bir qadamda hisoblash aniqligi yo'qori bo'lishi tufayli hisobni tezlashtirishni ta'minlaydi. Runge-Kutta usuli aniqligi  $E=h^2$  qiymati bilan belgilanadi.

Aniqlikni oshirishga to'rtta nuqta koordinatalarini maxsus tanlash orqali erishiladi; bu nuqtalarda  $f(x,y)$ ning birinchi hosilasi hisoblab chiqiladi. Eyler formulasida ishlatiladigan  $hf(x,y)$ , birinchi hosila o'rniga o'rtalangan birinchi hosila  $f_1$  ning qiymati topiladi.

Runge-Kutta usuli bo'yicha integrallash formulalari ko'rinishi quyidagicha:

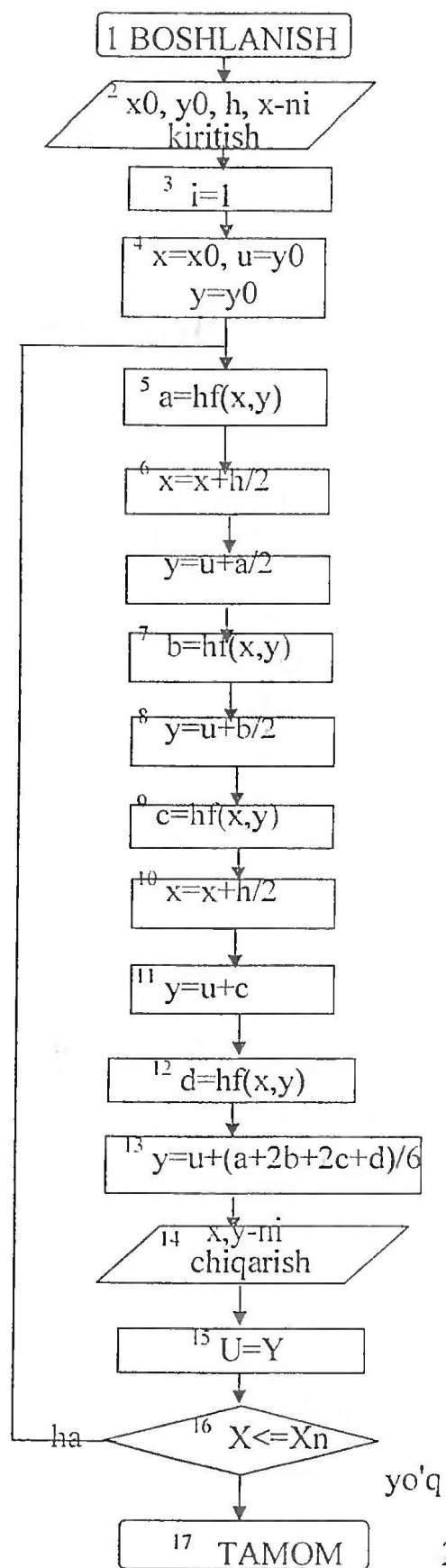
$$Y_{i+1}=Y_i+F_i, \quad (2.36)$$

Bunda

$$F_i=(a_i+2b_i+2c_i+d_i)/6;$$

$$a_i=hf(x_i, y_i)$$

$$b_i=hf(x_i+h/2, y_i+a_i/2)$$



2.20-shakl.



```

PROGRAM RUNGE_KUTTA;
LABEL 1;
VAR X0,Y0,XH, H, Y, X; REAL;
A,B,C,D,V; REAL;
BEGIN
READLN (X0, Y0, XN, H);
X:=X0; Y:=Y0; V:=V0;
1: A:=H*f(x,y);
X:=X+H/2;Y:=V+A/2;
B:=H*f(x+H/2; Y+a/2;
Y:=V+B/2;
C:=H*f(x+H/2, Y+b/2);
X:=X+H/2;
Y:=V+C;
D:=H*f(x+H, y+C);
Y:=V+(A+2*B+2*C+D)/6;
WRITELN ('x=', x, 'y=', y);
V:=v;
IF X<=XN THEN GOTO 1;
END.

```

## MASHQLAR

Differensial tenglamani yechishda blok-sxema va dastur tuzilsin.

1.  $y' = \ln y(\cos^3 x + 1) + \frac{x^3}{y}$  Eyler usulida. Boshlang'ich shart  $x_0=2.3$ ;  $y_0=4.7$ ;  
 $h_0=0.01$ ;  $n=10$
2.  $y' = \frac{\sin^2 x + 2xy}{e^x + x^2}$  Runge-Kutta usulida. Boshlang'ich shart  $x_0=7$ ;  $y_0=4$ ;  
 $h_0=0.01$ ;  $n=20$
3.  $y' = \frac{y^x}{\cos x + y}$  Eyler usulida. Boshlang'ich shart  $x_0=2.7$ ;  $y_0=3.2$ ;  $h_0=0.001$ ;  
 $n=100$
4.  $y' - \frac{y' y^x}{x+1} + (x+1) = 0$  Runge-Kutta usulida. Boshlang'ich shart  $x_0=5.3$ ;  $y_0=7$ ;  
 $h_0=0.01$ ;  $n=50$
5.  $y' = (ax^2 + bxy)/x^a$  Eyler usulida  $x_0=7$ ;  $y_0=4$ ;  $h_0=0.01$ ;  $n=10$  o'zgaruvchi a,b,c  
- ixtiyoriy son.

6.  $(y'^2 + x)y' - 2xyy' = 12.4$  Runge-Kutta usuli. Boshlang'ich shart  $x_0=1.4$ ;  $y_0=3$ ;  $h_0=0.001$ ;  $n=10$
7.  $y' = \frac{\cos^2 x + 2xy'^0}{e^x + y^2}$  Eyler usuli  $x_0=5.3$ ;  $y_0=7.7$ ;  $h_0=0.001$ ;  $n=10$
8.  $y' = \frac{x^y}{\sin x - y}$  Runge-Kutta usulida. Boshlang'ich shart  $x_0=3.75$ ;  $y_0=5$ ;  $h_0=0.05$ ;  $n=20$
9.  $y' = yx^2 + yx + 2y$  Eyler usulida. Boshlang'ich shart  $x_0=0.15$ ;  $y_0=-21$ ;  $h_0=0.002$ ;
10.  $y' + ax^2b^x(e^x + 1) = 0$  Runge-Kutta usulida. Boshlang'ich  $x_0=0.5$ ;  $y_0=1$ ;  $h_0=0.0005$ ;  $n=10$ ;  $A=2.71$ ;  $b=1.06$

## ADABIYOTLAR

1. В.Е. Фигурнов. IBM PC для пользователя. М.; 1997.
2. А.Кенин. IBM PC для пользователей или как научиться работать на компьютере. М.; 1997.
3. В.М. Брябрин. Программное обеспечение персональных ЭВМ. М.; 1998.
4. А.В. Петров и другие. Вычислительная техника в инженерных и экономических расчетах. М.; 1986.
5. Д. Прайс . Программирование на языке Paskal. 1982.
6. Р. Форсайт. Paskal для всех. 1986.
7. А.Файсман. Профессиональное программирование на Turbo-Paskale. Т., 1992
8. В.М. Гостев. Среда программирования Turbo-Paskal .К.,1995.
9. А.М.Епанешников, В.А.Епанешников. Программирование в среде Turbo-Paskal. М., 1993.
10. М.М.Аliyev Algoritm lash asoslari va Beysikda programmalashtirish. Т., Mehnat, 1992.
11. Журналы "Информатика и образование". М.; 1998, 1999, 2000, 2001.
12. Delphi 4, Delphi 5, Delphi 6. Инструкция пользователя. 1998,1999,2000.
13. Кулвин Н. Основы программирования в Delphi 7. Санкт-Петербург, "БХВ-Петербург", 2003.

## MUNDARIJA

|                                                                                       |     |
|---------------------------------------------------------------------------------------|-----|
| 1. Borland Delphi vositasi                                                            | 3   |
| 1.1. Delphi ni ishga tushirish                                                        | 4   |
| 1.2. Delphi ni dasturlash muhiti                                                      | 12  |
| 1.2.1. Dasturlash muhit asosiy qismlari                                               | 12  |
| 1.2.2. Standartli komponentlar                                                        | 16  |
| 1.2.3. Obyektlar Inspektori to'g'risida batafsilroq ma'lumot                          | 19  |
| 1.2.4. Dasturlarni saqlab qolish                                                      | 22  |
| 1.2.5. Tbutton, dastlabki matn, sarlavhalar va z - tartiblash                         | 22  |
| 1.3. Loyiha ustidan boshqarish                                                        | 25  |
| 1.3.1. Boshqa menyular bilan ishlash                                                  | 28  |
| 1.3.2. Dasturlash muhitini konfiguratsiyalash (IDE)                                   | 36  |
| 1.4. Komponentlar qatori to'g'risida qisqacha ma'lumot                                | 40  |
| 1.4.1. Standartli komponentlar                                                        | 40  |
| 1.4.2. Additional betidagi komponentlar                                               | 41  |
| 1.4.3. Dialogs beti                                                                   | 42  |
| 1.4.4. System beti                                                                    | 43  |
| 1.5. Delphi dagi xususiyatlar                                                         | 45  |
| 1.5.1. Bajarish rejimida vizualli komponentlarning<br>xususiyatlari ustida boshqarish | 48  |
| 1.5.2. SHAPEDEM2 dasturi                                                              | 55  |
| 1.6. Delphi da usullar                                                                | 63  |
| 1.6.1. Vizualli vositalar yordamida usullar barpo etish                               | 64  |
| 1.6.2. Parametrlarni uzatish                                                          | 67  |
| 1.6.3. Murakkabroq bo'lgan usullar va boshqaruvchi elementlar                         | 72  |
| 1.6.4. Bajarish davrning axboroti. CONTROL3 dasturi                                   | 75  |
| 1.7. Derazali yondoshlar                                                              | 77  |
| 1.7.1. Derazali yondoshlarni barpo etish bosqichlari                                  | 78  |
| 1.7.2. Delphi loyihasining strukturasi                                                | 83  |
| 1.7.3. Oddiy derazali yondoshni barpo etish misoli                                    | 87  |
| 1.8. Sinflar va obyektlar                                                             | 98  |
| 1.8.1. Obyekt bo'yicha oriyentatsiyalangan dasturlashning<br>asosiy prinsiplari       | 98  |
| 1.8.2. Maydonlar                                                                      | 100 |
| 1.8.3. Usullar                                                                        | 101 |
| 1.8.4. Xususiyatlar                                                                   | 106 |

|                                                                |     |
|----------------------------------------------------------------|-----|
| 1.8.5. Hodisalar                                               | 108 |
| 1.8.6. Sinf elementlarining ko'rinish sohalari                 | 110 |
| 2. Kompyuterda sonli hisoblash usuli bilan masalalarni yechish | 111 |
| 2.1. Algebraik va transsendent tenglamalar                     | 111 |
| 2.1.1. Lagranj formulasi                                       | 112 |
| 2.1.2. O'zgaruvchi ishorali yig'indilar usuli                  | 113 |
| 2.1.3. Ketma-ket tanlash usuli                                 | 114 |
| 2.1.4. Yarim bo'lish usuli                                     | 118 |
| 2.1.5. Nyuton usuli                                            | 122 |
| 2.1.6. Iteratsiya usuli                                        | 125 |
| 2.2. Funksiyalarni interpolyatsiyalash                         | 130 |
| 2.3. Chiziqli tenglamalar sistemasining yechimi                | 136 |
| 2.4. Funksiyalarni sonli integrallash                          | 141 |
| 2.4.1. To'g'riburchaklilar usuli                               | 142 |
| 2.4.2. Trapetsiya usuli                                        | 144 |
| 2.4.3. Simpson usuli                                           | 146 |
| 2.5. Differensial tenglamalarning tarqibiy yechimlari          | 150 |
| 2.5.1. Eyler usuli                                             | 150 |
| 2.5.2. Runge-Kutta usuli                                       | 152 |
| Adabiyotlar                                                    | 155 |
| Mundarija                                                      | 156 |

|                        |                 |                        |
|------------------------|-----------------|------------------------|
|                        |                 | Muharrir-P.A`zamova    |
| Bosishga ruxsat etildi | Hajmi 9,75 b.t. | Buyurtma № 406         |
| Bichimi 60x80 1/16     | Adadi 100 nusxa |                        |
| ToshTYMI bosmaxonasi   | Toshkent,       | Odilxo'jayev ko'ch.,1. |





