O`ZBEKISTON RESPUBLIKASI OLIY VA O`RTA MAXSUS TA'LIM VAZIRLIGI

TOSHKENT TO`QIMACHILIK VA YENGIL SANOAT INSTITUTI

"Informatika" kafedrasi

Dasturlash asoslari

Dasturlash asoslari

Toshkent-2011y.

A n n o t a t s i ya

Qo`llanma Pascal va Delphi dasturiy vositalarida dasturlar yaratishga bag`ishlangan. Unda Pascal tili asosiy elementlari va Delphining asosiy komponentalari haqida ma'lumotlar keltirilgan. Qo`llanmadan Oliy texnika o`quv yurtlarida, akademik litsey va kasb hunar kollejlarida informatika va axborot texnologiyalari fanidan dasturlash asoslari bo`yicha ma'ruza va amaliy, hamda mustaqil mashg`ulotlarni olib borishda foydalanish mumkin.

Tuzuvchilar: F-m.f.n., dotsent Ne'matov A.
Katta o`qituvchi Kulmuradov M.
Katta o`qituvchi Tangirov A.
Assisent Akbarova N.

Ma'sul muharir. f.-m.f.n., dotsent Atamirzaev M.

Taqrizchilar: Axborot texnologiyalar Universiteti
"Informatika" kafedrasi
mudiri, professor Nazirov SH.

TTESI, "Texnologik jarayonlarni
kompyuterlashtirish"
kafedrasi mudiri, t.f.n., dotsent Siddiqov I.

Toshkent to`qimachilik va engil sanoat
instituti ilmiy uslubiy kengashida
tasdiqlangan.
17. 05. 2011 y. Bayonnoma № 8

TTYESI bosmaxonasida 30 nusxada ko`paytirilgan

M U N D A R I J A

K I R I SH

Kompyuter texnologiyasining kun sayin rivojlanib borishi bilan kompyuterda dasturlashga bo`lgan qiziqish ham toboro ortib bormoqda. Ayniqsa vizual dasturlash texnologiyalaridan foydalanib dasturlar yaratish kompyuter texnologiyasining rivojlanishiga katta ta’sir etmoqda. Yangidan yangi dasturlash vositalarining yaratilishi esa nafaqat professional dasturchilar, balki oddiy dastur tuzuvchilar uchun ham keng yo`l ochib bermoqda.

Qo`llanma 15ta bo`limdan iborat. Birinchi bo`limda algoritm tushunchalari, ikkinchi bo`limdan to`qqizinchi bo`limlarda Pascal tilining asosiy operatorlariga qisqacha tushuntirish berilgan bo`lib har bir bo`lim uchun misollar keltirilgan. O`ninchidan o`nbeshinchi bo`limgacha bo`lgan mavzularda Delphi dasturiy vositasi komponentalarida vizual dasturlash texnologiyalari keltirilgan va misollar keltirilib ularni bajarish tartibi berilgan.

Qo`llanmada asosiy e’tibor dasturlar tuzish usullariga qaratilgan bo`lib, uning yordamida o`quvchi kompyuterda tez mustaqil holda dastur tuzish imkoniga ega bo`ladi va zamonaviy vizual dasturlash texnologiyalari bilan tanishadi.

1.ALGORITM VA DASTUR TUSHUNCHALARI

Algoritm so`zi buyuk matematik Al-Xorazmiyning nomi bilan bog`liq bo`lib, u birinchi bo`lib arab raqamlaridan foydalangan holda arifmetik amallarni bajarish qoidasini bayon etdi.

Har qanday qo`yilgan masalani kompyuterda yechish uchun oldin uning yechish usulini tanlab, keyin uning algoritmini ishlab chiqish kerak bo`ladi. Demak, hech bir masala yo`qki uning echilish yo`llarini bilmasdan va algoritmini tasavvur qilmasdan turib uni kompyuterda echib bo`lmaydi.

1.1.Algoritm va uning xossalari

Elektron hisoblash mashinalarining vujudga kelishiga qadar algoritmga har xil ta’rif berib kelindi. Lekin ularning barchasi ma’no jihatdan bir-biriga juda yaqin bo`lib, bu ta’rif hozirgi kunda quyidagicha talqin qilinadi.

Ta’rif. Algoritm deb, qo`yilgan masalani yechish uchun ma’lum qoidaga binoan bajariladigan amallarning chekli qadamlar ketma-ketligiga aytiladi.

Har qanday algoritm ma’lum ko`rsatmalarga binoan bajariladi va bu ko`rsatmalarga buyruq deyiladi.

Algoritm quyidagi muhim xossalarga ega:

Aniqlik va tushunarlilik - deganda algoritmda ijrochiga berilayotgan ko`rsatmalar aniq mazmunda bo`lishi tushuniladi.

Ommaviylik - deganda har bir algoritm mazmuniga ko`ra bir turdagi masalalarning barchasi uchun ham o`rinli bo`lishi tushuniladi.

Natijaviylik - deganda algoritmda chekli qadamlardan so`ng albatta natija bo`lishi tushuniladi.

Diskretlik - deganda algoritmlarni chekli qadamlardan tashkil qilib bo`laklash imkoniyati tushuniladi.

1.2.Algoritm turlari

Algoritmning uchta turi bor: chiziqli, tarmoqlanuvchi va takrorlanuvchi.

CHiziqli algoritm - deb hech qanday shartsiz faqat ketma-ket bajariladigan jarayonlarga aytiladi.

Tarmoqlanuvchi algoritm - deb ma’lum shartlarga muvofiq bajariladigan ko`rsatmalardan tuzilgan algoritmga aytiladi.

Takrorlanuvchi algoritm - deb biron bir shart tekshirilishi yoki biron parametrning har xil qiymatlari asosida algoritmda takrorlanish yuz beradigan jarayonlarga aytiladi.

Algoritmlarni turli usullarda tasvirlash mumkin. Masalan: so`z bilan ifodalash; formulalarda berish; blok-sxemalarda tasvirlash; dastur shaklida ifodalash va boshqalar.

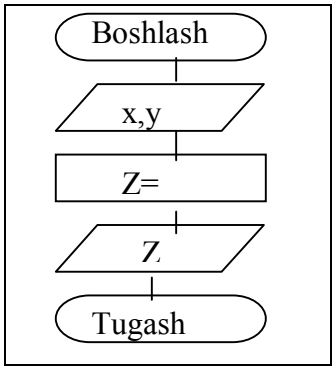
Algoritmlarni blok-sxema ko`rinishda tasvirlash qulay va tushunarli bo`lgani uchun ko`p ishlatiladi. Bunda algoritmdagi har bir ko`rsatma o`z shakliga ega. Masalan: parallelogramm ko`rinishdagi belgi ma'lumotlarni kiritish va chiqarish; to`g`ri to`rtburchak belgisi hisoblash jarayonini; romb belgisi shartlarning tekshirilishini bildiradi.

Misollar: CHiziqli algoritmga doir: $Z = \frac{x^2 + \sin(2x-1)}{\sqrt{x^2 + y^2 + 10}}$ hisoblash algoritmini tuzing.

So`zda berilishi:

1.Boshlash.
2.x va u qiymatini kiritish.
3.z qiymatini hisoblash.
4.z qiymatini chiqarish.
5.Tamom.

Blok-sxemada:

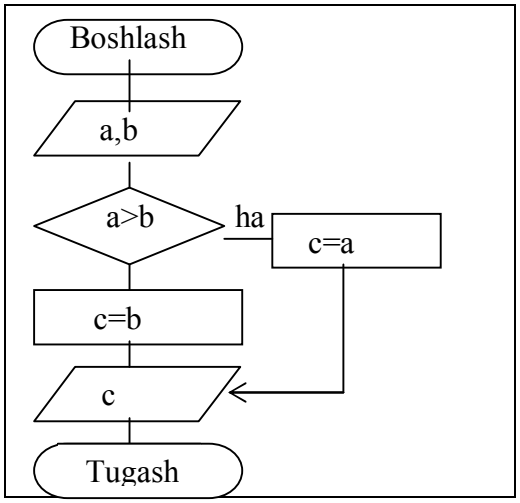


Tarmoqlanuvchi algoritmga doir:
Ikkita a va b sonlardan kattasini aniqlash algoritmini tuzing.

So`zda berilishi:

1.Boshlash.
2.a va b-qiymatini kiritish.
3.agar a>b bo`lsa, natija a deb olinib 5ga o`tilsin.
4.natija b deb olinsin.
5.Tamom.

Blok-sxemada:



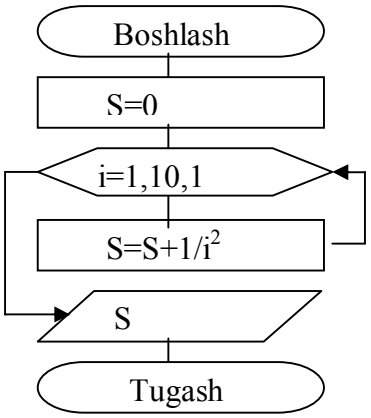
Takrorlanuvchi algoritmga doir:

$S = \sum_{i=1}^{10} \frac{1}{i^2} = 1 + \frac{1}{4} + \frac{1}{9} + ... + \frac{1}{100}$ yig`indini hisoblash algoritmini tuzing.

So`zda berilishi:

1. Boshlash.
2. S=0
3. i=1
4. $S=1/i^2$
5. Agar i<100 bo`lsa, u holda i=i+1 va 4 ga o`tish. Aks holda 6 ga.
6. S qiymati chiqarilsin.
7. Tamom

Blok-sxemada:



Hayotimizda algoritmlarni turli sohalarda ba'zan bilgan holda ba'zan esa bilmagan holda ishlatamiz. Algoritmlar faqat matematik xarakterga ega bo`lmasdan ularni oddiy hayotiy turmushimizda ham ko`p qo`llaymiz. Masalan, ovqat tayyorlash, choy damlash, biror berilgan ishni bajarish va boshqa. Bu ishlarni bajarishda ma'lum bo`lgan aniq ko`rsatmalarni ketma ket bajaramiz. Agar bu ko`rsatmalar aniq bir ketma ketlik tartibida bajarilmasa kerakli natijani olaolmaymiz. Misol tariqasida matematik xarakterga ega bo`lmagan butelbrod tayyorlash algoritmini ko`rib chiqaylik. Bunda boshlang`ich berilganlar: non, kolbasa va pishloq. Natija: butelbrod. Butelbrod tayyorlash algoritmi:

- non bo`lagini kesib olish;
- kolbasa va pishloq bo`lagini kesib olish;
- kolbasa va pishloq bo`lagini non bo`lagi orasiga qo`yish.

Agar bu jarayonning ketma ketlik o`rinlari almashsa yoki biror bir bosqich amalga oshirilmasa natija bo`lmaydi.

1.3.Algoritmik tillar

Masalani yechish algoritmi ishlab chiqilgandan so`ng dastur tuziladi. **Dastur** - bu berilgan algoritmga asoslangan biror bir algoritmik tilda yozilgan ko`rsatmalar, ya'ni buyruqlar yoki operatorlar to`plamidir. **Dasturlash -** esa bu dastur tuzish jarayoni bo`lib, u quyidagi bosqichlardan iboratdir:

- dasturga bo`lgan talablar;
- qo`yilgan masala algoritmini tanlash yoki ishlab chiqish;
- dastur kodlarini (matnlari, buyruqlarni) yozish;
- dasturni to`g`rilash va test o`tkazish.

Hozirgi kunda juda ko`plab algoritmik tillar mavjud. Ularga dasturlash tillari deb ataymiz. Algoritmik til - algoritmlarni bir xil va aniq yozish uchun ishlatiladigan belgilashlar va qoidalar tizimidir. Algoritmik til oddiy tilga yaqin bo`lib u matematik belgilarni o`z ichiga oladi. Tuzilgan algoritmlarni to`g`ridan-to`g`ri mashinaga berib bo`lmaydi, shu sababli yozilgan algoritmni biror bir algoritmik tilga o`tkazish zarur. Har qanday algoritmik til o`z qo`llanilish sohasiga

ega. Masalan, muxandislik hisob ishlarini bajarishda Pascal, Beysik va Fortran. Iqtisod masalalarini yechishda Pascal va Kobol. Mantiqiy dasturlash uchun Prolog va boshqalar. O`quv jarayonlari uchun Beysik, Pascal va boshqalar.

Pascal, Fortran va Kobol tillari universal tillardan hisoblanadi. Assembler tili mashina tiliga ancha yaqin til bo`lib o`rta darajadagi tildir. Algoritmik til inson tillariga qancha yaqin bo`lsa, u tilga yuqori darajali til deyiladi. Mashina tili esa eng pastki darajali tildir.

1.4.Masalalarni yechish bosqichlari

Kompyuterda masalani yechish bosqichlari quyidagilar:

1.Masalani qo`yish va uning matematik modelini ishlab chiqish. Ushbu bosqichda boshlang`ich malumotlar tarkibi aniqlanadi. Masalani qo`yish odatda echiladigan masalaning asosiy xususiyatlarini og`zaki tavsiflash bilan tugallanadi va so`ngra, masala matematik modeli yoziladi.

2.Yechish usulini tanlash. Masala matematik ifodalanib bo`lgandan so`ng uni yechish usuli tanlanadi. Bunda izlanayotgan natijalarning boshlang`ich ma’lumotlarga bog`liqligini aniqlaydi. Hozirgi zamon hisoblash matematikasi fan va texnikaga oid turli masalalarni yechish uchun ko`plab sonli usullarga ega.

3.Masalani yechish algoritmini ishlab chiqish. Bu bosqichda kompyuterda masalani yechish uchun bajariladigan amallar (buyruqlar) ketma-ketligi tavsiflanadi. Biz buni algoritmlash deb ataymiz.

4.Dasturlash. Bunda algoritm biror algoritmik tilga ko`chiriladi.

5.Dasturni kiritish va xatolarini tuzatish.

6.Masalani bevosita kompyuterda yechish va natijalarni tahlil qilish. Bunda dasturda kerakli boshlang`ich qiymatlar berilib kompyuterda natija olinadi va tahlil qilinadi. Bu bosqichda agar kerak bo`lsa algoritm va dastur modernizatsiya qilinishi ham mumkin. Yangi olingan ma’lumotlar asosida kerakli xulosalar ishlab chiqiladi. Bu ma’lumotlar qo`yilgan masalani turlicha tahlil qilishga, murakkab jarayonlarni tushunishga, olamshumul yangiliklarning ochilishiga, yangi nazariyalarning, texnika mo`’jizalarining yaratilishiga xizmat qiladi. Umuman olganda «Model-algoritm-dastur» uchligi modellashtirishning intelektual mag`zini tashkil etadi, bunday uchliksiz kompyuterda murakkab masalalarni yechishda muvaffaqiyatga erishib bo`lmaydi.

1.5.Dasturlash vositalari

Dasturlash vositalarining asosiy vazifasi dastur tuzuvchilarga kompyuterda dastur matnlarini kiritish, tahrirlash va ishga tushurish kabi bir qancha ishlarni bajarishda qulaylik yaratadi.

Turbo Pascal dasturlash vositasi Pascal tilida kompyuterda dasturlar yaratish uchun maxsus dastur bo`lib xizmat qiladi. Turbo Pascal dasturlash vositasi ham o`z matn muhaririga ega bo`lib, u ko`p tomondan Turbo Basic, Turbo Prolog, Turbo C va boshqa dasturlash vositalarining matn muharrirlariga o`xshab ketadi. Turbo Pascal dasturlash vositasining dasturiy ta'minoti quyidagilardan tashkil topgan:

Turbo.exe -asosiy dastur bo`lib tizimni ishga tushiradi;
Turbo.tpl -dasturlar bibliotekasi;
Turbo.hlp -yordam chaqirish fayli;
Grapf.tpu -grafik funktsiya va protseduralar bibliotekasi;
CGA.bgi, EgaVga.bgi -monitor uchun grafik drayverlari;
Tpc.exe -Pas kengaytmali faylni Exe kengaytmali fayliga o`tkazadi.

Turbo Pascal oynasi gorizontal menyu va dastur matnini kiritish maydonidan iborat. Gorizontal menyu quyidagilardan iborat:

File punkti quyidagi buyruqlar mavjud:
New - ekranni tozalash va operativ xotirada yangi fayl ochish;
Save -operativ xotiradagi dasturni asosiy xotiraga yozadi;
Save As -boshqa nom bilan dasturdan nusxa ko`chiradi;
Open -kerakli faylni ochishb ekranga chaqiradi;
Exit -tizimdan chiqish.
Edit -punkti dastur matnini qayta tahrirlashda ishlatiladi.
Run -punkti xotiradagi dasturni bajarishga beradi.
Comple -punkti dasturni kompilyatsiya qilib xotiraga yozadi.
Options -punkti tizimning parametrlarini o`zgartiradi.

Funktsional tugmachalar vazifalari:

F1 -yordam chaqirish; F2 -faylni diskga saqlash;
F3 -faylni ishga tushurish; F5 -ekranni ochish va yopish;
F6 -ekranni aktiv holatga o`tkazish; F9 -dasturni qadam bilan bajarish;
F10 -bosh menyuga o`tish.

Matn muharirida tugmalar yordamida buyruqlarni berish.

Tugmani bosish	Buyruqning bajarilishi	Tugmani bosish	Buyruqning bajarilishi
Ctrl →	o`nga bitta so`z o`tish	Ctrl N	qator qo`yish
Ctrl ←	chapga bitta so`z o`tish	Ctrl K B	blok boshini ajratish
Ctrl W	yuqoriga siljish	Ctrl K K	blok oxirini ajratish
Ctrl Z	pastga siljish	Ctrl K T	so`zni ajratish;
Home	qator boshiga o`tish	Ctrl K Y	ajratilgan blokni o`chirish
End	qator oxiriga o`tish	Ctrl K S	Blokni ko`chirish

Ctrl Home	birinchi qatorga o`tish	Ctrl U	buyruqni bekor qilish
Ctrl End	oxirgi qatorga o`tish	Ctrl F9	dasturni ishga tushurish
Del	o`ng yozuvni o`chirish	Ctrl C	dasturni ishdan to`xtatish
Ctrl Y	qatorni olib tashlash		

Savollar

1.Algoritm nima va u qanday xossalarga ega?
2.Algoritmning qanday turlari mavjud?
3.Algoritmni tasvirlashning qanday usullarini bilasiz?
4.Qanday algoritmik tillarni bilasiz?
5.Masalani kompyuterda yechish qanday bosqichlardan iborat?
6.Matematik model deganda nimani tushinasiz?
7.Qanday dasturlash vositalarini bilasiz?
7.Turbo Pascal dasturi matn muharririning qanday buyruqlarini bilasiz?

2.PASCAL ALGORITMIK TIL ELEMENTLARI

Pascal tili universal tillardan biri bo`lib, boshqa tillarga qaraganda imkoniyatlari kengroq tildir. So`ngi yillarda Pascal tili juda takomillashib, tobora ommalashib bormoqda. Pascal tili Shvetsariyalik olim N.Virt tomonidan yaratilib, keyinchalik Borland korporatsiyasi tamonidan rivjlantirildi.

2.1.Pascal algoritmik tilining alfaviti

Pascal tili ham boshqa dasturlash tillari kabi o`z alfavitiga va belgilariga ega. Pascal tili 26 ta lotin harflariin, 0 dan 9 gacha bo`lgan arab raqamlarini va quyidagi belgilarni ishlatadi: bo`shlik belgisi; 4 ta arifmetik amallar +, - , * , / ; mantiiiy amallarni bajarish uchun <, >, <=, >= , <>, = belgilarini ishlatadi. Ulardan tashqari vergul, nuqta, ikki nuqta, kichik, katta va o`rta qavsni ham ishlatadi.

1.2.Sonlar va o`zgarmaslar

Pascal tilida quyidagi turdagi sonlar ishlatiladi:

- haqiqiy turdagi sonlar;
- butun turdagi sonlar;
- belgililar (simvollar).

Haqiqiy turdagi sonlar umumiy holda quyidagi ko`rinishda bo`ladi:

s a1a2...an. b1b2...bk

Bu erda s ishora(+ yoki -) yoki bo`sh joy; a1a2...an butun qism; b1b2...bk kasr qism.
Masalan: +3,147 soni +3.147 yoki 3.147

-143,03 soni -143.03
57,0 soni 57.0
0,493 soni 0.493 yoki .493

Haqiqiy sonlarning o`zgarish diapazoni kompyuterning turiga karab turlicha bo`ladi.

$10^{-38}<X<10^{38}$ X - ixtiyoriy son. Ular eksponentsial (darajali) ko`rinishda ifodalanishi ham mumkin.

Masalan: $0,43\cdot10^{-6}$ kompyuterda .43E-6
0,0003 kompyuterda 3E-4

Butun sonlar umumiy holda quyidagicha yoziladi s a1a2...an.

Masalan: +345 soni +345 yoki 345
-106 soni -106

Butun sonlar o`zgarish diapozoni -32768 dan +32767 gacha. Agar butun son qiymati bu dipazondan chiqsa, u haqiqiy son shaklida ifodalanadi yoki kompyuter turiga qarab, u o`noltilik sanoq tizimida ifodalanishi ham mumkin. Belgili o`zgarmaslar diapazoni 0 dan 255 tagacha bo`ladi. Misol. `Pascal`, `405.5`

Pascal tilida identifikator so`zi ishlatilib ular o`zgarmaslarni, o`zgaruvchilarni, belgi(metka), protsedura va funktsiyalar nomlarini belgilashda ishlatilgan nomlvrdir. Identifikator lotin alfaviti harflaridan boshlanib qolganlari belgi yoki raqam ketma-ketligidan tashkil topgan bo`lishi mumkin. Masalan: xx, xx1, alfa&.

2.3.O`zgaruvchilar va o`zgarmaslarning dasturda tavsiflanishi

Pascal tilida dastur ishlash mobaynida qiymati o`zgarmaydigan identifikatorlar **o`zgarmaslar** deyiladi. Ular dasturning bosh qismida **Const** so`zi bilan e'lon qilinib unga aniq qiymat tenglashtiriladi.

Misol. Const aa1=2.27; Pi=3.14; Radius=14;

Dasturda qiymatlari o`zgarishi mumkin bo`lgan identifikatorga **o`zgaruvchilar** deyiladi. Ular dastur bosh qismida **Var** so`zi bilan e'lon qilinadi. O`zgaruvchilar nomi keltirilib, ularning turlari beriladi. O`zgaruvchilar turlari asosan butun, haqiqiy, belgili(simvolli), matnli va mantiqiy bo`ladi. Ular mos ravishda butun - **Integer**, haqiqiy - **Real**, belgili- **Char**, matnli(qatorli) - **String** va mantiqiy- **Boolean** deb yoziladi. Masalan: Var a, d1, alfa : Integer; c121, df : Real;
Etx, xx : Char; st: String; fl : Boolean;

Mantiqiy o`zgaruvchilar faqat ikkita qiymat qabul qiladi: "True" (chin) va "False"(yolg`on).

Ma'lumotlar turlarini umumiy holda ikkiga ajratish mumkin:

- standart turlar. Bu turlar Pascal tili tomonidan aniqlangan bo`ladi;
- dasturchi tomonidan kiritiladigan (aniqlanadigan) turlar.

Standart turlar tarkibiga quyidagilar kiradi: butun, haqiqiy, belgili (simvol), qator (strok), mantiqiy, ko`rsatgichli va variant.

Dasturchi turlarni dasturning **Var** bo`limida o`zgaruvchilarni tavsiflashda aniqlaydi yoki maxsus turlarni aniqlash uchun bo`lim bo`lgan turlarni tavsiflash **Type** bo`limida aniqlaydi. Bu bo`lim umumiy holda quyidagicha bo`ladi.

Type

<tur nomi>=<turning tavsifi>;

Misol: Type

TColor=(Red,Blue,Black);

Var Color1,Color2,Color3: TColor;

Type bo`limida dasturchi tomonidan yangi Tcolor nomli tur kiritilmoqda va u Red, Blue, Black mumkin bo`lgan qiymatlarni qabul qilishi mumkin.

Var bo`limida dasturchi tomonidan turi aniqlangan uchta Color1, Color2, Color3 o`zgaruvchilar tavsiflanmoqda.

Bu o`zgaruvchilarni to`g`ridan to`g`ri quyidagicha ham tavsiflash mumkin.

Var Color1,Color2,Color3: (Red,Blue,Black);

Standart turlarni Type bo`limida tavsiflash shart emas, ularni to`g`ridan to`g`ri Var bo`limida tavsiflash mumkin.

2.4.Standart va nostandart matematik funktsiyalar

Standart matematik funktsiyalar.

Funktsiya nomi	Tilda yozilishi	Funktsiya nomi	Tilda yozilishi
Sinx	SIN(x)	$\sqrt{x}$	SQRT(x)
Cosx	COS(x)	Arctgx	ARCTAN(x)
Lnx	Ln(X)	|x|	ABS(x)
e^x	EXP(x)	x^2	SQR(x)
		a^b	EXP(b*LN(a))

Nostandart matematik funktsiyalar.

$1.\ Secx = \frac{1}{Cosx};\quad 2.\ Co\sec x = \frac{1}{Sinx};\quad 3.\ Tgx = \frac{Sinx}{Cosx};\quad 4.\ Arcctgx = Arctg\ \frac{1}{x};$

$5.\ Arc\sin x = Arctg\ \frac{x}{\sqrt{1-x^2}};\quad 6.\ Arc\cos x = Arctg\ \frac{\sqrt{1-x^2}}{x};\quad 7.\ Arc\sec x = Arctg\ \frac{1}{\sqrt{1-x^2}};$

$8.\ Arc\cos ecx = Arctg\sqrt{1-x^2};\quad 9.\ Log_a b = \frac{Lnb}{Lna};\quad 10.\ Padian = \frac{Gradius \cdot \pi}{180}$

O`zgartirish funktsiyalari.

Funktsiya	Qiymati
Chr(n)	Kodi n ga teng simvol
IntToStr(k)	Butun k ni tasvirlovchi satr
FloatToStr (n)	Haqiqiy n sonini tasvirlovchi satr
StrToInt (s)	Satrni butun songa o`tkazish
StrToFloat (s)	Satrni haqiqiy songa o`tkazish
Round (n)	Haqiqiy sonni yaxlitlash
Trunc (n)	Haqiqiy son kasr qismini olib tashlash
Frac(n)	Kasrli sonning kasr qismi
Int (n)	Kasr sonning butun qismi

2.5.Arifmetik va mantiqiy ifodalar

Ifodalarda hisoblashlar tartibi qavslar ichidagi ifodalar bajarilgandan keyin quyidagi amallar bajariladi:

1.NOT amali;

2. *, /, DIV, MOD, AND;

3.+, -, OR;

4.taqqoslash belgilari: <, >, <=, >=, <>, =, IN.

Ifodadagi amal natijasi qanday turda bo`lishi amallarda qatnashayotgan o`zgaruvchilarning turlariga bog`liq. Agar ikkita o`zgaruchining turi Integer yoki Real bo`lsi, amal natijasi ham Integer yoki Real bo`ladi. Agar biri Integer ikkinchisi Real bo`lsa natija Real bo`ladi. NOT, OR, AND va taqqoslash amallarining natijalari esa Boolean turida bo`ladi.

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

Quyidagi mantiqiy amallar ishlatiladi: NOT-«inkor»; AND-«mantiqiy va»; OR-«mantiqiy yoki».

Bu mantiqiy amallarning bajarilish natijalari quyidagicha:

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

Masalan: (5<6) AND (6<50) -mantiqiy ifoda rost (True),
(20>0) OR (20<0.5) -mantiqiy ifoda rost (True),
NOT(100>3) -mantiqiy ifoda yolg`on (False).

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

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

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

2.6.Pascal tilida dastur strukturasi

Pascal tilida dastur matni bosh va asosiy bo`limdan tashkil topadi. Bosh bo`lim dastur nomi va o`zgaruvchilar, o`zgarmaslar, massivlar, belgilar(metkalar), protseduralar va funktsiyalarni tavsiflashdan iborat bo`ladi. Asosiy bo`lim dastur tanasi deyilib, unda dasturda bajariladigan hamma operatorlar ketma-ketligi beriladi va u Begin (boshlamoq) so`zi bilan boshlanib End (tugash) so`zi bilan tugaydi. Umumiy holda dastur strukturasi quyidagi ko`rinishga ega:

Program <dastur nomi>;
Uses <Foydalanadigan bibliotekalar (modullar) ro`yxati>;
Label <Ishlatiladigan belgilar(metkalar) ro`yxati>;
Const <Ishlatiladigan o`zgarmaslarni aniqlash>;

```
Type   <YAngi turlarni aniqlash>;
Var    <O`zgaruvchilarni e'lon kilish>;
       <Protsedura va funktsiyalarni aniqlash>
Begin
    <Asosiy qism >
End.
```

2.7.Ma'lumotlarni kiritish va chiqarish operatorlari

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

Kompyuter foydalanuvchi tomonidan qo`yilgan masalani aniq va tushunarli ko`rsatmalar berilgandagina bajara oladi. Bu ko`rsatmalar ma'lum bir ma'noni anglatuvchi so`zlardan iborat bo`lib, kompyuterga qanday operatsiyani bajarish lozimligini bildiradi va bu ko`rsatmalarga **operatorlar** deyiladi. Operatorlar dastur ishlaganda ketma-ket ravishda bajariladi. Pascal tilida bir satrga bir necha operatorlarni yozish umkin. Dasturdagi o`zgaruvchilar qiymatlarini dastur ichida o`zlashtirish operatori yordamida ham berish mumkin. Lekin dasturda o`zgaruvchi qiymatini tashqaridan kiritish qulaylik tug`diradi va umumiylikni ta'minlaydi.

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

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

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

Misollar: Read(Sm1,Sm2); Readln(x1,x2,x3); Readln;

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

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

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

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

Misollar: Write(Summa); Write('Natija yuk');

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

Oddiy ma'lumotlarni chiqarishda ular matn deb qaraladi va u qo`shtirnoq ichida yoziladi. CHiqarish operatori yordamida o`zgaruvchilar qiymatini format ko`rinishda ham berish mumkin: Write(c:m:n);

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

Misol. Write(x:8:4);

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

Write(‘Maxsulot soni:’, kol:5);

Agar kol=15 bo`lsa, quyidagi yozuv ekranga chiqadi. Maxsulot soni: 15

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

Masalan: { Bu matn dasturga izoh beradi } { Bu joyda echim aniqlanmoqda }

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

Masalan: i:=0; i-qiymati nolga tenglashadi, ya’ni i o`zgaruvchiga nol yuboriladi deb tushuniladi. Bunda mashina i o`zgaruvchi uchun ajratilgan xotirasiga nol yozib saqlaydi. Misol: B:=5; C:=4; A:=(B+C)/2;

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

O`zlashtirish operatori yordamida ixtiyoriy ifodalarni hisoblash va natijalarni xotiraga joylashtirish mumkin.

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

Misol: Tekislikdagi ikki nuqta orasidagi masofani topish dasturi.

```
Program XY;
 Var   x1,y1,x2,y2,d: Real;
   Begin
      Write(‘Nukta koordinatalarini kiriting:’);  Readln(x1,y1,x2,y2);
         d:=Sqrt(Sqr(x1-x2)+Sqr(y1-y2));
         Writeln(‘Masofa=’,d);
         Readln;
    End.
```

Savollar

1.Arifmetik va mantiqiy ifodalarda qanday belgilar ishlatiladi?
2.Dasturda ishlatiladigan sonlarning o`zgarish diapazoni qanday?
3.O`zgarmas va o`zgaruvchilar dasturda qanday e’lon qilinadi?
4.O`zgaruvchilarning qanday turlari mavjud va identifikator nima?
6.Mantiqiy o`zgaruvchilar qanday qiymat qabul qiladi?
7.150 div 25 mod 5 va (2.7)+round(3.4) ifoda qiymatlarini aniqlang.
8.Ma’lumotlarni chiqarish operatorini tushuntirib bering.
9.Ma’lumotlarni chiqarish operatori nima uchun kerak?
10.Ma’lumotlarni kiritish operatorini tushuntirib bering.
11.Ma’lumotlarni kiritish operatori nima maqsadda ishlatiladi?
12.Izoh va o`zlashtirish operatori nima uchun ishlatiladi?

3.SHARTLI VA SHARTSIZ O`TISH OPERATORLARI

SHunday masalalar borki ularning algoritmi to`g`ridan to`g`ri ketma-ket ravishda bajarilmaydi. Bunda shartning berilishiga qarab u yoki boshqa hisoblashlar bajariladi. Bunday hisoblash jarayonlarni dasturlash Pascal tilida shartli va shartsiz o`tish operatorlari yordamida amalga oshiriladi. Tarmoqlanuvchi hisoblash jarayoni tarkibida yana tarmoilanishlar bo`lishi mumkin. Bunday jarayonlarga murakkab tarmoqlanuvchi hisoblash jarayonlari deyiladi.

3.1.SHartli o`tish operatori

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

To`liq ko`rinish:

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

Qisqa ko`rinish:

```
If <shart> then Begin
        <shart rost bo`lganda bajariladigan operatorlar>
            End;
```

Bu erda IF -agar; then -u holda; else -aks holda ma’nosini bildiruvchi xizmatchi (kalit) so`zlar.

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

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

Agar bajariluvchi operatorlar soni bitta bo`lsa Begin va End so`zlarini yozish shart emas.

Misollar:

1) If A>0 Then Begin C:=1; B:=C+1; End
   Else Begin C:=0; B:=4; End;
2) If D=A Then D:=A Else A:=D;

Har bir shartli o`tish operatori ichida boshqa ichki shartli operatorlar joylashishi ham mumkin, masalan If b1 then a1 else If b2 then a2 Else a3;

Misollar.

A:=0.5; B:=-1.7; IF A<B THEN A:=B ELSE B:=A;

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

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

IF (A<B) OR (A>C) THEN D:=B+C ELSE

IF B=A THEN BEGIN D:=C; C:=A; END;

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

3.2.SHartsiz o`tish va tanlash operatorlari

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

SHartsiz o`tish operatorining ko`rinishi quyidagicha:

Goto n;

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

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

Misol:

```
  .....
  Goto L2;
   .....
L2: C:=x*y;
   .....
```

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

```
Case s of
  1: A1;
  2: A2;
  . . . . . .
  n: An;
       Else Begin
            <B1,B2,..Bn>
            End;
 End;
```

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

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

If B Then A1 Else A2;

tanlash operatorining quyidagi operatoriga ekvivalentdir.

```
Case B of
        True: A1;
        False: A2;
    End;
```

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

```
Program kvt;
 Var a,b,c,x1,x2,d:Real;
 label L1;
 Begin
   Writeln('Kvadrat tenglama koeff.kiriting:');
    Write('a='); Readln(a); Write('b='); Readln(b); Write('c='); Readln(c);
    If a=0 Then begin x1:=c/b; Write('Javob bitta x=',x1); Goto L1; end;
    d:=b*b-4*a*c;
    If d<0 Then Write ('Haqiqiy echim yo`q')
         Else  begin x1:=(-b+Sqrt(d))/2/a; x2:=(-b-Sqrt(d))/2/a;
                  Writeln('x1=',x1); Write('x2=',x2); Readln;
               End;
    L1:
 End.
```

Savollar

1.Tarmoqlanuvchi algoritm qaysi hollarda mavjud bo`ladi?
2.Murakkab tarmoqlanuvchi jarayon nima?
3.SHartli va shartsiz o`tish operatorini tushuntiring?
4.Tanlash operatorini tushuntiring?
5.Quyidagi dastur bo`lagi natijasini aniqlang?

```
X:=0;  Y:=1;
If X>Y  then  begin  X: =X-10 ;  Y:=Y+10  end;
Writeln ('X=',X,'; Y=',Y);
```

4.TSIKL OPERATORLARI

Ko`plab shunday masalalar borki parametrlarning o`zgarishiga qarab ma'lum hisoblashlar bir necha marta takrorlanib bajarilishi mumkin. Masalan, biror bir funksiyani nom'alum x ning bir necha qiymatida uning mos qiymatlarini hisoblash kerak deylik. Bunday hisoblashlarni kompyuterda dastur tuzib bajarish uchun tsiklik dasturlar tuzish kerak bo`ladi. Bu kabi dasturlarni shartli operatorlar yordamida tuzsa ham bo`ladi. Lekin Pascal tilida tsiklik strukturali dastur tuzish uchun bir necha maxsus operatorlar mavjud. Ular For, While va Repeat operatorlaridir.

4.1.FOR tsikl operatori

For operatori takrorlanishlar soni aniq bo`lgan tsikllik jarayonlar tashkil etishda ishlatiladi. Uning umumiy ko`rinishi quyidagicha:

For i:=m1 to m2 Do S;

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

Agar tsikl tanasi bir necha operatordan iborat bo`lsa ular **Begin** va **End** ichiga olinadi.

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

```
Program S10;
    Const kn=10;
    Var  i: Integer; S: Real;
    Begin
      S:=0;
      For i:=1 to kn do S:=S+i;
      Write ('S=',S);   Readln;
    End.
```

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

For i:=m1 DoWnto m2 Do S;

Misol. 10 dan 1 gacha conlarni ekranga chiqarish dastursini tuzing.

```
Program SP;
    Var i: Integer;
    Begin
      For i:=10 DoWnto 1 do Write (i);
    End.
```

4.2.WHILE tsikl operatori

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

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

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

Misol. 1,2,...,10 conlar yiQindisini hisoblash dastursini tuzing.

```
Program S10;
      Const  n=10;
      Var  i: Integer; S: Real;
      Begin
        S:=0;  i:=0;
        While i<n do Begin i:=i+1; S:=S+i; End;
        Write (‘S=’,S);
      End.
```

4.3.REPEAT tsikl operatori

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

Repeat
S
Until B

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

Agar mantiqiy ifoda ‘False’ qiymat qabul qilsa tsiklda takrorlanish davom etadi, aks holda to`xtaydi.

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

```
Program S10;
      Const  n=10;   Var  i: Integer; S: Real;
      Begin
        S:=0;  i:=0;
        Repeat  i:=i+1; S:=S+i;  Until i>=n;
         Write (‘S=’,S);
```

End.

Odatda WHILE operatori REPEAT operatoriga nisbatan ko`p ishlatiladi. Bunga sabab ko`pchilik masalalarda tsikl tugallanish sharti tsikl boshlanmasdan tekshirish maqsadga muvofiqdar.

4.4.Murakkab tsikllar

Ko`pchilik masalalarni yechishda tuzilgan dasturda ichma-ich joylashgan tsikllar tashkil etishga to`g`ri keladi. Bunday tsikllarga murakkab tsikllar deyiladi. Murakkab tsikllarda quyidagi talablar bajarilishi zarur.

- ichki tsikl tashqi tsikl ichida to`liq yotishi kerak;
- tsikllar bir-biri bilan kesishmasligi kerak;
- tsikl ichiga tashqaridan to`g`ridan-to`g`ri kirish mumkin emas;
- tsikl parametrlari har xil identifikatorlar bilan belgilanishi kerak;

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

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

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

```
Program SP;
    Var  i,j: Integer; S,P: Real;
    Begin
      S:=0;
      For i:=1 to 10 do
        Begin
            P:=1;          For j:=1 to 5 do P:=P*(i+j)/Sqrt(i*j);
             S:=S+P;
       End;  Write ('S=',S);
    End.
```

Savollar

1.TSikl operatori nima uchun kerak bo`ladi?
2.For operatori qanday hollarda ishlatiladi?
3.While va Repeat operatorlarining ishlashini tushuntirib bering.
4.Murakkab tsikllarga qanday talablar qo`yiladi?
5.Quyidagi dastur natijasini aniqlang.

```
    S: = 0;
For  I: =1  to 10  do   Begin  S: = S + 1/SQR(I);  Writeln ('I=',I,'; S=',S)   end;
```

5.MASSIVLAR

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

5.1.Massivlar va ular ustida ish yuritish.

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

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

Har bir massiv o`z o`lchamiga ega bo`lib va u dasturda e'lon qilingan bo`lishi kerak. Massivni e'lon qilish dasturning bosh qismida berilib, uning yozilishi umumiy holda quyidagicha bo`ladi:

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

Masalan:

A,B:Array[1..100] of real;

C,A1,D:Array[1..10,1,,15] of real;

Bu erda A va B massivlari 100tadan elementga ega. C,A1,D1 massivlari esa 10x15=150 tadan elementga ega.

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

Massiv elementlari qiymatlarini kiritish uchun tsikl operatorlaridan foydalaniladi. Misol: For i:=1 to 10 do Read(A[i]);

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

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

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

5.2.Massiv elementlarini tartiblash usullari.

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

-tanlash usuli;

-almashtirish usuli.

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

1.Massivning birinchi elementidan boshlab qarab chiqilib eng kichik element topiladi.

2.Birinchi element bilan eng kichik element joylari almashtiriladi.

3.Ikkinchi elementidan boshlab qarab chiqilib eng kichik element topiladi.

4.Ikkinchi element bilan eng kichik element joylari almashtiriladi.

5.Bu protsess bitta oxirgi elementgacha takrorlanadi.

Bu algoritm dasturi quyidagicha bo`ladi:

```
Program Sort;
  Const  n=5;
  Var i, j, min, k, buf: Integer;   a: Array[1..n] of Integer;
  Begin
     Writeln (‘Massivni tartiblashtirish’);
     Write (n:3,’ -ta massiv elementini kiriting`);
     For k:=1 to n Do Read(a[k]);
     For i:=1 to n-1 Do
           Begin           { kichik elementni topish }
             min:=i;
             For j:=i+1 to n Do
               Begin
                  If a[j]<a[min] then min:=j;
                  buf:=a[i];   a[i]:=a[min];  a[min]:=buf;
                  For k:=1 to n Do  Write (a[k],’ ‘);
                  Writeln;
               End;
           End;
       Writeln(`Massiv tartiblashtirildi.`);
   End.
```

Dastur natijasi:

```
Massivni tartiblashtirish
 5 ta massiv elementini kiriting
 12 -3 56 47 10
Tartiblatirish
 -3 12 56 47 10
 -3 10 56 47 12
 -3 10 12 47 56
 -3 10 12 47 56
```

Massiv tartiblashtirildi.

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

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

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

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

Masalan:

3 2 4 5 1 bunda 3 bilan 2 va 5 bilan 1 almashtiriladi.
2 3 4 1 5 bunda 4 bilan 1 almashtiriladi.
2 3 1 4 5 bunda 3 bilan 1 almashtiriladi.
2 1 3 4 5 bunda 2 bilan 1 almashtiriladi.
1 2 3 4 5
Bu algoritm dasturi quyidagicha bo`ladi:

```
Program Sort;
    Const  n=5;
    Var i,j,min,k,buf: Integer;  a: Array[1..n] of Integer;
  Begin
    Writeln (‘Massivni puzirek(kupikcha) usulida tartiblashtirish’);
    Write (Size:3,’ta massiv elementini kiriting`);
    For k:=1 to n Do Read(a[k]);
    Writeln (‘Tartiblatirish’);
     For i:=1 to n-1 Do
          Begin
             For k:=1 to n-1 Do
              Begin
                 If a[k]>a[k+1] then
                   Begin
                        buf:=a[k];  a[k]:=a[k+1];  a[k+1]:=buf;
                  End;
                End;
          For k:=1 to n Do  Write (a[k],’ ‘);  Writeln;
       End;
      Writeln(‘Massiv tartiblashtirildi.’);
  End.
```

Dastur natijasi:
Massivni puzirek usulida tartiblashtirish

```
 5 ta massiv elementini kiriting
 3  2  4  1  5
Tartiblashtirish
 2  3  4  1  5
 2  3  1  4  5
 2  1  3  4  5
 1  2  3  4  5
Massiv tartiblashtirildi.
```

Massivda eng kichik yoki eng katta elementni izlash algoritmi ma'lumki birinchi element eng kichik (katta) deb olinib keyin boshka elementlar bilan ketma-ket solishtirilib chiqiladi. Solishtirilish oxirgi elementgacha bajariladi. Quyida bu algoritm dasturi keltirilgan:

```
Program MinMax;
  Var i,min: Integer;     a: Array[1..10] of Integer;
   Begin
       Writeln ('Massivdan eng kichik elementni izlash');
       Write (' 10-ta massiv elementini kiriting`);
       For i:=1 to 10 Do Read(a[i]);
         min:=1;
        For i:=2 to 10 Do
            If a[i]<a[min] Then min:=i;
            Writeln('Izlanayotgan eng kichik element:',a[min]);
              Writeln('Element nomeri',min);
    End.
```

Savollar

1.Massiv nima va qanday o`lchamli massivlar bor?
2.Massivlar dasturda qanday e'lon qilinadi?
3.Massivni tartiblashtirishning qanday usullarini bilasiz?
4.Puzirka usulini tushuntirib bering.
5.Massivdan kerakli ma'lumotni izlash algoritmini tushuntirib bering.
6.Massivdan eng kichik elementni topish algoritmini tushuntiring.

6.QISM DASTURLAR

Dasturlash jarayonida shunday holatlar bo`ladiki, bir xil operatorlar ketma-ketligini dasturning bir necha joylarida takroran yozishga to`g`ri keladi. Bunday takrorlanishni yo`qotish maqsadida dasturlashning ko`pgina tillarida qism dastur tushunchasi kiritilgan. Ular mustaqil dastur bo`lagi sifatida dasturning bosh qismida bir marotaba yoziladi.

Pascalь tilida qism dastur protsedura yoki funktsiya ko`rinishida beriladi. Asosiy dastur bilan protsedura orasida o`zgaruvchilar qiymat almashuvi formal va faktik parametrlar yordamida amalga oshiriladi. Protsedura ichida yana bir necha protsedura yoki funktsiya ishlatilishi mumkin. Dasturda e'lon qilingan o`zgaruvchilar, shu dasturdagi protsedura va funktsiyalarga nisbatan global deyiladi. Protsedura va funktsilar ichida e'lon qilingan o`zgaruvchilar lokal deyiladi. Ularning ta'sir doirasi shu qism dastur ichida bo`ladi.

6.1.Protseduralar

Protseduralarni e'lon qilish quyidagicha bo`ladi.

Procedure <prots.nomi> (<formal parametrlar>);
<o`zgaruvchi va o`zgarmaslarni tavsiflash qismi>
Begin
<asosiy qism>
End;

Formal parametrlarni shu protsedura bosh qismida yoki sarlavhada e'lon qilish mumkin. Masalan. **Procedure AB (x,y: Real);**

Har qanday protsedurani kichik bir dastur deb qarash mumkin. Protsedura ham dasturga o`xshab bosh va asosiy qismlardan toshkil topadi. Bosh qismda protsedura nomi va uning parametrlari e'lon qilinadi. Asosiy qism operatorlar ketma-ketligidan tashkil topgan bo`lib, ular Begin - End ichiga olinadi. Protsedura nomi foydalanuvchi tamonidan beriladi.

Misol. Procedure Dr(Var x,h1,h2,z1,z2 : Real);
Var h,z: Real;
Begin
h:=h1/z1+h2/z2; z:=z1/z2; x:=(h+z)/2;
End;

Bu protsedurada h1,z1,h2,z2 parametrlar qiymati protseduraga murojat qilinganda aniqlangan bo`lishi kerak. Natijani esa x- parametr uzatadi. h va z o`zgaruvchilar ichki o`zgaruvchilardir. Bu protseduraga dasturdan quyidagicha murojaat qilinadi Dr(x,h1,h2,z1,z2). Protseduraga murojaat qilinganda mos parametrlar qiymati bir biriga uzatiladi. Beriladigan formal va faktik parametrlar soni teng va ular turlari bir xil bo`lishi shart. Lekin parametrlar nomlari har xil bo`lishi mumkin.

6.2.Funktsiyalar

Funktsiyalardan foydalanish va ularni tashkil qilish xuddi protsedura kabi bo`lib, u quyidagicha bo`ladi:

Function <f-ya nomi>(<formal parametrlar>):<f-ya turi>;
<o`zgaruvchi va o`zgarmaslarni tavsiflash qismi>
Begin
<asosiy qism>
End;

Funktsiyaning protseduradan farqi, unga murojat qilinganda natija faqat bitta bo`lib, u shu funktsiya nomiga uzatiladi.

Misol 1. Quyidagi hisoblashni funktsiyani ishlatgan holda dasturini tuzing.

$$C_n^m = \frac{n!}{m!(n-m)!}.$$

```
Program Kol;
  Var ncm:Real;  n,m,i: Integer;
    Function Fact (k: Integer): Integer;
      Var P,i: Integer;
       Begin
            P:=1;    For i:=1 to k do P:=P*i;
            Fact:=P;
      End;
    Begin
         Read(n,m); l:=n-m;  ncm:=Fact(n)/Fact(m)/Fact(i);
         Write('ncm=',ncm);
    End.
```

Misol 2. Quyidagi hisoblashni protsedurani ishlatgan holda dastursini tuzing.

$$z = \frac{tha - th^2(a-b)}{\sqrt{th(a^2-b^2)}}, \qquad thx = \frac{e^{2x}-1}{e^{2x}+1}.$$

```
Program Fun1;
  Var a,b,z,c,d,t1,t2,t3: Real;
    Procedure Th(Var x,r: Real);
      Var c: Real;
       Begin      c:=exp(2.0*x);    r:=(c-1)/(c+1);       End;
    Begin
         Read(a,b);
          th(a,t1);   c:=a-b;    th(c,t2);
         d:=Sqr(a)-Sqr(b);   th(d,t3);
         z:=(t1+Sqr(t2))/Sqrt(t3);
         Write('z=',z:10:3);
    End.
```

6.3.Ayrim nostandart protsedura va funktsiyalar

Pascalь tilida bir qancha maxsus protsedura va funktsiyalar mavjud bo`lib, ular quyidagi guruhlarga bo`linadi:

- qatorni qayta ishlash;
- fayllar bilan ishlash;
- dinamik o`zgaruvchilar uchun xotirani boshqarish;
- arifmetik funktsiyalar;
- ekran bilan ishlash.

Ularning ayrimlarini ko`rib chiqamiz:

Halt- dasturni bajarishdan to`xtatish;

Odd(i)- I-toq bo`lsa “True” aks holda “False” qiymat oladi;

Edit- bajarilayotgan blokdan chiqish;

Random- 0 dan 1 gacha bo`lgan sonni tasoddifan olish;

GotoXY(x,y)- kursorni ko`rsatilgan joyga qo`yish;

ClrScr- ekranni tozalab, kursorni ekran boshiga qo`yish;

Trunc- argumentning butun qismi:

Str(I;Var S:String)- raqamni simvolga o`tkazish (I-ifoda yoki o`zgaruvchi);

Val(S:String; Var P;ko:Integer)- simvolni raqamga o`tkazish (P-o`zgaruvchi);

Length (S:String)- qator uzunligini aniqlash.

Pos(st1,st) - qatordagi qator qismi holatini aniqlash.

Misol: st:=’Toshkent’; st1:=’kent’; p:=pos(st1,st); Javob: p=4. Agar izlanayotgan qator qism qatorda yo`q bo`lsa qiymat nulga teng bo`ladi.

Copy(st,m,n) - qatordan fragment kesib oladi.

Misol: st:=’Toshkent’; p:=Copy(st,4,4); Javob: p=’kent’.

Delete(st,m,n) - qatordan fragment kesib olib tashlaydi.

Misol: st:=’Toshkent’; p:=Delete(st,4,4); Javob: p=’Tosh’.

Savollar

1.Protsedura va funktsiyalar dasturda qachon ishlatiladi?
2.Protsedura nima va u dasturda qanday e’lon qilinadi?
3.Funkqiya nima va u daqurda qanday e’lon qilinadi?
4.Funktsiya va protsedura nima bilan farqlanadi?
5.Qanday maxsus prottsuduralar va funktsiyalarni bilasiz?

7.BELGILAR VA QATORLAR

SHunday masalalar borki ularni matnlar ustida ish bajibgina hal etish mumkin. Masalan, fayldagi matnlar ichidan biror bir ism yoki familiyani izlash kerak yoki bir xil ma'lumotni boshqa bir ma'lumotga almashtirish kerak yoki ular ustida biror bir operatsiyalarni bajarish kerak bo`lsin deylik. Dasturlashda bunday masalalarni yechishda belgilar va qatorlardan foydalanishga to`g`ri keladi. Dasturlashda ularni ishlatishda belgi va qatorli tipdagi maxsus o`zgaruvchilar yoki o`zgarmaslardan foydalaniladi.

7.1.Belgilar va ularni qayta ishlash

Maxsus belgilarni qayta ishlash va saqlash uchun CHAR tipdagi o`zgaruvchilar ishlatiladi. CHAR tipdagi o`zgaruvchilar qiymati istalgan belgi bo`lishi mumkin. Belgili tipdagi o`zgaruvchilar dasturning bosh qismida o`zgaruvchilarni e'lon qilish bo`limida quyidagicha ko`rsatilishi kerak:

```
Var
   nom: CHAR;
```

bu erda *nom* – belgili tipdagi o`zgaruvchi nomi.
Misollar:

```
Var
    Otv: Char;
    Ch, Cr: Char;
```

Boshqa o`zgaruvchilar kabi belgili tipdpgi o`zgaruvchilar ham kiritish (READ) yoki yuborish (qiymat berish) operatorlari yordamida qiymatga ega bo`lishi mumkin. Agar CHAR tipdagi o`zgaruvchilar qiymat berish operatsiyasining natijasida qiymat qabul qilishi kerak bo`lsa, u holda yuborish operatsiyasining o`ng tomonida CHAR tipdagi ifoda bo`lishi yoki belgili (simvolli) o`zgaruvchi yoki belgili o`zgarmas bo`lishi kerak. Simvolli o`zgarmas ochiluvchi va yopiluvchi kavichka ichida yoziladi.
Masalan, quyidagi dastur natijasida o`zgaruvchi c1 o`zgarmas qiymat, c2 o`zgaruvchi c1 qiymatini qabul qiladi, keyin esa otvet o`zgaruvchi qiymati klaviaturadan kiritiladi. Bunda otvet o`zgaruvchi qiymati sifatida bitta belgi berilishi kerak bo`ladi. Masalan "Da" belgisi kiritilsa otvet o`zgaruvchisi "D" qiymatni qabul qiladi.

```
Var
    c1,c2,otvet: Char;
 Begin
    c1:=`*`;
    c2:=c1;
    Write (`Siz dastur tuzishni o`rganishni xohlaysizmi?`);
```

```
    Readln(otvet);
    Writeln (`Sizning javobingaz:`, otvet);
  End.
```

CHAR tipdagi o`zgaruvchini boshqa bir CHAR tipidagi o`zgaruvchi bilan yoki belgili o`zgarmas bilan taqqoslash mumkin. Taqqoslash shunga asoslanganki, har bir belgiga mos son qo`yilgan.

`0`<`1`<…<`9`<…<`A`<`B`<…<`Z`<`a`<`b`<…<`z`

Xuddi shunday kril alfavitida ham har bir belgiga mos son berilgan:

`A`<`B`<…<`YU`<`YA`<`a`<`b`<…<`ya`

Belgili o`zgaruvchi va o`zgarmaslarni shartli operatorlarda ham ishlatish mumkin. Masalan, quyidagi dasturda otvet o`zgaruvchisi dastur so`roviga javob berishni tekshirishda ishlatilmoqda.

```
Var
    otvet: Char;
  Begin
    ….
    Writeln (`Hisoblsh davom ettirilsinmi? (d/n)`);
    Readln(otvet);
    If (otvet=`d`) or (otvet=`D`) Then begin
        …
    End;
  End.
```

Har bir belgi son bilan kodlashtirilgan. Agar dasturda klaviaturada yo`q belgini ekrandan kiritish kerak bo`lsa, u holda CHR funktsiyasidan foydalaniladi. Masalan, Ch:=CHR(218); operatorining bajarilishi natijasida Ch belgili o`zgaruvchi `┌` belgisini o`zlashtiradi.

Quyidagi dastur 0 dan 128 gacha bo`lgan belgilar kodirovkasi jadvalini ekranga chiqaradi. Bu kodlardan 7 sidan boshlab 13 gachasi xizmatchi, masalan 7 kodi bilan kiritilgagn belgi tovush signalini beradi, 13 kodi esa yangi qatorga o`tish buyrug`ini chaqiradi.

```
Var
    ch: Char;          {belgi}
    Dec: Integer;      {belgining o`nlikdagi kodi}
    i, j: Integer;
  Begin
    Dec:=0;
    For  i:=0 to 16 do
     Begin
       Dec:=i;
```

```
            For j:=1 to 8 do {sakkizta qator }
               Begin
                  If (Dec<7) Or (Dec>=14) Then
                           Write (Dec:4,'-',' ',Chr(Dec):1,Chr(179))
                          Else   {7 dan 14 gacha belgidan chiqmaydi}
                            Write (Dec:4,'-  ',Chr(179));
                  Dec:=Dec+16;
               End;
            Writeln;    {yangi qatorga o`tish}
      End;
  End.
```

Dastur ishlashi natijasi quyidagicha:

0 - !	16 -	32 -	48 – 0	64 -	80 – P	96 – ‘	112 – p
1 - ^	17- □	33- !	49- 1	65- A	81- Q	97- a	113- q
2 - ε	18- □	34- "	50- 2	66- B	82- R	98- b	114- r
3 - ə	19- □	35- #	51- 3	67- C	83- S	99- c	115- s
4 - ∂	20- □	36- $	52- 4	68- D	84- T	100- d	116- t
5 - ♣	21- □	37- %	53- 5	69- E	85- U	101- e	117- u
6 - ♠	22- □	38- &	54- 6	70- F	86- V	102- f	118- v
7 -	23- □	39- '	55- 7	71- G	87- W	103- g	119- w
8 -	24- □	40- (	56- 8	72- H	88- X	104- h	120- x
9 -	25- □	41-)	57- 9	73- I	89- Y	105- i	121- y
10-	26- □	42- *	58- :	74- J	90- Z	106- j	122- z
11-	27- □	43- +	59- ;	75- K	91- [	107- k	123- {
12-	28- □	44- ,	60- <	76- L	92- \	108- l	124- |
13-	29- □	45- -	61- =	77- M	93-]	109- m	125- }
14- ∃	30- -	46- .	62- >	78- N	94- ^	110- n	126- ~
15- *	31-	47- /	63- ?	79- O	95- _	111- o	127- □

Quyidagi dastur matnlarni chiqarishni rasmiylatirish, ya’ni bezashda CHR funktsiyasini ishlatadi.

```
Var
      t1, tr, b1, br, g, v: Char;
      i: Integer;
  Begin
      t1:=Chr(218);            { yuqori chap burchak belgisi }
      tr:=Chr(191);            { yuqori o`ng burchak belgisi }
      b1:=Chr(192);            { pastki chap burchak belgisi }
      br:=Chr(217);            { pastki o`ng burchak belgisi }
```

```
    g:=Chr(196);              { gorizontal to`g`ri chiziq }
    v:=Chr(179);              { vertikal chiziq }
    Write (t1);
    For i:=1 to 32 do Write(g);
    Writeln(tr);
    Writeln (v, '      Ramka chizish uchun CHR       ', v);
    Writeln (v, 'funktsiyasini ishlatishga misol', v);
    Write (b1);
    For i:=1 to 32 do Write(g);
    Writeln (br);
End.
```

Dastur ishlashi natijasi quyidagicha:

Ramka chizish uchun CHR funkциyasini ishlatishga misol

Tovush signalini 7 kodi bilan ekran orqali kiritishni, masalan dasturda ma'lumotni kiritishda agar xatolik yuz berganda yoki nato`g`ri ma'lumot kiritilganda ishlatish mumkin. Quyidagi dastur parolni tekshiradi, agar parol nato`g`ri bo`lsa kompyuter dinamikasi tovush signalini beradi.

```
Label
      Bye;
Var
      Password: Integer;
     Begin
       Write ('Parolь');   Readln (Password);
        If Password<>377 then
                                Begin Writeln (Chr(7), 'Parolь nato`g`ri!');
                                      Goto Bye;
                                End;
           Bye:
     End.
```

7.2.Belgili massivlar

Ketma-ketlikdagi belgilarni qayta ishlash va saqlash uchun belgili massivlarni ishlatish mumkin. Masalan, dasturda klaviaturadan familiyalarni kiritish kerak bo`lsin. Odatda bunday masalani yechishda belgili massivga ketma-ketlikdagi belgilarni kiritish bilan amalga oshiriladi. Bunda kiritilgan belgilar qatori familiyadagi harflarning soniga teng olinadi. Agar Buf belgili massiv SIZE

o`lchamga ega bo`lsa, u holda eng sodda usulda massiv elementlarini FOR operatori yordamida kiritish mumkin:

```
For i:=1 to SIZE do Read (Buf[i]);
```

Dasturda belgili massivni kiritishda ancha effektiv bo`lgan EOLN (End Of Line) qurilgan funktsiyasidan foydalanish mumkin. Bu funktsiya agar massivga belgi kiritilgan bo`lsa TRUE qiymatni qaytaradi. Agar birorta ham belgi kiritilmagan bo`lsa funktsiya klaviaturadan kiritishni kutadi.
Quyidagi dasturda belgili massiv elementlarini kiritishda EOLN funktsiyasi ishlatilgan.

```
const
      SIZE=30;
Var
      Buf: Array[1..Size] of Char;
      n: Integer;    {kiritilgan qatorning real uzunligi}
Begin
      Write (‘->’);
      n:=0;
      Repeat
        If Not EOLN then Begin  n:=n+1; Read (Buf[n]); End;
      Until EOLN OR (n=Size);
    Writeln (‘Kiritilgan qator uzunligi ‘, n, ‘belgilar.’);
 End.
```

7.3.Qatorlar va ularni qayta ishlash funktsiyalari

Pascal tilida ketma-ketlikda berilgan belgilar qator deyiladi. Qatorlar dastur boshida STRING so`zi bilan e’lon qilinadi. Har qanday STRING tipidagi o`zgaruvchi ketma-ketlikdagi belgilar uzunligi 255 tadan oshmagan belgili qiymatlarni qabul qiladi. Qatorli tipdagi o`zgaruvchilar dastur boshida quyidagicha ko`rsatilishi kerak:

```
Var
   nom: STRING;
yoki
   nom: STRING [uzunlik];
```

bu erda *nom* – qator tipdagi o`zgaruvchi nomi.
Misollar:

```
Var
      name: STRING [30];
      Buff, RT: STRING;
```

Agar e’lon qilinganda o`zgaruvchi uzunligi ko`rsatilmasa, unda qator uzunligi 255 belgiga teng deb qabul qilinadi. Qatorli o`zgarmas ochiluvchi va yopiluvchi kavichka ichida yoziladi. Masalan,

Parol:=’sekret’; yoki Parol:=’1997’;

STRING tipdagi o`zgaruvchini boshqa bir STRING tipidagi o`zgaruvchi bilan yoki qatorli o`zgarmas bilan =, <, >, <=, >=, <> operatorlari yordamida taqqoslash mumkin.

Quyidagi dastur parol so`raydi, agar parol to`g`ri kiritilgan bo`lsa familiyani kiritish kerak bo`ladi.

```
Var
    Name: String[30]; Parol: String[6];
 Begin
    Write (‘Parol (6 belgi)?’);
    Readln(Parol);
    If Parol=’sekret’ then begin
                            Write (‘Sizning familiyangiz?’);
                            Readln(Name);
                            Writeln(Name, ‘, Sizga ruxsat berildi.’);
                          End
                            Else Writeln(‘Parolь nato`g`ri! Sizga ruxsat yo`q!’);
 End.
```

Taqqoslash operatsiyasidan tashqari qatorli o`zgaruvchilar va o`zgarmaslar ustida qo`shish amalini ham bajarish mumkin. Masalan:

```
F_Name:=’Ivanov’;
L_Name:=’Ivan’;
Ful_name:=F_Name+L_Name;
```

Bunda Ful_Name o`zgaruvchi ‘IvanovIvan’ qiymatni oladi.

Turbo Pascal qatorlarni qayta ishlashda qo`llaniladigan bir necha protsedura va funktsiyalarga ega.

LENGTH funktsiyasi. Bu funktsiya bitta qator tipli o`zgaruvchi - parametrga ega bo`lib, qator uzunligini qaytaradi. Qaytariladigan funktsiya qiymati butun sondan iborat, u qatordagi bulgilar sonini aniqlaydi. Agar qator boshida yoki oxirida bo`shliqlar bo`lsa ularni hisobga olmaydi. Masalan, LENGTH(‘Ivanov’) funktsiya qiymati 6 ga teng, LENGTH(‘ Ivanov Ivan ’) funktsiya qiymati 11 ga teng.

DELETE protsedurasi. Bu protsedura qatorning ma’lum bir qismini yo`qotadi. Umumiy holda bu protsedura quyidagicha yoziladi:

Delete (*qator,p,n);*

bu erda *qator* – qator tipdagi o`zgaruvchi; *p* – o`chirilishi kerak bo`lgan boshlang`ich belgining nomeri; n – o`chiriladigan qator bo`lagi uzunligi. Misol:

```
S:=’Shahar Sankt-Peterburg`;
Delete (S,8,6);
```

Natija ‘Shahar Peterburg` bo`ladi.

POS funktsiyasi. Bu funktsiya qatorda qator bo`lagining joyini aniqlaydi. Umumiy holda bu funktsiya quyidagicha yoziladi:

POS (*qatorbo`lagi, qator);*

bu erda *qatorbo`lagi* – qatordan topilishi kerak bo`lgan qator tipidagi o`zgaruvchi yoki o`zgarmas. Misol: p:=POS(’Pe’, ‘Sankt-Peterburg`);

Bunda o`zgaruvchi p qiymati 7 ga teng bo`ladi. Agar izlanayotgan qator bo`lagi bo`lmasa funktsiya nul qiymat qaytaradi.

COPY funktsiyasi. Bu funktsiya qatordan fragment, ya’ni bo`lak ajratib oladi. Umumiy holda bu funktsiya quyidagicha yoziladi:

COPY (*qator, p, n);*

bu erda *qator* – qator tipidagi o`zgaruvchi yoki o`zgarmas; p – qator bo`lagi boshlanadigan qatordagi birinchi belgi nomeri; n – qator bo`lagi uzunligi. Misol:

st:=’Muxandis Ahmedov’;

Fam:=Copy(st, 10, 7);

Bunda o`zgaruvchi Fam qiymati ‘Ahmedov’ bo`ladi.

VAL protsedurasi. Bu protsedura qator ko`rinishida berilgan sonlarni oddiy songa o`zgartiradi (o`tkazadi). Umumiy holda bu protsedura quyidagicha yoziladi:

VAL (*qator, son, kod);*

bu erda *qator* – sonni tasvirlovchi qator tipdagi o`zgaruvchi yoki o`zgarmas; *son* – qatordagi sonni o`zlashtiruvchi o`zgaruvchi; *kod* – protsedura qaytaradigan xatolik kodi. Agar qatorda berilgan son oddiy songa o`tkazilgan bo`lsa xatolik kodi nulga teng bo`ladi. Misol:

S:=’891.01’; VAL (S, R, kod); Natija R=891.01 bo`ladi.

Savollar

1.Belgili tipdagi o`zgaruvchilar qanday e’lon qilinadi?
2. Belgililarni qayta ishlashda CHR funktsiyasi nima vazifa bajaradi?
3. Belgili massivlarga tushuntirish bering.
4. EOLN (End Of Line) qurilgan funktsiyasidan qachon foydalanish mumkin?
5. Qatorli tipdagi o`zgaruvchilar qanday e’lon qilinadi?
6. Qatorning maksimal uzunligi nechta belgiga teng bo`lishi mumkin?
7. Qatorlarni qayta ishlashda qo`llaniladigan qanday protsedura va funktsiyalarni bilasiz?
8. LENGTH va POS funktsiyalari nima vazifa bajaradi?
9. DELETE protsedurasi nima vazifa bajaradi?

8.TURBO PASCAL GRAFIK MODULLARI

Kompyuter grafikasi ilmiy tajriba va loyihalashning avtomatlashtirilgan tizimlarda, robototexnika va boshqa sohalarda keng ishlatiladi.

Kompyuter monitori grafik va matnli ma'lumotlarni ekranga chiqaradi. Monitor maxsus videoadapter boshqaruvida matnli va grafikli holatda ishlaydi. Matnli ish holatida kompyuter ekrani har bir qatorda 80 tadan pozitsiya bo`lgan 25 ta satrga bo`linadi. Grafikli ish holatida esa monitordagi har qanday tasvir xuddi televizor ekranidek, ya'ni ma'lum bir ranga bo`yalgan nuqtalar to`plami kabi hosil bo`ladi. Tasvirning qanchalik tiniqligi videoadapterning imkoniyatiga, ya'ni gorizontal va vertikal bo`yicha chiqariladigan nuqtalar miqdoriga qarab belgilanadi. Masalan, videoadapter SVGA ekranga quyidagicha - 640x480, 800x600, 1024x768 nuqtalarni beradi.

8.1.Ekranni grafik holatga o`tkazish

Oddiy matnli щolatdan grafik щolatga o`tish uchun Graph modulining **InitGraph** prosedurasi ishlatiladi.

InitGraph(Gd,Gm,Path);

bu erda Gd -drayver nomeri; Gm -rejim nomeri;
Path -kerakli drayverga yo`l ko`rsatadi.

Agar Path='' bo`sh bo`lsa drayverni joriy kotologdan izlaydi. Agar Gd=0 bo`lsa, kerakli drayverni o`zi avtomatik ravishda tanlaydi, ya'ni Gd=Detect. Detect nolga teng parametr. Graph - modulini chaqirish dastur bosh qismida beriladi, ya'ni

```
uses
  Graph;
```

Grafik rejimni oldingi holatiga qaytarish uchun, ya'ni yopish uchun **CloseGraph**; prosedurasi ishlatiladi.

8.2.Graph modulining protsedura va funktsiyalari

Ekranda har bir nuqta o`z koordinatasiga ega. Koordinata boshi ekranning yuqori chap burchagi bo`lib, u (0,0) dan boshlanadi. Nuqtalar koordinatasi pastga va o`nga qarab o`sib boradi.

Graph – modulining protsedura va funktsiyalarini ko`rib chiqamiz:

PutPixel(x,y,color); -protsedura ekranda koordinatasi (x,y) bo`lgan nuqtani tasvirlaydi. Color parametri shu nuqtaga rang beradi.

Misol: PutPixel(100,120,Red); bu erda ekranning (100,120) nuqta qizil rangda tasvirlanadi. Red=4 nuqtaga qizil rang beradi.

GetPixel(x,y); -funktsiyasi nuqta rangini aniqlaydi.

Misol: Col:=GetPixel(50,80); bu erda Col parametriga nuqta rangi yuboriladi.

Line(x1,y1,x2,y2); -protsedura koordinatalari (x1,y1) va (x2,y2) bo`lgan kesmani chizadi.

Circle(x,y,Radius); - markazi (x,y) nuqta bo`lgan va radiusi Radius bo`lgan aylana chizadi.

Rectangle(x1,y1,x2,y2); -protsedura to`g`ri to`rtburchak chizadi.

(x1,y1) -yuqori chap burchak koordinatasi;

(x2,y2) -pastki o`ng burchak koordinatasi.

SetColor(Color); -protsedura rasmga rang beradi. Color rang nomeri.

Quyidagi ranglar ishlatiladi:

Rang nomeri	Rang	Rang nomeri	Rang
Bleck=0	qora	Blue=1	ko`k
Green=2	yashil	Cyan=3	Bryuzarang
Red=4	qizil	Magenta=5	malinarang
Brown=6	jigarrang	LightGray=7	och kulrang
DarkGray=8	to`q kulrang	LightBlue=9	och havorang
LightGreen=10	och yashil	LightCyen=11	och bryuzarang
LightRed=12	och qizil	LightGray=13	och malinarang
Yellow=14	sariq	White=15	oq

Misol 2.

Quyidagi dastur to`g`ri to`rtburchak chizadi.

```
Program TT;
  Uses Graph;
       Var gd,gm: Integer;
       Begin
          gd:=detect;   InitGraph(gd,gm,'');
              Rectangle(30,30,120,120); Readln;
              CloseGraph;
      End.
```

Bar(x1,y1,x2,y2); -rangli yoki shtrixlangan to`g`ri to`rtburchak chizish;

Bar3d(x1,y1,x2,y2,depth,top); -rangli yoki shtrixlangan paralelopeped chizish;

FillEllipse(x,y,xradius,yradius); -rangli yoki shtrixlangan ellips chizish;

SetFillStyle(Style,Color); -shtrix va rang berish. Bu erda Style shtrix, Color rang tanlovchi o`zgarmas parametr.

Style o`zgarmas parametr bo`lib, u har xil shtrixlar bilan figuralarni to`ldiradi. Ular quyidagilardir:

```
Const
  EmptyFill=0;      { fon rangi }
  SolidFill=1;      { joriy rangni beradi }
```

```
LineFill=2;          { qalin gorizantal chiziq }
StstashFill=3;       { ingichka qiyshiq chiziq }
StoshFill=4;         { qalin qiyshiq chiziq }
BkStashFill=5;       { qalin qiyshiq chiziq }
LtstashFill=6;       { qiyshiq yo`l-yo`l chiziq }
HatchFill=7;         { katakchalar bilan to`ldirish }
XhatchFill=8;        { qiyshiq katak bilan to`ldirish }
InterLeaveFill=9;    { qiyshiq shtrix chiziq }
WideDotFill=10;      { kam nuqtalar bilan to`ldirish }
CloseDotFill=11;     { bo`lak-bo`lak nuqtalar  }
UseFill=12;                { foydvlanuvchi shtrixi }
```

Grafik rejimda matn yozish uchun quyidagi protseduralar ishlatiladi.

SetTextStyle(Font, Detection, Size); -kerakli shriftni ishga tushiradi.

Bu erda

Font - shriftni tanlash;
Detection - yozuv yo`nalishini belgilash;
Size - shrift o`lchamini tanlash.

SHrift va matn yozuvi yo`nalishi quyidagi o`zgarmas bilan aniqlanadi.

```
Const                          { shrift }
    DefaultFont=0;     { standart shrift }
    TriplexFont=1;     { vektorli shrift }
                      { tekst yo`nalishi }
    HarizDir=0;         { chapdan o`nga }
    VertDir=1;          { patsdan yuqoriga }
```

OutTextXY(x,y,TextString); -(x,y) koordinatali nuqtadan TextString nomli matn qatorini kiritadi.

Savollar

1.Kompyuter monitori qanday holatlarda ishlaydi?
2.Videoadapter monitorda nima ish bajaradi?
3.Grafik modulini dasturda chaqirish qanday bajariladi?
4.Grafik holatga qaysi protsedura yordamida o`tiladi?
5.Ekranda nuqta va to`g`ri chiziqlar chizish qaysi protseduralar yordamida bajariladi?
6.Ekranda aylana va to`g`ri to`rtburchak chizish qaysi protseduralar yordamida bajariladi?
7.Rang berish protsedurasini tushuntirib bering.
9.Grafik rejimda matn yozish uchun qaysi protseduralar ishlatiladi.

9.MODULLAR VA MA'LUMOTLAR FAYLLARI

Turbo Pascalь sistemasida juda ko`p maxsus tayyor protsedura va funktsiyalar mavjudki ular har qaysi o`z vazifasiga ega bo`lib unga biblioteka modullari deyiladi. Har bir biblioteka funktsiya va protseduralardan tashkil topgan bo`lib ma'lum bir turdagi masalani yechishga mo`ljallangan bo`ladi.

9.1.Modul tushunchasi

Modul deb protsedura va funktsiyaning alohida kompilyatsiya qilinib maxsus .tpu kengaytmali fayl shaklidagi ifodalangan dastursiga aytiladi. Moduldan ixtiyoriy dastur ichida foydalanish mumkin. Moduldan foydalanish ya'ni uni aktivlashtirish uchun dasturning bosh qismida quyidagilarni keltirish zarur.

Uses <Modul1 nomi, modul2 nomi, . . . ,moduln nomi>;

Misol: Program SS;
Uses Crt,Graph;

Turbo-Pascalь sistemasida har bir foydalanuvchi o`z modulini yaratishi uchun yaratiladigan modul strukturasini quyidagicha tashkil kilish zarur.

Unit <modul nomi>;
Interface
. . . . .
{Interfeys qism- ochiq (yozuvlar) qismi}
. . . . .
Implementation
. . . . .
{YOpiq (yozuvlar) qismi}
. . . . .
Begin
. . . . .
{Modulning asosiy qismi}
. . . . .
End.

Bu erda **Unit** - modulning sarlavhasi;

Interface - modulning interfeysi, ya'ni dastur va boshqa modullar uchun ochiq (ko`rinarli) qismining boshlanishini bildiradi. Bu qismda o`zgarmaslar, kattaliklar tiplari, protsedura va funktsiyalar aniklanib kursatilgan bo`ladi, lekin ularning butun ko`rinishi keyingi yopiq qismda beriladi.

Implementation - modulning dastur va modullar uchun yopiq, ya'ni ko`rinmaydigan qismining boshlanishini bildiradi. Bu erda interfeys qismda aniqlangan protsedura va funktsiyalar yana bir marta ko`rsatilishi shart (ularning sarlavhalari bir xil bo`lishi kerak).

Initsializatsiya qismi Begin yozuvidan keyin boshlanadi, agar bu qism mavjud bo`lmasa Begin ham bo`lmaydi. Bu qismda boshqaruvni asosiy dasturga o`tkizishgacha qadar bajarilishi kerak bulgan operatorlar ruyxati joylashadi.

9.2.Pascal biblioteka modullari

Turbo Pascalь sistemasida quyidagi biblioteka modullari majud:

System - standart protsedura va funktsiyalarni o`z ichiga olgan bo`lib, bu modul avtomatik ravishda aktivlashtirilgan bo`ladi.

Dos - Ms Dos operatsion tizim protsedura va funktsiyalarni o`z ichiga oladi.

Crt - monitor ekrani va klaviatura bilan ishlash imkoniyatini yaratuvchi protseduralar to`plamini o`z ichiga olgan.

Graph - grafik protsedura va funktsiyalar tuplamini o`z ichiga oladi.

Printer - printer bilan ishlovchi kichik modul.

9.3.Modul tashkil qilish

Turbo Passal dasturlash vositasi yordamida har bir dastur tuzuvchi o`z modulini yaratish imkoniga ega. Buning uchun kerkli dastur qismini yuqorida ko`rsatilgan struktura ko`rinishda yozish va uni kompilyatsiya qilish kerak.

Misol tariqasida ikki sonning eng katta va eng kichigini topish modulini yaratish dasturini qaraymiz.

```
Unit Study;
  Interface {Interfeys qism }
        Function Min(x,y:Integer):Integer;
        Function Max(x,y:Integer):Integer;
  Implementation {YOpik qism}
        Function Min(x,y:Integer):Integer;
      Begin
              If x<=y Then Min:=x Else Min:=y;
       End;
        Function Max(x,y:Integer):Integer;
      Begin
              If x>=y Then Max:=x Else Max:=y;
       End;
     {Inistalyatsiya qism mavjud emas}
End.
```

Bu modul kompilyatsiya qilinib Stadu.tpu fayl nomga ega bo`lishi kerak. Undan dasturda foydalanish uchun dastur bosh qismida Uses Stadu qatorini yozish kerak bo`ladi.

9.4.O`zgaruvchilar qiymatlarini fayldan o`qish va faylga yozish

Kiritiladigan va chiqariladigan ma'lumotlar soni ko`p miqdorda bo`lsa ularni faylda saqlash dasturda qulaylik tug`diradi. Bu ma'lumotlar oddiy matn (tekst) fayllarida saqlanadi. Fayl o`zgaruvchisi dastur bosh qismida e'lon qilinadi, ya'ni

```
Type f=text;
Var fx:f;
```

bu erda f -fayl tipi, oddiy matn faylni bildiradi; fx-fayl o`zgaruvchisi.

Kerakli fayldan ma'lumotlarni o`qishga tayyorlash uchun Assign va Reset funktsiyalari ishlatiladi.

Assign-fayl o`zgaruvchisi bilan ososiy fayl orasida aloqa o`rnatadi.

Assign (fx,'c:\a\f1.txt');

Reset-faylni topib uni ishga tayyorlaydi. **Reset (fx);**

Bu erda fx- fayl o`zgaruvchisi; 'c:\a\f1.txt'-c: diskning A katalogidagi fx.txt fayldan o`qishni bildiradi.

Fayldagi ma'lumotlarni o`qish uchun Read funktsiyasi ishlatiladi.

Read (<fayl o`zgaruvchisi>, <o`zgaruvchilar,massivlar>);

Misol. Read (fx, x,y,z,A[i],B[I]);

Fayldan o`zgaruvchilar yoki massivlar qiymatlarini o`qib bo`lgandan keyin fayl yopiladi. Faylni yopish quyidagi funktsiya yordamida bajariladi. Close (fayl o`zgaruvchisi); Misol. Close (fx);

Misol 1. C: diskdagi AA katalogdagi AB fayldagi ma'lumotlarni o`qib A va B massivlarga joylashtiring. Fayl ma'lumotlari strukturasi quyidagicha.

```
15.2 20.5
                20.1 25.5
                . . . . . .
```

ya'ni fayl strukturasi ikki ustundan iborat ma'lumotlar to`plamidan iborat.

```
Program FAB;
  Type  f=text;
  Var A,B: Array[1..100] of Real;   m: Integer;
       fxz: f;
  Begin
      Assign(fxz,'c:\AA\AB.txt');   Reset(fxz);
      m:=0;
      While not eof(fxz) do
        Begin  m:=m+1;  Readln(fxz,A[m],B[m]);   End;    Close (fxz);
 End.
```

Massiv qiymatlarini biron matn fayliga yozib qo`yish uchun Assign, Rewrite, Append, Write va Close funktsiyalari ishlatiladi.

Assign- fayl o`zgaruvchisi bilan asosiy fayl o`rtasida a'loqa o`rnatadi va u quyidagicha yoziladi.

Assign(fayl o`zgaruvchisi, diskdagi fayl joyi va nomi);

Append- Fayldan yozish uchun joy tayyorlayli.

Append(fayl o`zgaruvchisi);

Write- o`zgaruvchi qiymatini fayl o`zgaruvchisiga uzatadi va faylga yozadi. **Write(fayl o`zgaruvchisi, o`zgaruvchilar);**

Close- ochilgan faylni yopadi. **Close (fayl o`zgaruvchisi);**

Misol 2. YUqoridagi C: diskdagi AA katalogdagi AB fayldagi ma'lumotlarni A va B massivlarga o`qib shu massiv mos elementlarini qo`shib S massivni tashkil qiling va A,B,C massivlarini ABC nomli faylga yozing.

Fayl ma'lumotlari strukturasi quyilagicha bo`lsin.

i	A	B	C
1	15.2	20.5	45.7
2	20.1	25.5	45.6
. . . . .			

```
Program FABS;
  Type  f=text;
  Var  A,B,C: Array[1..100] of Real;  i,m: Integer;
        fax: f;
  Begin
      Assign(fax,'c:\AA\AB.txt');   Reset(fax);
      m:=0;
      While not eof(fax) do
        Begin     m:=m+1;   Readln(fax,A[m],B[m]);  End;
         Close (fax);
      Assign(fax,'c:\AA\ABC.txt');   Rewrite(fax); Append(fax);
      For i:=to m do
       Begin  Write(fax,i,A[i],b[i],c[i]); Writeln(fax);  End;
      Close(fax);
End.
```

Savollar

1.Modul deganda nimani tushunasiz?
2.Oddiy protsedura va funktsiyaning biblioteka moduli protsedura va funktsiyasi bilan nima farqi bor?
3.Modullar qanday chaqiriladi, ya'ni aktivlashtiriladi?
4.YAngi modul qanday tashkil qilinadi?
5.Qanday maxsus biblioteka modullarni bilasiz?
6.System -moduliga qanday protsedura va funktsiyalar kiradi?
7.Crt -moduli nima maqsadda ishlatiladi?
8.Graph -modulini tushuntiring.

10.DELPHI DASTURLASH VOSITASI

Pascal tili SHvetsariyalik olim N.Virt tomonidan yaratilib, keyinchalik Borland korporatsiyasi tamonidan rivjlantirildi. Bu til rivjlantirilib Turbo Pascal, Borland Pascal va keyinchalik esa Object Pascal nomini oldi. Hozirgi kunda Object Pascal tili asosi bo`lgan Windows muhitida ishlovchi Delphi dasturlash vositasining yaratilishi nafaqat professional dasturchilar, balki oddiy dastur tuzuvchilar uchun ham keng yo`l ochib berdi.

10.1.Delphi tizimi oynasi elementlari

Delphi -bir necha muhim ahamiyatga ega bo`lgan texnologiyalar kombinatsiyasini o`zida mujassam etgan:

- yuqori darajadagi mashinali kodda tuzilgan komplyator;
- ob'ektga yo`naltirilgan komponentalar modullari;
- dastur ilovalarini vizual tuzish;
- ma'lumotlar bazasini tuzish uchun yuqori masshtabli vosita.

Delphi oynasi ko`rinishi odatdagidan ancha farq qiladi va u o`z ichiga beshta oynachani oladi:

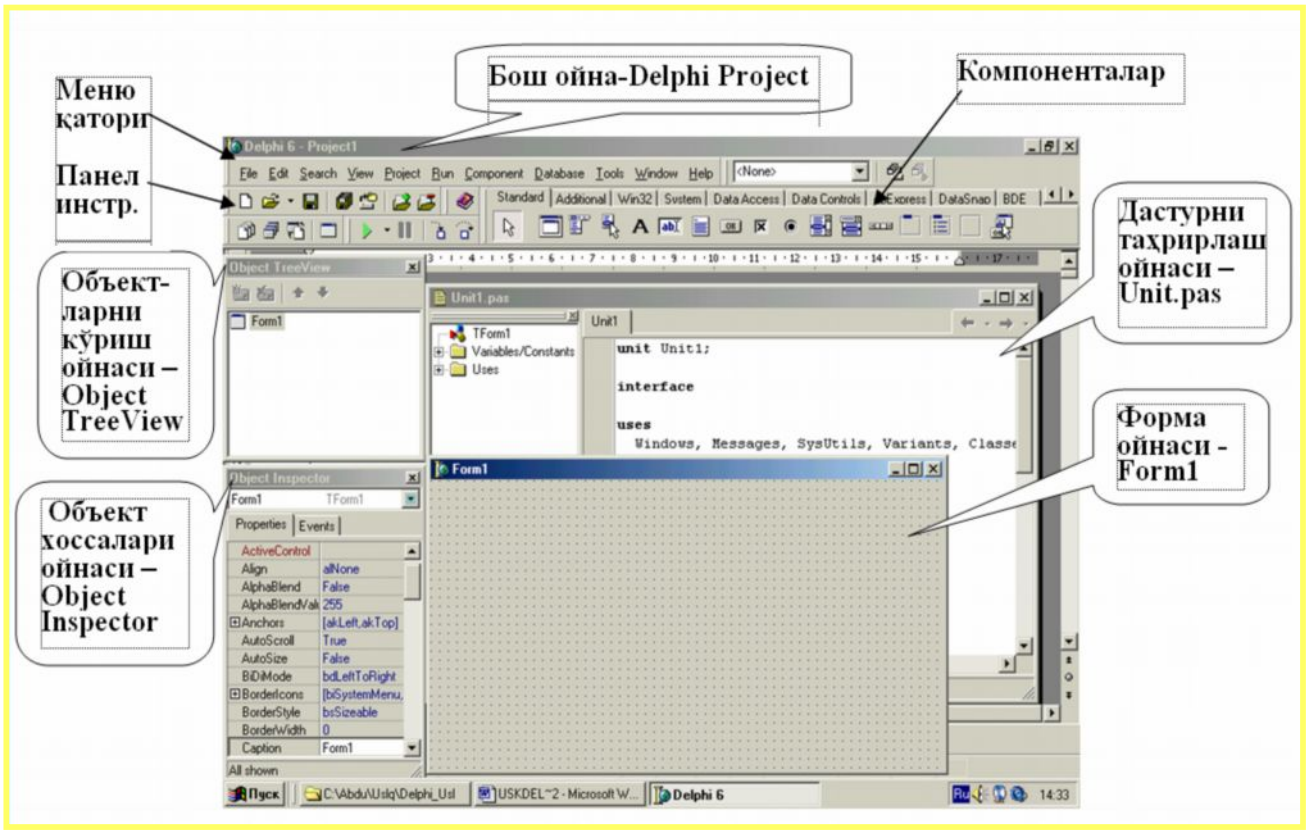
- bosh oyna - Delphi Project1;
- forma oynasi - Form1;
- ob'ekt xossalarini o`rnatish oynasi-Object Inspector;
- ob'ektlar ierarxik ro`yxatini ko`rish oynasi - Object tree View;
- dastur kodlarini kiritish va tahrirlash oynasi - Unit.pas.

Delphi dasturlash muhitida ishlash jarayonida quyidagi kengaytmali fayllar tashkil etiladi:

-loyiha fayli, kengaytmasi **.dpr**;
-Pascal moduli fayli, kengaytmasi **.pas**;
-komponentalar joylashgan fayl, kengaytmasi **.dcu**;
-formalar ma'lumotlari joylashgan fayl, kengaytmasi **.dfm**;
-ma'lumotlar bazasi fayli, kengaytmasi **.dbf**.

Delphi avtomatik ravishda dasturngizni Bin papkasiga joylashtiradi.

Dastur kompilyatsiya qilinishi paytida Delphi tizimi pas, dfm va dcu kengaytmali modullar tuzadi. .pas kengaytmali fayl kodlarni yozish oynasida kiritilgan dastur matnini, .dfm forma oynasi tashkil etuvchilarini, .dcu kengaytmali fayl esa .pas va .dfm kengaytmali fayllarning birgalikdagi mashina kodiga o`tkazilgan variantini saqlaydi. .dcu kengaytmali fayl komplyator tamonidan tashkil qilinadi va yagona ishchi (bajariluvchi) .exe kengaytmali fayl tashkil qilishga baza yaratadi.



10.2.Delphida dastur loyihasi strukturasi

Delphida tuzilgan dastur - bu bir necha bir biri bilan bog`liq fayllardir. Har qanday dastur .dpr kengaytmali loyiha fayli va bir yoki bir necha .pas kengaytmali modullardan tashkil topadi. Loyiha fayli dasturchi tamonidan kiritilmaydi, u foydalanuvchining ko`rsatmalari asosida avtomatik ravishda Delphi tizimi tamonidan tuziladi. Loyiha fayli matnini ko`rish uchun Project/View Source buyrug`ini berish mumkin. Loyiha matni umumiy holda quyidagicha bo`ladi.

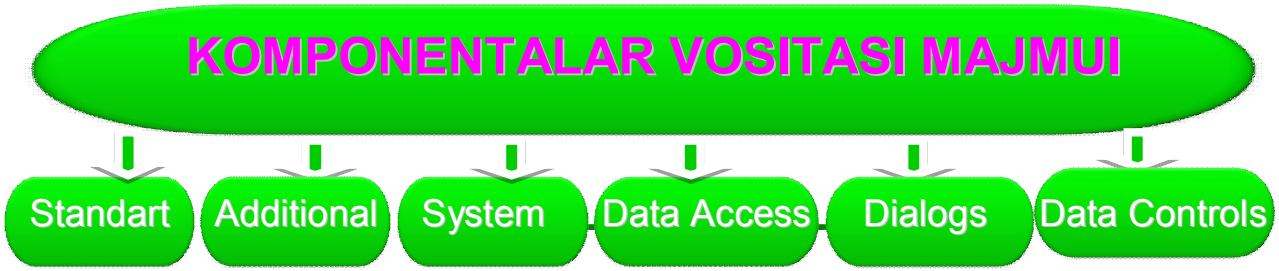
```
Program Project1;
        Uses
            Forms,
            Unit1 in ‘Unit1.pas’ {Form1}
        {$R *.res}
  Begin
        Application.Initialize;
        Application.CreateForm(Tform,Form1);
        Application.Run;
  End.
```

Delphi tizimini ishga tushirgandagi modul strukturasi quyidagi ko`rinishda bo`ladi.

```
Unit unit1;
Interface
Uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms, Dialogs;
```

```
Type
  TForm1 = class(TForm)
  Private
        { Private declarations }
  Public
        { Public declarations }
  end;
Var
  Form1: TForm1;
  Implementation
  {$R *.dfm}
End.
```

10.3.Delphi forma komponentalari



Forma komponentalari dasturni boshqarish uchun maxsus tugmachalar bo`lib uni formaga joylashtirishdan oldin bosh oynadan kerakli komponentalar palitrasi tanlanadi. Masalan, Standart (Standart) komponentalar palitrasida quyidagi piktogrammalar (tugmachalar) majmuasi mavjud:

MainMenu -murakkab ierarxik strukturali bosh menyu yaratadi.
PopupMenu - kontekst menyusini yaratadi.
Label -formaga matnli ma'lumotlarni joylashtiradi.
Edit -bir qatorli matnli ma'lumot kirish va chiqarish.
Memo -ko`p qatorli matnli ma'lumot kiritish va chiqarish.
Button -formada buyruq tugmasini yaratadi.
CheckBox -bog`liq bo`lmagan tanlash tugmalarini yaratadi.
RadioButton -bog`liq bo`lgan tanlash tugmalarini yaratadi.
ListBox -ro`yxat variantlarini taqdim etadi.
ComboBox -ro`yxatdan tanlab kiritish qatorini yaratadi.
GroupBox -bir necha bog`lik komponentalarni gruhlaydi.
RadioGroup -bog`liq guruhlangan tanlash tugmalarini yaratadi.

Forma va komponentalar xossalari. (mustaqil o`rganiladi**)**

ActiveControl – Ko`zda tutilgan bo`yicha aktiv bo`lishi lozim bo`lgan komponentaga ko`rsatadi.
Align – Komponentani tekislash. Qiymatlari:
alNone -Tekislanmaydi.
alBottom -Pastki chegaraga tekislash.
alLeft -CHap chegaraga tekislash.
alRight -O`ng chegaraga tekislash.
alTop -YUqori chegaraga tekislash.
AlphaBlend – Mantiqiy tur. Forma xossasi. Agar qiymati rost bo`lsa forma shaffof bo`ladi.
AlphaBlendValue – Butun tur. SHaffoflik darajasi. Qiymati 0 dan 255gacha.
Anchors – Forma va komponenta xossasi. Ajdod ob'ektga mahkamlanishni ko`rsatadi. Qiymatlari:
akLeft -chap chegaraga mahkamlash.
akTop -yukori chegarana mahkamlash.
akRight -o`ng chegaraga mahkamlash.
akBottom -pastki chegaraga mahkamlash.
AutoScroll – Mantiqiy tur. Agar kiymati true bo`lsa forma avtomatik siljitishni amalga oshiradi.
AutoSize – Mantiqiy tur. Komponentalar formada avtomatik o`lchamlarini o`zgartirishini ko`rsatadi.
BorderIcons – Oynada qanday tugmalar bo`lishini ko`rsatadi.
biSystemMenu -menyuni ko`rsatish.
biMinimize -minimallash tugmasi.
biMaximize -maksimallash tugmasi.
biHelp -yordam tugmasi.
BorderStyle – Forma xossasi. Forma chegarasi turini belgilaydi.
bsSizeable -Standart oyna. Kattaligini o`zgartirish mumkin.
bsNone -CHegaraviy xoshiyasiz oyna.
bsSizeToolWin -Ingichka o`zgatiriladigan xoshiyali oyna.
bsToolWindow -Ingichka o`zgatirilmaydigan xoshiyali oyna.
BorderWidth – Butun tur. Forma chegarasi kengligini belgilaydi.
Caption – Satrli tur. Oyna yoki komponenta sarlavhasi.
ClientHeight – Butun tur. Oyna ishchi qismi balandligi.
ClientWidth – Butun tur. Oyna ishchi qismi kengligi.
Color – Oyna klient qismi rangi.
Constraints – Oyna o`lchamlari qiymatlari. Parametrlari:
MaxHeight - maksimal balandlik.
MaxWidth - maksimal kenglik.
MinHeight - minimal balandlik.
MinWidth - minimal kenglik.
Cursor – Sichqoncha tomonidan forma yoki komponentaga keltirilganda ko`rsatiladigan kursor shakli.
DockSite – Mantiqiy tur. Formaga boshqa komponentalarni Drag&Drop yordamida tashlash mumkinligini ko`rsatadi.

DragKind - Drag&Drop da ob'ektni ko`chirish turi:
dkDrag -Ctandart Drag&Drop. Ob'ekt joyida qoladi.
dkDock -Ob'ekt o`zi ko`chiriladi.
DragMode –Drag&Drop rejimi. Ikki variant mavjud:
dmManual - Ob'ektni ko`chirish rejimi foydalanuvchi tomonidan o`rnatiladi.
dmAutomatic -Draq&Drop rejimi avtomatik ishga tushadi.
Enabled – Mantiqiy tur. Agar xossa qiymati true bo`lsa, foydalanuvchi bu komponenta bilan ishlashi mumkin.
Font – Matnni formaga chiqarishda ishlatiladigan shrift. Ikki marta shu qatorga chertilsa Windows shrift tanlash standart oynasi chiqadi.
FormStyle – Forma turi. Quyidagi variantlar mavjud:
fsNormal -Normal oyna.
fsMDIForm -MDI oynalar uchun ajdod oyna.
fsMDIChild -Avlod MDI oyna.
fsStayOnTop - Oyna har doim qolganlari ustida bo`ladi.
Height – Butun tur. Oyna balandligi.
Hint – YOrdamchi ma'lumot matnini chiqarish.
HorzScrollBar – Gorizontal siljitish yulchasi.
Left –Butun tur. Oyna chap pozitsiyasi.
Menu –Asosiy oynada foydalaniladigan menyu.
Name - Forma yoki komponenta nomi.
ParentFont – Mantiqiy tur. Agar qiymati true bo`lsa matn uchun bosh ob'ekt shrifti, aks holda foydalanuvchi ko`rsatgani tanlanadi.
Position – Dastur ishga tushganda oyna pozitsiyasi. Variantlari:
poDefault – Oyna o`rni va o`lchamlarini Windows tanlaydi.
poDefaultPosOnly – Oyna o`rnini Windows, o`lchamlarini foydalanuvchi tanlaydi.
poDefaultSizeOnly – Oyna o`rnini foydalanuvchi, o`lchamlarini Windows tanlaydi.
poDesigned – Oyna o`rni va o`lchovlarini foydalanuvchi tanlaydi.
poDesktopCenter – Oyna ishchi stoli markazida joylashadi.
poMainFormCenter – Oyna forma markazida joylashadi.
poOwnerFormCenter – Oyna o`zini chaqirgan oyna markazida joylashadi.
poScreenCenter - Oyna ekran markazida joylashadi.
ShowHint – Mantiqiy tur. YOrdamchi axborot ko`rsatish kerakligini belgilaydi.
Tag – Butun tur. Hech narsaga ta'sir qilmaydi.
Top – Butun tur. Oyna yuqori pozitsiyasi.
TransparentColor – Mantiqiy tur. Agar qiymati true bo`lsa forma yoki komponenta har doim shaffof bo`ladi.
TransparentColorValue – SHaffof rang.
VertScrollBar – Vertikal siljitish yulchasi.
Visible – Mantiqiy tur. Agar kiymati true bo`lsa, forma yoki komponenta ko`rinadi, aksincha ko`rinmaydi.
Width – Butun tur. Oyna kengligi.
WindowState – Oyna holati. Quyidagi parametrlari mavjud:
wsNormal – oyna normal holatda.
wsMaximized – oyna maksimal holatda.
wsMinimized – oyna minimal holatda.

10.4.Label, Edit, Memo va Button komponentlari

Oddiy masalalarni yechish uchun dastur yaratishda asosan Label, Edit, Memo matn komponentlari va Button tugmachasi etarlidir. Bu kompontalarni ko`rib chiqamiz.

Label belgisi. Belgi tushuntirishlar, nomlar, mavzular va boshqa har xil turdagi matnli ma'lumotlarni ekranga joylashtirish uchun ishlatiladi. Belgi uchun **Caption** asosiy xossalardan biri bo`lib, unda ekranga chiqariladigan matn joylashadi.

Label komponentasi nafaqat ma'lumotlarni ekranga joylashtirish uchun xizmat qiladi, balki dastur natijalarini chiqarishda ham ishlatish mumkin. Buning uchun dasturda **Label5.caption:='Dastur natijasi';** buyrug`i berilishi kerak. Misol, **Label5.caption:='Echim='+s;** bu erda **s:String** o`zgaruvchisi.

Edit kiritish qatori. Edit kiritish qatori matnni bir qatordan kiritish va uni tahrirlash uchun ishlatiladi.

Memo matn chiqarish qatori. Memo matnlarni bir necha qator qilib chiqarish uchun ishlatiladi.

Bu matn chiqarish maydoni dasturda natijalarni chiqarishda qo`l keladi. Natijani chiqaoishda u dastur ichida quyidagicha ishlatiladi. **Memo1.Lines.add('Echim='+S);**

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

Button tugmachasi. Button tugmachasi bosilishi natijasida kutilishi lozim bo`lgan jarayonlar ishga tushiriladi.

Dasturdagi hisoblash jarayonlari hosil qilingan tugmachalarni ikki marta tez-tez bosish bilan "sobitiyani qiyta ishlash" darchasiga o`tilib, u erdan modul ichiga kerakli operatorlarni yozish bilan amalga oshiriladi.

10.5.Boshlang`ich forma ilovasini yaratish

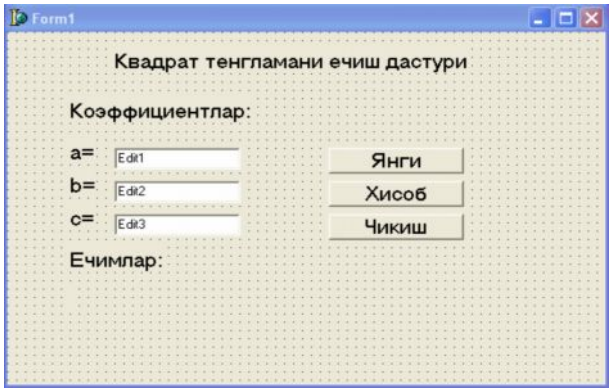
Delphida boshlang`ich formani tuzish forma Form1 xossalarini o`zgartirish bilan boshlanadi. Forma xossalarini uning tashqi ko`rinishini, ya'ni uning o`lchami, ekranda joylashishi, oynasi ko`rinishi va sarlovha matnini aniqlab beradi. Uning xossalari Object Inspector oynasida keltirilgan bo`lib, oynaning chap ustunida xossa nomlari va o`ng ustunida uning qiymatlari berilgan.

Formaga yangi komponentalarni joylashtirish uncha katta qiyinchilik tug`durmaydi. Masalan, Label (metka) komponentasini formaga joylashtirish uchun uni Standart komponentalar politrasidan topib sichqonchada ikki marta chiqilatiladi, u formada Label1 nom bilan joylashadi. Uning boshlang`ich holatini va xossalarini o`rnatish uchun ob'ekt inspektori oynasidan o`tiladi, masalan, Caption xossasiga kirib Label1 nomni o`chirib o`rniga, "Mening birinchi dasturim" degan so`zni yozsak farmada shu so`z namayon bo`ladi. YOzilgan matn rangi va o`lchamini o`zgartirish esa Font xossasidan Size va Color parametrlarini o`zgartirish bilan amalga oshiriladi.

Botton (tugmacha) komponentasi dasturchi tomonidan tuzilgan dasturni ishga tushirish uchun mo`ljallangan bo`lib unga hodisalarni qayta ishlovchi (On Click - obrabotchik sobыtiya) deyiladi.

M i s o l. Delphining imkoniyatlarini va uning vizual loyihalash vositasi texnologiyasini namoyish etish uchun misol tariqasida kvadrat tenglama echimlarini topish dasturini yaratishni qaraylik.

Formaga oltita metka, birinchi Label1 tenglama echimlarini chiqarish uchun, ikkinchi Label2 forma boshida ma'lumot berish uchun (masalan, tenglama koeffitsientlari:) va qolgan uchtasi Label3, Label4, Label5 taxrirlash maydaniga tushuntirish berish uchun (masalan, koeff. a) formaga qo`yiladi. Formaga yangi, hisob va chiqish tugmachalarini joylashtirish Button komponentasi uch martaformaning kerakli joylariga qo`yiladi va keyin ular nomlari, ya'ni qiymatlari xossadan aniqlanadi. Natijada quyidagi loyiha formasiga ega bo`linadi.



Formadagi buyruq tugmachalari biror ish bajarishi uchun ular sichqonchada ko`rsatilib chiqillatiladi. Sichqonchada tugmachani chiqillatish (bosish) xodisaga misol bo`lib, u ilovaning ishlash jarayonida hosil bo`ladi. Bu erda hodisa so`zini yuz beradigan jarayon deb tushinish kerak.

Hodisalarga javob Delphida ularning qayta ishlovchi protseduralar ko`rinishida tashkil qilinadi. Turbo Pascal tilida yoziladigan bu protseduralar hodisa qayta ishlovchisi ("obrabotchik") deb ataladi.

"Hisob" tugmasini ikki marta tez-tez chiqillatish bilan ekranga **Tform1.Button1click** protsedurasi chaqiriladi va kerakli dastur kodlari kiritiladi. Masalan:

```
Procedure Tform1.Button1click(Sender:Tobject);
 Var
     A, B, C, D, X1, X2: Real;    S1, S2: String;
 Begin
    A:=StrToFloat(Edit1.Text);
     B:=StrToFloat(Edit2.Text);
      C:=StrToFloat(Edit3.Text);
 If A=0 Then
      Begin
        X1:=-C/B;   S1:=FloatToStr(X1);
        Label6.Caption:='A=0 ! X=-C/B '+'Echim X='+S1;
      End;
 If A<>0 Then
       Begin
            D:=B*B-4*A*C;
        If D>=0 Then
            Begin
              X1:=(-B+Sqrt(D))/(2*A);   x2:=(B+Sqrt(D))/(2*A);
```

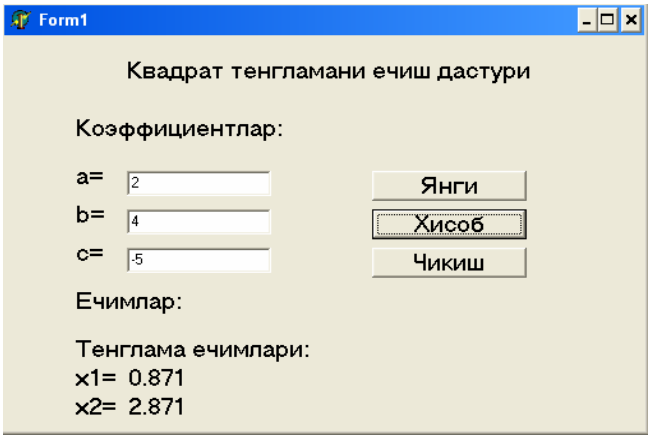
```
            S1:=FloatToStr(X1);        S2:=FloatToStr(X2);
            Label6.Caption:=' x1='+S1+' x2='+S2;
          End
          Else Label6.Caption:=' Mavjud emas D<0';
      End;
End;
```

Xuddi shunday "yangi" va "chiqish" tugmachalari uchun ham qayta ishlovchi protseduralari matnlari quyidagi ko`rinishga ega bo`ladi.

```
Procedure Tform1.Button2Click(Sender:Tobject);
 Begin
  Edit1.Text:=' ';   Edit2.Text:=' ';   Edit3.Text:=' ';
  Label2.Caption:=' ';
  Edit1.SetFocus;
End;

Procedure Tform1.Buton3click(Sender: Tobject);
 Begin
  Form1.Close;
End;
```

Delphi tizimidan chiqmasdan turib ilovani ishga tushirish mumkin, buning uchun Run menyusining Run buyrug`ini yoki F9 tugmachasini bosish kifoya bo`ladi. YUqoridagi misol uchun ilova ishga tushirilib a, b va c qiymatlari kiritilib "xisob" tugmasi bosilsa dastur quyidagi natijani ekranga chiqadi.



Protsedura TForm1.Button2Click "yangi" tugmachasini sichqonchada chiqillatish bilan ishlaydi va taxrirlash maydoniga kursorni koeffitsient qiymatlarini kiritish uchun olib kelib qo`yadi.

Protsedura TForm1.Button3Click "tamom" tugmachasini sichqonchada chiqillatish bilan ishlaydi va formani yopadi.

Savollar

1.Delphi dasturlash tili qanday kengaytmali fayllar ishlatiladi?

2.Delphi oynasi qanday elementlardan tashkil topgan?
3.Tayyor dastur nechta asosiy etapdan o`tiladi?
4.Delphida ishga tushiriladigan modul strukturasi qanday ko`rinishda bo`ladi?
5.Standart komponentalar palitrasidagi piktogrammalar majmuiga izoh keltiring.
6.Label, Edit va Memo matn komponentlarini tushuntiring
7.Button tugmachasi vazifasi nima?

11.TANLASH TUGMALARINI O`RNATISH

Ayrim masalalarda bir necha variantlardan bittasini yoki bir nechasini tanlash kerak bo`ladi. Masalan test dasturini oladigan bo`lsak unda savolga berilgan javoblardan bitta to`g`risini belgilash kerak bo`ladi. Bunday masalalarni hal etishda Delьfining RadioGroup komponentasidan foydalanish mumkin. Lekin shunday testlar borki ikkita va uchta to`g`ri javobni belgilash lozim bo`ladi. Bunday masalalarni hal etishda esa Delьfining CheckListBox komponentasidan foydalanish mumkin. Bu komponentalar Pascaldagi CASE operatoriga o`xshab ketadi.

11.1.RadioGroup komponentasi

RadioGroup guruhli tanlash tugmalari ilovalar yaratishda bir necha variantlardan birini tanlash imkonini beradi. Bu komponenta Standart komponentalar palitrasida joylashgan va u  ko`rinishdagi piktogrammaga ega. Uning asosiy xossasi Items bo`lib, u tugmalar nomlari ro`yxatini o`zida saqlaydi. Tugmalar nomlari ro`yxatini kiritishda StringList Editor oynasidan foydalaniladi. Bu oynani chaqirish uchun oldin RadioGroup tugmasiga formadan joy ajratilib, keyin Items xossasining uch nuqtali tugmachasini bosish bilan bajariladi. Bu oyna tanlash tugmalari nomlarining har qaysisi yangi qatordan kiritiladi va keyin Ok tugmasi bosiladi. Formaga RadioGroup guruhli tanlash tugmasi joylashtirilganda u RadioGroup1 nom bilan yoziladi. Bu nomni boshqa mos nomga almashtirish Caption xossasiga kirib amalga oshiriladi.

11.2.CheckListBox komponentasi

CheckListBox komponentasi ro`yxatdan bir nechtasini tanlash imkonini beradi. CheckListBox komponentasi Additional palitrasida joylashgan. RadioGroup bog`lik pereklyuchatellarni, CheckListBox esa bog`liq bo`lmagan pereklyuchatellarni birlashtiradi. Bunda klyuchatel uch xil holatda bo`lishi mumkin:

-yoqilgan (vklyuchen) -to`g`ri belgisi;
-o`chirilgan (vыklyuchen) -bo`sh belgisi;
-neytral holat -ko`kish rangda to`g`ri belgisi.

CheckListBox ning asosiy xossalari:

AllowGryer -uchinchi neytral holat variantini ishlatishni taqiqlaydi;
Items -tanlash tugmalari nomlari ro`yxatini saqlaydi.

Misol. Edit kiritish qatorida terilgan matn holatini o`zgartiruvchi ilova yaratish.

Ilova loyihasini yaratish algoritmi:

1.Formaga Standart komponentalar palitrasidan RadioGroup komponentasini RadioGroup1, RadioGroup2, RadioGroup3 nomlar bilan va uning caption xossasi qiymatini mos ravishda “yozuv”, “o`lcham” va “rang” qiymatlar bilan o`rnatamiz.

2.Items xossasiga o`tib, uch nuqtali tugmachadan StringList Editor oynasiga kirib, bu oynadan klyuchatellar nomlari har qaysisini yangi qatordan kiritamiz:

- RadioGroup1 uchun: obыchnыy; kursiv; polujirnыy; polujirnыy kursiv
- RadioGroup2 uchun: 8; 10; 12;14
- RadioGroup3 uchun: chernыy; zelenыy; krasnыy; siniy

3.CheckListBox komponentasini Additional palitrasidan formaga o`rnatamiz va uning Items xossasiga kirib vыklyuchatellar nomlarini kiritamiz: Zacherknutыy, Podcherknutыy

4.Edit komponentasini formaga o`rnatamiz va uning Text xossasiga “Kompyuter” qiymatini kiritamiz.

5.CheckListBox va Edit1 ramkalari yuqorisiga Label1 va label2 metkalarini o`rnatib Coption xossasiga “Atributlar” va “Obrazetslar” qiymatini beramiz.

6.RadioGroup1 komponentasi maydonini ikki marta tez bosib kiritish maydoniga quyidagi kodlarni kiritamiz.

```
Case RadioGroup1.ItemIndex of
  0: Edit1.Font.Style:=[ ];
  1: Edit1.Font.Style:=[FsItalic];
  2: Edit1.Font.Style:=[FsBold];
  3: Edit1.Font.Style:=[FsItalic,FsBold];
 End;
```

7.RadioGroup2 komponentasi maydonini ikki marta tez tez bosib kiritish maydoniga quyidagi kodlarni kiritamiz.

```
Case RadioGroup2.ItemIndex of
  0: Edit1.Font.Size:=8;
  1: Edit1.Font.Size:=10;
  2: Edit1.Font.Size:=12;
  3: Edit1.Font.Size:=14;
 End;
```

8.RadioGroup3 komponentasi maydonini ikki marta tez bosib kiritish maydoniga quyidagi kodlarni kiritamiz.

```
Case RadioGroup3.ItemIndex of
  0: Edit1.Font.Color:=ClBlack;
  1: Edit1.Font.Color:=ClGreen;
  2: Edit1.Font.Color:=ClRed;
  3: Edit1.Font.Color:=ClBlue;
 End;
```

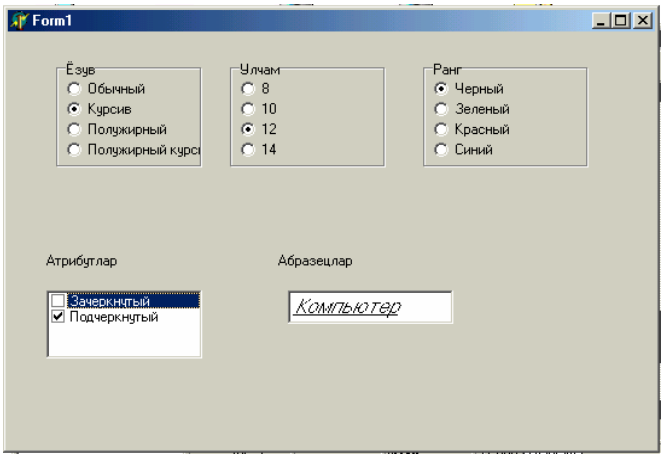
10. CheckListBox komponentasi maydonini ikki marta tez bosib kiritish maydoniga qayidagi kodlarni kiritamiz.

```
If CheckListBox1.Checked[0]
    Then Edit1.Font.Style:=Edit1.Font.Style+[FsStrikeOut]
```

Else Edit1.Font.Style:=Edit1.Font.Style-[FsStrikeOut];
If CheckListBox1.Checked[1]
Then Edit1.Font.Style:=Edit1.Font.Style+[FsUnderLine]
Else Edit1.Font.Style:=Edit1.Font.Style-[FsUnderline];

11.Tuzilgan loyihani saqlaymiz va uni F9 tugmachasini bosish bilan ishga tushuramiz.

Ilova ishga tushirilganda uning quyidagi ko`rinishi ekranda namoyon bo`ladi. Endi ilova bilan ishlash mumkin.



Savollar

1.Test tuzish dasturini yaratishdan tashqari yana qanday masalalarni yechishda tanlash tugmalarini ishlatish mumkin?

2.RadioGroup va CheckListBox komponentalari farqi nimada?

3.RadioGroup va CheckListBox komponentalari asosiy xossasi nima?

4.RadioGroup va CheckListBox komponentalari tugmalari nomlarini qanday kiritish mumkin?

12.MULOQOT OYNALARINI YARATISH

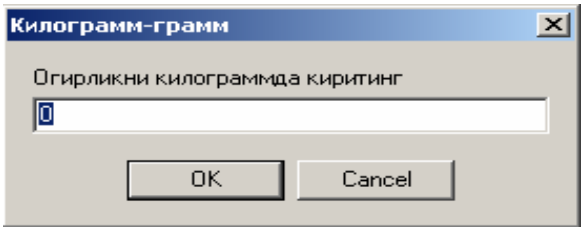
Malumki Windows tizimi bir qancha standart muloqot oynalariga ega. Bu oynalar misoliga fayllarni ochish va saqlash, shriftlarni tanlash va to`g`rilash, rang berishlarni keltirish mumkin. Delphi ham bu muloqot oynalarini ishlatadi. Bu oynalarni ishlatish uchun Delьfida maxsus komponentalar va usullar mavjud..

12.1.Muloqotli kiritish oynasi

Kiritish oynasi -InputBox funktsiyasini chaqirish natijasida ekranga chiqariladi. Bu funktsiya o`zgaruvchilar qiymatlarini kiritishda Pascaldagi Read operatori vazifasini o`taydi. InputBox funktsiyasi qiymati - foydalanuvchi kiritgan qatordir. U umumiy holda quyidagi ko`rinishga ega:

O`zgaruvchi := InputBox(Sarlavha, Izoh, Qiymat);

Uning dialog oynasi ko`rinishi quyidagicha:



Bu oyna dasturda quyidagicha berilgan:

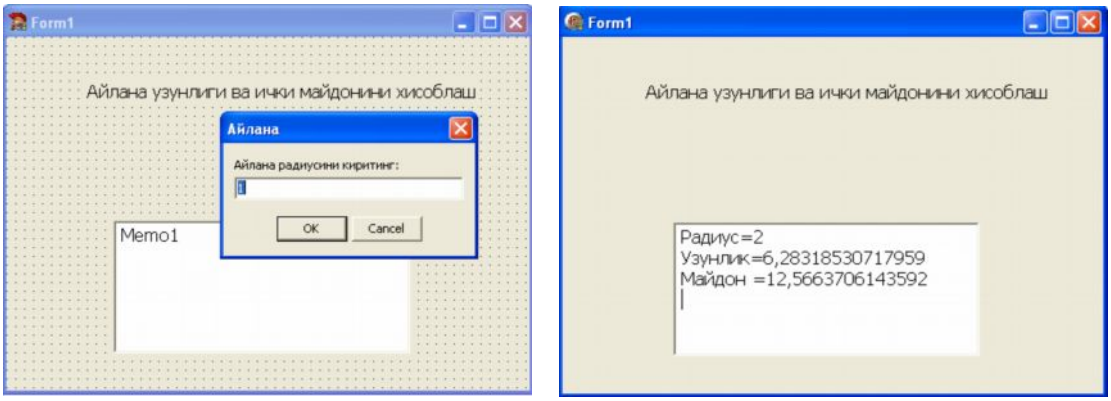
```
s:=InputBox('Kilogramm-gramm','Og`irlikni kilogrammda kiriting`,'0');
```

Misol. Aylana radiusi *R* beilgan. Aylana uzunligini *L* va uning ichki maydoni *S* ni topish ilovasini yarating.

Bajarish: Formaga Label1 belgisi qo`yilib, uning Caption xossasiga “Aylana uzunligi va ichki maydonini xisoblash” so`zi o`rnatiladi. Natijani Memo1 komponentasiga chiqarish uchun formaga o`rnatiladi va forma ikki marta chertilib quyidagi dastur kiritiladi:

```
procedure TForm1.FormCreate(Sender: TObject);
 Var R,L,S: Real;
begin
 R:=StrToFloat(InputBox('Aylana','Aylana radiusini kiriting:','1'));
   L:=Pi*R;  S:=Pi*R*R; Memo1.Clear;
   Memo1.Lines.Add('Radius='+FloatToStr(R));
   Memo1.Lines.Add('Uzunlik='+FloatToStr(L));
   Memo1.Lines.Add('Maydon ='+FloatToStr(S));
end;
```

Ilova ishga tushganda quyidagi muloqot oynasi chiqadi:

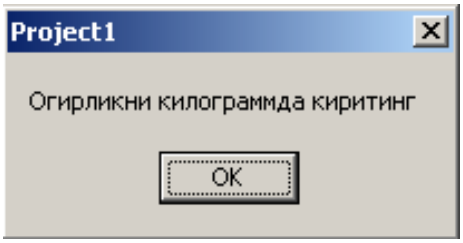


12.2.Ma’lumot oynasini chiqarish

ShowMessage protsedurasi ekranga matnli hamda OK buyrq tugmasiga ega bo`lgan ma’lumot oynasini chiqaradi. U quyidagi ko`rinishga ega:

ShowMessage(Ma’lumot);

Rasmda ma’lumot oynasi ko`rsatilgan:



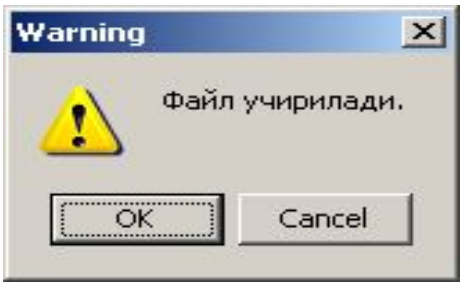
Uning dasturda yozilishi quyidagicha:

ShowMessage('Ogirlikni kilogrammda kiriting`);

MessageDig funktsiyasi ma’lumotli oynaga standart belgilar, masalan "Vnimanie", buyruq tugmalari sonini va turini berishga, hamda foydalanuvchi qaysi tugmani bosganligini aniqlashga imkon beradi.

Tanlov:=MessageDlg(Ma’lumot,Tur,Tugmalar, KontekstSpravka);

Rasmda ma’lumot oynasi ko`rsatilgan:



Uning dasturda yozilishi quyidagicha:

r:=MessageDlg('Fayl uchiriladi.', mtWarning, [mbOk,mbCancel],0);

Tanlov: = MessageDlg (Ma’lumot, Tur, Tugmalar, KontekstSpravkalar);

- Маълумот - маълумот матни;
- Тур - маълумот тури. Маълумот тури номланган константа билан берилади.

MessageDlg функцияси константалари:

Константа	Маълумот типи	Белги
mtWarning	Диққат	
mtError	Хато	
mt Information	Маълумот	
mtConfirmation	Тасдиқлаш	
mtCustom	Оддий	Белгисиз

- Тугмалар — маълумот ойнасида акс этувчи тугмалар рўйхати. MessageDlg функцияси константалари:

Константа	Тугма	Константа	Тугма
mb Yes	Yes	Mb Abort	Abort
mbNo	No	mbRetry	Retry
mbOK	OK	Mblgnore	Ignore
mbCancel	Cancel	mbAll	All
mbHelp	Help		

Kontekst Spravkalar - foydalanuvchi <F1> tugmasini bosganda ekranda paydo bo`luvchi spravka tizimining bo`limidir.

12.3.OpenDialog komponentasi

OpenDialog komponentasi kompyuter fayl tizimini ko`rish va undan kerakli fayl nomini tanlash imkonini beradi.

Bu komponenta piktogrammasi ko`rinishga  ega va u vizual bo`lmagan komponentadir.

Uning asosiy xossalari:

1.DefaultExt -faylning kengaytma nomini saqlaydi.

2.FileName -tanlangan fayl nomini saqlaydi.

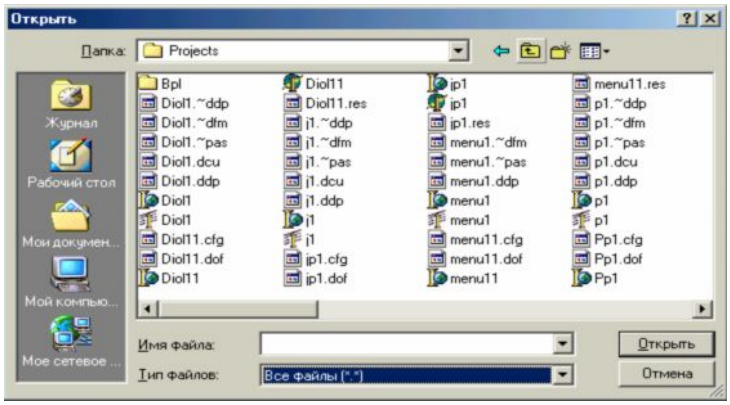
3.Filter -fayl nomlarini muloqot darchasiga ko`rsatilgan kengaytma nom bo`yicha filьtrlab chiqaradi. Filter xossasiga o`tilib uch nuqtali tugmacha bosilsa Filter Editor muloqot oynasi chiqadi. U ikki qismdan iborat bo`lib,

birinchi qismda filьtr matni ikkinchi qismida esa filьtrning o`zi beriladi.

Masalan: *.pas

*.txt; *.doc

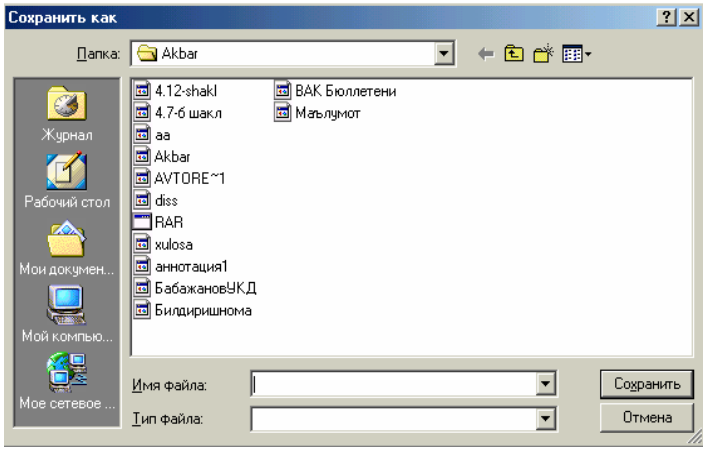
.



12.4.SaveDialog komponentasi

SaveDialog komponentasi kompyuter xotirasiga fayllarni saqlash imkonini beradi.

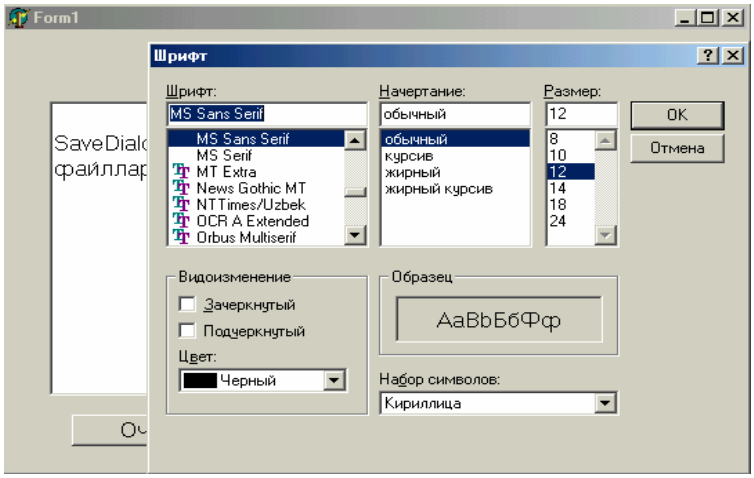
Bu komponenta piktogrammasi  ko`rinishga ega. Uni formaga sichqonchada bir marta bosib qo`yiladi va uning xossalari o`rnatiladi. DefaultExt xossasi qiymati .txt qilib tenglashtirilsa, faylni saqlashda avtomatik ravishda uning kengaytmasi .txt qilinib saqlanadi.



12.5.FontDialog komponentasi

FontDialog komponentasi foydalanuvchiga shriftlarni tanlaydi va uning xarakteristikasini belgilaydi.

Bu komponenta piktogrammasi  ko`rinishga ega. U vizual bo`lmagan komponenta bo`lib, uni formaga sichqonchada bir marta bosib qo`yiladi va keyin uning xossalari o`rnatiladi. Uning Font xossasi shrift xarakteristikasini beradi.



Misol: OpenDialog, SaveDialog va FontDialog komponentalarini ishlatgan holda oddiy matn muharririni yarating.

Bajarish tartibi:

1.Formaga Memo komponentasini o`rnatamiz.

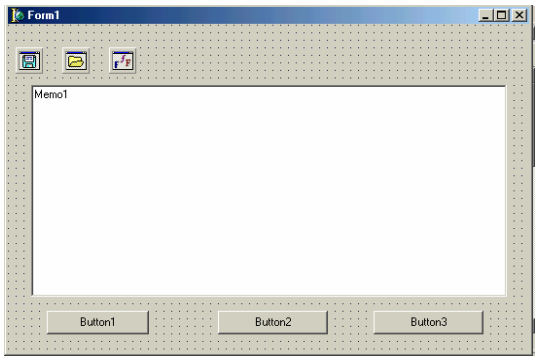
2.Formaga OpenDialog, SaveDialog va FontDialog koiponentalarini o`rnatamiz.

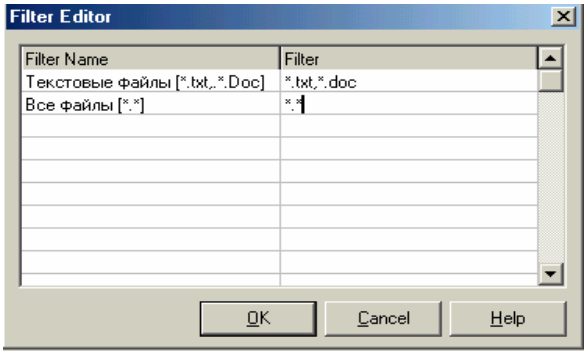
3.Formaning pastki qismiga Botton komponentasini uch marta Botton1, Botton2 va Botton3 nomlar bilan o`rnatamiz.

4.Memo1 komponentasining Lines xossasiga kelib, uch nuqtali ugmachani bosamiz va muloqotoynasidan Memo1 so`zini o`chirib va Ok tugmasini bosamiz.

5.OpenDialog komponentasi xossalarini o`rnatamiz. Buning uchun Filter xossasiga kirib, muloqot darchasiga quyidagilarni kiritamiz va Ok tugmasini bosamiz.

Filter Name qismiga
Tekstovыe dokumentы (*.txt,*.doc)
Vse faylы (*.*)
Filter qismiga
*.txt; *.doc
.





6.SaveDialog komponentasi DefaultExt xossasi qiymatini .txt qilib o`rnatamiz.

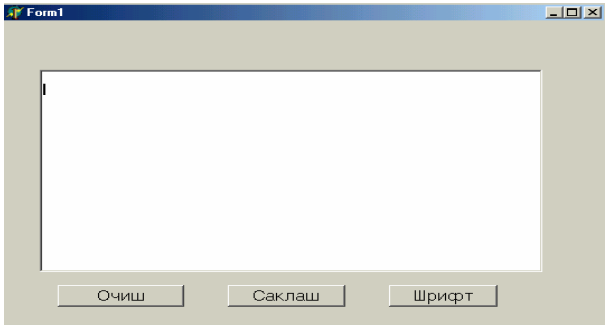
7.Botton1, Botton2 va Botton3 tugmachalar nomlarini Caption xossasiga kirib, mos ravishda “Ochish”, “Saqlash” va “SHrift” nomlariga o`zgartiramiz.

8.Botton1 tugmasi uchun quyidagi dastur kodlarini kiritamiz.

```
With Opendialog1 do
 Begin
  If not Execute then Exit;
```

```
    Memo1.Lines.LoadFromFile(Filename);
  End;
```
9.Botton2 tugmasi uchun quyidagi dastur kodlarini kiritamiz.
```
  With Savedialog1 do
  Begin
   If not Execute then Exit;
    Memo1.Lines.SaveToFile(Filename);
  End;
```
10.Botton3 tugmasi uchun quyidagi dastur kodlarini kiritamiz.
```
  With Fontdialog1 do
  Begin
   If not Execute then Exit;
    Memo1.Font:=Font;
  End;
```
11.Tuzilgan loyihani saqlanadi va ishga tushuriladi.

Ilova ishga tushirilgandagi ko`rinishi:



12.6.RichEdit komponentasi

RechEdit komponentasi ham xuddi Memo komponentasi kabi ko`p qatorli matnlarni tahrirlash uchun maxsus taxrirlash oynasidan foydalaniladi. RechEdit va Memo komponentalarida ham matndan nusxa olish Ctrl-C (Copy), qirqib olish Ctrl-X (Cut), qo`yish Ctrl-V (paste) imkoniyatlari mavjud.

RechEdit komponentasi Win32 komponentalar palitrasida joylashgan. RechEdit komponentasi orqali RTF va oddiy TXT kengaytmali fayllari bilan ishlash mumkin. Bu komponentanining ayrim xossalari va usullarini ko`rib chiqamiz.

Paragraph xosasi orqali matnlarni tekslash mumkin. Uning qiymatlarini faqat dastur ishlash darayoni uchun o`rnatish mumkin.Unga bog`liq Alignment usuli quyidagi qiymatlarni qabul qilishi mumkin:

```
RechEdit1.Paragraph.Alignment:=taLeftJustify;  //chapga tekislash
RechEdit1.Paragraph.Alignment:=taCenter;       //o`rtaga tekislash
RechEdit1.Paragraph.Alignment:=taRightJustify; //o`nga tekislash
```

Bu komponentaning asosiy xossalaridan yana biri **Lines** bo`lib u qatorli tipli matn ma'lumotlarini saqlaydi. Matn oxiriga yangi qator qo`shish uchun u Add yoki Append usulidan foydalanadi. Biror faylni o`qish uchun LoadFromFile usuliga, saqlash uchun esa SaveToFile usuliga murojaat qilinadi.

Masalan

```
RechEdit1.Clear;                                  //oynani tozalash
RechEdit1.Lines.Add('Assalomu alaykum');          //matnni chiqarish
RechEdit1.Lines.LoadFromFile('Referat.txt');      //faylni o`qish
RechEdit1.Lines.SaveToFile('Referat.txt');        //faylni yozish
```

Savollar

1.Kiritish oynasi -InputBox funktsiyasi dasturda qanday ko`rinishda yoziladi?
2.Kiritish oynasida nimalar aks etadi?
3.Ma'lumot oynasini chiqarish dasturda qanday ko`rinishda yoziladi?
4.Tanlov – MessageDlg funktsiyasi qanday o`zgarmaslarga ega?
5.SaveDialog komponentasi qanday vazifani bajaradi?
6.OpenDialog komponentasi qanday vazifani bajaradi?
7.FontDialog komponentasi qanday vazifani bajaradi?
8.RechEdit komponentasi qanday vazifani bajaradi?

13.RO`YXAT VA MASSIVLAR BILAN ISHLASH KOMPONENTALARI

Ko`p hollarda ro`yxat va massiv ko`rinishidagi ma'lumotlar bilan ish yuritishga to`g`ri keladi. Ma'lumki massiv - bu ma'lumotlar strukturasi bo`lib, uni bitta umumiy nomga ega bo`lgan bir turdagi o`zgaruvchilar to`plami deb qarash mumkin. Massivlarni bir turdagi ma'lumotlarni saqlashda ishlatish juda qulay. Masalan, jadval elementlarini, tenglamalar tizimi koeffitsientlarini, matritsa va vektor elementlarini massiv sifatida ishlatish mumkin. Delьfida ro`yxat va massiv ko`rinishidagi ma'lumotlar bilan ish yuritishda maxsus ListBox, ComboBox va StringGrid jadval komponentalari mavjud bo`lib ular yordamida bunday ko`rinishdagi ma'lumotlarni kiritish, chiqarish va qayta ishlash mumkin.

13.1.ListBox komponentasi

ListBox komponenti ro`yxat va bir o`lchamli massiv ko`rinishdagi ma'lumotlarni ekranga aks ettirishda ishlatiladi. Ma'lumotlarni kiritishda esa Edit komponentasidan foydalaniladi. ListBox komponentasi Standart komponentalar palitrasida joylashgan. Uning xossalari quyidagilar:

Items -ro`yxat elementlarini aniqlaydi;
Sorter -ro`yxat elementlarini tartiblaydi;
Clear -ro`yxat elementlarini o`chiradi.

13.2.ComboBox komponentasi

ComboBox komponenti ro`yxat va bir o`lchamli massiv ko`rinishdagi ma'lumotlarni ekrandan kiritish uchun ishlatiladi. U ListBox va Edit komponentalarining birgalikdagi ishini bir o`zi bajaradi. Tashqi ko`rinishdan bu komponent oddiy Edit kiritish qatorini eslatadi. Uning o`ng qismida pastga belgisi bo`lib, kiritilayotgan ma'lumotlarni ko`rib borish mumkin. Bu komponenta Standart komponentalar palitrasida joylashgan va u quyidagi xossalarga ega:

- DropDownCount -ro`yxatdagi ekranga chiqadigan ma'lumotlar sonini aniqlaydi;
- Style -ro`yxatdagi ma'lumotning ko`rinishini tasvirlaydi;
- Text -ro`yxatdagi kiritilgan ma'lumot matn ekanini baldiradi.

Misol 1. Butun qiymatli A(10) massiv elementlari ichidan eng katta va eng kichiklari topilsin. Ilovada Listbox komponentasini ishlating.

1.Formaga Standart komponentalar palitrasidan ListBox1, Edit1, Memo1, ikkita Botton1 va Botton2 tugmalarini joylashtiramiz. Ular Caption xossalarini quyidagicha o`rnatamiz:

Edit1 qiymatiga bo`sh qator;

Botton1 va Botton2 tugmachalari xossalariga "Kiritish" va "Yechish" qiymat;

"Memo1" bo`sh qiymat;

2.Forma ustiga sichqonchada ikki marta bosib, kodlarni yozish oynasiga o`tamiz va quyidagilarni kiritamiz:

```
i:=0;
ListBox1.Clear;
```

Inteface bo`limiga massiv va ishlatiladigan o`zgaruvchilarni Var so`zidan keyin tavsiflaymiz.

```
a:Array[1..10] of integer;
k,i,maxx,minn: Integer;
s1,s2: String;
```

3."Kiritish" tugmasini aktivlashtirish uchun uni ikki marta tez-tez bosib, kodlarni yozish oynasidan quyidagilarni kiritamiz:

```
ListBox1.Items.Add(Edit1.text);
i:=i+1;
a[i]:=StrToInt(Edit1.text);
Edit1.SetFocus;
```

4."Yechish" tugmasini aktivlashtirish uchun uni ikki marta tez-tez bosib, kodlarni yozish oynasidan quyidagilarni kiritamiz:

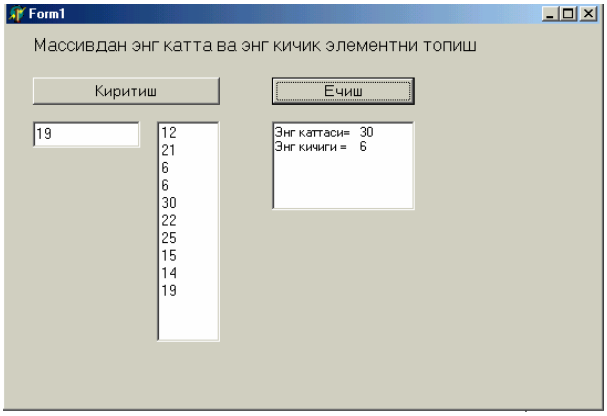
```
minn:=a[1];  maxx:=a[1];
 For k:=1 to 10 do
  Begin
    If minn>a[k] Then Minn:=a[k];If maxx<a[k] Then maxx:=a[k];
   End;
Str(maxx:5,S1);    Str(minn:5,S2);    Memo1.Clear;
Memo1.Lines.Add('Eng kattasi='+s1);
Memo1.Lines.Add('Eng kichigi ='+s2);
```

5.Kiritish fokusi Edit1 kiritish qatorida turishi uchun uni ikki marta bosib kodlarni yozish oynasidan quyidagini kiritamiz:

If key=13 Then Button1.SetFocus;

6.Tuzilgan loyihani saqlaymiz va uni ishga tushuramiz.

Ilova ishga tushirilganda uning quyidagi ko`rinishi ekranda namayon bo`ladi. Unga ketma ket kiritish tugmasini bosib qiymatlar kiritiladi va hisoblash uchun Yechish tugmasi bosiladi.



M i s o l 2. Butun qiymatli A(10) massiv elementlarining eng katta va eng kichiklari topilsin. Ilovada ComboBox komponentasini ishlating.

E ch i sh

1.Formaga Standart komponentalar palitrasidan ComboBox1, Memo1, ikkita Botton1 va Botton2 tugmalarini o`rnatamiz.

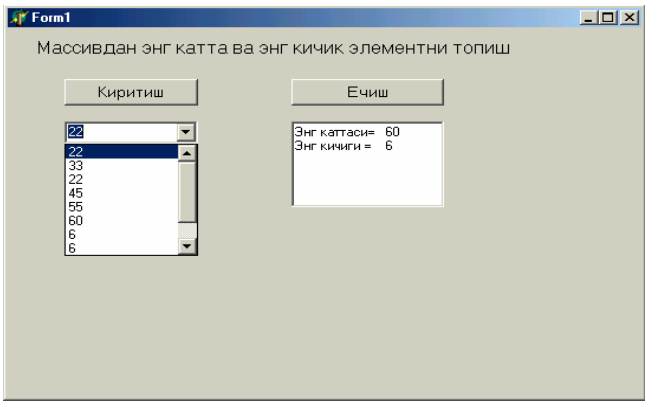
2.Oldingi misol kabi bu komponentalarning xossalarini ham o`rnatamiz va “Kiritish” tugmasiga bog`liq dastur kodlarini ham kiritamiz.

```
ComboBox1.Items.Add(Combobox1.text);
 i:=i+1;
 a[i]:=StrToInt(ComboBox1.text);
 ComboBox1.SetFocus;
```

3.Tuzilgan loyihani saqlaymiz.

4.Ilovani ishga tushuramiz.

Ilova ishga tushirib “Yechish” tugmasini bosganda quyidagi ko`rinishda bo`ladi.



13.3.StringGrid jadval komponentasi

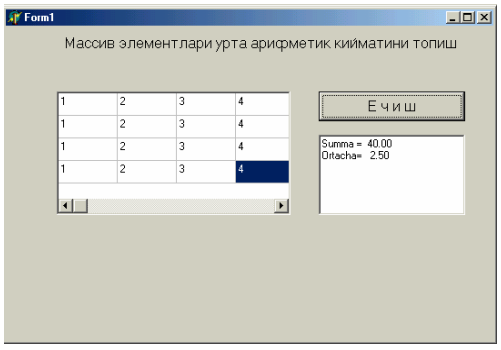
StringGrid jadval komponentasi ikki o`lchovli ma'lumotlarni, masalan matritsa elementlari qiymatini ekranda jadval ko`rinishda tasvirlash, ular qiymatini kiritish va tahrirlash uchun ishlatiladi. StringGrid jadval komponentasining asosiy xossalari quyidagilar:

ColCount -jadvaldagi ustunlar sonini aniqlaydi;
RowCount -jadvaldagi satrlar sonini aniqlaydi;
FixedCols -fiksirlangan ustunlar sonini aniqlaydi;
FixedRows -fiksirlangan satrlar sonini aniqlaydi;
Options -jadvad holatini aniqlaydi;
ColWidths -ustun kengligini aniqlaydi;
DefaultColWidth -ustunlar boshlang`ich kengligini aniqlaydi;
DefaultRowHeight -satrining boshlang`ich balandligini aniqlaydi;
FixedColor -fiksirlangan yacheyka rangini aniqlaydi;
RowHeights -jadval satri balandligini aniqlaydi;
Cells -simvol qatorli ikki o`lchamli massivni aniqlaydi

Misol. Butun qiymatli A(4,4) massiv elementlari yig`indisi va o`rta armfmetik qiymati topilsin.

1.Formaga StrinGrid, Memo va Botton tugmalarini o`rnatamiz.
2.StrinGrid komponentasining xossalarini o`rnatamiz:
FixedCols -0, FixedRows -0, ColCount -4, RowCount -4.
3.Option xossasining GoEditing parametrini True qiymatga tenglashtiramiz.
4.Botton1 tugmasining Coption xossasiga kirib uning nomini "Yechish" nomiga o`zgartiramiz.
5."Yechish" tugmasiga dastur kodlarini kiritamiz.

```
Var i,j,cod:integer;
    A:array[1..4,1..4] of Real; S:real; s1:String;
begin
  For i:=1 to 4 do  For j:=1 to 4 do A[I,j]:=StringGrid1.cells[i-1,j-1];
     S:=0;
    For i:=1 to 4 do For j:=1 to 4 do s:=s+a[i,j];
     S1:=FloatToStr(s);  Memo1.Clear;
      Memo1.Lines.add('Summa ='+s1);
      s:=s/4/4; S1:=FloatToStr(s);
      Memo1.Lines.add('Ortacha='+s1);
end;
```



Tashkil qilingan forma va modulning to`liq ko`rinishi.

```
unit j1;
interface
uses  Windows, Messages, SysUtils,Variants, Classes, Graphics,
        Controls, Forms,Dialogs, Grids, StdCtrls;
type  TForm1 = class(TForm)
        StringGrid1: TStringGrid;
        Button1: TButton;
        Label1: TLabel;
        Memo1: TMemo;
   procedure Button1Click(Sender: TObject);
   private    { Private declarations }
   public    { Public declarations }
 end;
var  Form1: TForm1;
implementation
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
  Var i,j,cod:integer; A:array[1..4,1..4] of Real; S:real; s1:String;
  begin
    For i:=1 to 4 do  For j:=1 to 4 do A[I,j]:=StringGrid1.cells[i-1,j-1];
      S:=0;
      For i:=1 to 4 do For j:=1 to 4 do s:=s+a[i,j];
       S1:=FloatToStr(s);  Memo1.Clear; Memo1.Lines.add('Summa ='+s1);
        s:=s/4/4; S1:=FloatToStr(s);  Memo1.Lines.add('Ortacha='+s1);
  end;
end.
```

Savollar

1.Massivlarda qanday turdagi ma’lumotlarni saqlash mumki?
2. ListBox komponentsi nima ish bajaradi va uning asosiy xossalari nima?
3.ComboBox komponentsi vazifmsi va uning asosiy xossalari nima?
4.StringGrid komponentsi qanday ma’lumotlarni qayta ishlashda qo`l keladi?
5.StringGrid jadval komponentasining asosiy xossalarini aytib bering.

14.DELPHI GRAFIK IMKONIYATLARI

Delphi dasturchiga grafiklar, sxemalar, chizmalar va illyustratsiyalar yaratishga imkon beradi. Dastur grafikani forma yoki Image komponentasi yuzasiga chiqaradi. Ob'ekt yuzasiga canvas xossasi moc keladi. Ob'ekt yuzasiga grafik element (to`g`ri chiziq, aylana, turtburchak va hokazo), chiqarish uchun bu ob'ektning canvas xossasiga mos usul qo`llash lozim. CHizish sohasi alohida nuqtalar - piksellardan iborat. Piksel holati uning gorizontal (X) va vertikal (Y) koordinatalari bilan aniqlanadi. CHap yuqori piksel koordinatalari (0,0). Koordinatalar yuqoridan pastga va chapdan o`nga qarab o`sib boradi.

Soha o`lchovlarini image komponentasining Height i width xossalari va formaning ClientHeight va Clientwidth xossalari orqali aniqlash mumkin.

14.1.CHizish va bo`yash xossalari

QALAM. CHiziq ko`rinishi Tren ob'ekti xossalari orqali aniqlanadi.

Tren (qalam) xossalari: Color - chiziq rangi; Width - chiziq qalinligi; Style - chiziq ko`rinishi; Mode - akslantirish rejimi.

Color xossasi qiymatlari: Black-Qora; clSilver-Serebristыy; clMaroon-Kashtanovыy; clRed-Qizil; clGreen-YAshil; clLime-Salatnыy; clOlive-Olivkovыy; clBlue-Ko`k; clNavy-Tim-ko`k; clFuchsia-YArko-rozovыy; clPurple-Rozovыy; clAqua-Biryuzovыy; clTeal-Zeleno-goluboy; clWhite-Oq; clGray-Kul rang.

CHiziq qalinligi width xossasi orqali piksellarda beriladi.

Style xossasi kiymatlari: psSolid-Uzluksiz chiziq; psDash-Punktir chiziq uzun shtrixlar; psDot-Punktir chiziq, qisqa shtrixlar; psDashDot-Punktir chiziq, uzun va qisqa shtrixlar ketma ketligi; psDashDotDot-Punktir chiziq, bitta uzun va ikkita qisqa shtrixlar ketma ketligi; psClear-CHiziq aks etmaydi.

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

Mode xossasi qiymatlari: pmBlack-Qora, Pen.Color xossasi qiymatiga bog`liq emas; pmWhite-Ok, Pen.Color xossasi qiymatiga bog`liq emas; pmCopy-CHiziq rangi Pen. Color xossasi qiymatiga bog`liq; pmNotCopy-CHiziq rangi Pen.Color xossasi qiymatiga invers; pmNot-CHiziq rangi sohaning mos nuqtasi rangiga invers.

MUYQALAM (Canvas.Brush) yopiq sohalarni chizish va soha ichini bo`yash uchun ishlatiladi. Muyqalam ikki xossaga ega:

Color -Epiq sohani bo`yash rangi;

Style - Sohani to`ldirish uslubi.

Kontur ichidagi soha bo`yalishi yoki shtrixlanishi mumkin. Sohani to`ldirish usulini belgilovchi konstantalar Brush.style xossasi qiymatlaridir.

Brush.style xossasi qiymatlari:

BsSolid-Uzluksiz bo`yash; bsClear-Soha bo`yalmaydi; bsHorizontal-Gorizontal shtrixlash; bsVertical-Vertikal shtrixlash; bsFDiagonal-Diagonal shtrixlash, oldinga

og`ish; bsBDiagonal-Diagonal shtrixlash, orqaga og`ish; bsCross-Katakli gorizontal-vertikal shtrixlash; bsDiagCross-Katakli diagonal shtrixlash.

Masalan:

```
Canvas.Brush.Color := clGreen;
Canvas.Brush.Style := bsSolid;
Canvas.Rectangle (10,10,30,30) ;
```

Bunda to`rtburchak soha uzluksiz yashil ranga bo`yaladi.

14.2.Grafik primitivlarni chizish usullari

CHiziq. To`g`ri chiziq LineTo usuli orqali amalga oshiriladi.

Komponent.Canvas.LineTo(x,u);

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

Komponent.Canvas.MoveTo(x,u);

Misol. Image1.Canvas.MoveTo(10,10); Image1.Canvas.LineTo(20,20);

Bu misolda berilgan usullar Image oynasining (10,10) koordina-tasidan (20,20) sigacha bo`lgan to`g`ri chiziqni chizib beradi.

Tutashgan chiziq. O`zaro tutashgan kesmalardan iborat shaklni chizish uchun **Polyline** usulidan foydalaniladi. Bu usul parametri TPoint tipli massivdan iborat. Polyline usuliga misol tariqasida ma’lum qiymat o`zgarishi grafigini chizuvchi protseduracini keltiramiz:

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

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

Aylana va ellips. Aylana yoki ellips chizish uchun Ellipse usuli foydalaniladi:
Ob’ekt.Canvas.Ellipse(x1,y1, x2,u2);
Bu erda x1,y1,x2,u2 -ellipsni o`z ichiga olgan minimal turtburchak koordinatalari. Agar turtburchak kvadrat bo`lsa aylana chiziladi.

Ey. YOyni chizish uchun Arc usuli qo`llaniladi:

Ob'ekt.Canvas.Arc(x1,y1,x2,u2,x3,u3,x4,u4);

Bu erda x1,y1,x2,u2 -yoyga tegishli bo`lgan ellips yoki aylana parametrlari; x3,u3 -yoy boshlang`ich nuktasi parametrlari; x4, u4 - so`ngi nuqtasi parametrlari. YOy soat miliga teskari tartibda chiziladi.

**To`rtburchak.** To`rtburchak chizish uchun Rectangle usulidan foydalaniladi:

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

Bu erda x1,y1,x2,u2 -chapgi yuqori va o`nggi pastgi burchaklar koordinatalari.
RoundRec usuli burchaklari yumaloq to`rtburchak chizishga imkon beradi:

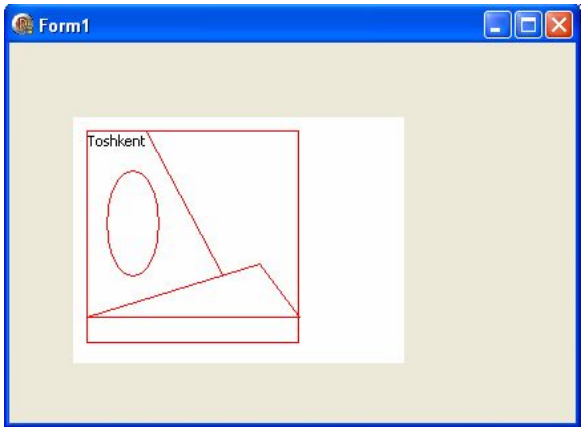
Ob'ekt.Canvas.RoundRec(x1,y1,x2, u2, x3, u3);

Bu erda x1,y1,x2,u2 -turtburchak parametrlari; x3,u3 -chorak kismi yumaloq burchak chizish uchun ishlatildadigan ellips kattaligi.
YA'na ikki usul muyqalamdan foydalanib to`rtburchak chizishga imkon beradi. FillRect usuli ichi bo`yalgan to`rtburchak chizadi, FrameRect - faqat kontur. Bu usullarda faqat bitta parametrga ega -TRect tipidagi struktura.

**Ko`pburchak.** Polygon usuli ko`pburchak chizishga mo`ljallangan bo`lib, parametri TPoint tipidagi massivdir. Quyida Polygon usuli yordamida uchburchak chizish protsedurasi keltirilgan:

```
 procedure TForm1.Button2Click(Sender: TObject);
  Var pol: array[1..3] of TPoint;
  begin
   pol[1].x := 10; polf1].y := 50; pol[2].x := 40; pol[2].y := 10;
   pol[3].x := 70; pol[3].u := 50;
   Form1.Canvas.Polygon(pol);
  end;
procedure TForm1.FormCreate(Sender: TObject);
  var pol: array[1..3] of TPoint;
Begin
  Image1.Canvas.Pen.Color := clRed;            {Chiziqqa rang berish}
  Image1.Canvas.Rectangle (10,10,170,170);   {tortburchak}
  Image1.Canvas.TextOut(11, 11, 'Toshkent'); {Tekst}
  Image1.Canvas.LineTo(113,120);                 {chiziq}
  Image1.Canvas.Ellipse(25,40, 65,120);        {Ellips}
  {kopburchak}
  pol[1].x := 10;  pol[1].y := 150;
  pol[2].x := 140;  pol[2].y := 110;
  pol[3].x := 170;  pol[3].y := 150;
  Image1.Canvas.Polygon(pol);
end;
```

Dastur natijasi rasmda berilgan.



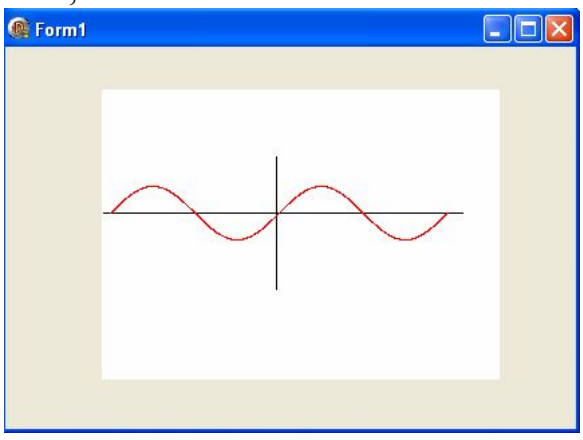
Sektor. Ellips yoki aylana sektori pie usuli bilan chizilib, chaqirish instruktsiyasi quyidagi umumiy ko`rinishga ega:

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

Bu erda: x1,y1,x2,u2 -ellips yoki aylana parametrlari; x3,u3,x4,u4 -sektor chegarasini tashkil qiluvchi to`g`ri chiziqlar oxirgi nuktalari koordinatalari.

Nuqta. Canvas ob'ektining pixels xossasi tipidagi ikki o`lchovli massiv bo`lib har bir soha nuqtasining rangi haqidagi ma'lumotni o`z ichiga oladi. Pixels xossasidan foydalanib ixtiyoriy nuqta rangini o`zgartirish, ya'ni nuqta chizish mumkin. Misol uchun
Form1.Canvas.Pixels[10,10]:=clRed;
instruktsiyasi soha nuqtasini qizil ranga bo`yaydi.
Quyidagi dastur y=sinx funktsiyasi grafigini [-2pi,2pi] oraliqda chizadi.

```
Procedure TForm1.FormCreate(Sender: TObject);
  Const Pi=3.1415;    Var x,y,mas:Real; x1,y1:integer;
 begin
   Image1.Canvas.MoveTo(1,92);       Image1.Canvas.LineTo(270,92);
    Image1.Canvas.MoveTo(130,50);    Image1.Canvas.LineTo(130,150);
         x:=-2*pi; mas:=20;
  Repeat
    y:=Sin(x);
    X1:=132-Round(mas*x);     Y1:=92+Round(mas*y);
    Image1.Canvas.Pixels[x1,y1]:=ClRed;
    x:=x+0.01;
  until x>2*pi;
 end;
```



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

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

Matn shrifti Font xossasi qiymati bilan aniqlanadi. Font xossasi TFont tipidagi ob'ekdir. Quyidagi jadvalda TFont ob'ekti xossalari keltirilgan.

TFont ob'ekti xossalari

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

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

Misol:

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

Textout uslubi orqali matn ekranga chiqarilgandan so`ng qalam matn chiqarish sohasining yuqori o`ng burchagiga keltiriladi. Agar matn uzunligi ma'lum bo`lmasa, chiqarilgan matn o`ng chegarasi koordinatalarini PenPos xossasiga murojaat qilib aniqlash mumkin.

Misol:

```
with Form1.Canvas do begin
TextOut(0, 10, 'Borland ') ;
TextOut(PenPos.X, PenPos.Y, 'Delphi 6');
end;
```

14.3.Grafik komponentalar

Image komponentasi formaga rasmlarni joylashtirish uchun ishlatiladi. Joylashtirilishi lozim bo`lgan rasmlar bitli fayllar (kengaytmalari .Bmp), piktogrammali (kengaytmalari .Ico), metafayllar (kengaytmalari .wmf) bo`lishi kerak.

Image komponentasi Additional palitrasida joylashgan bo`lib, u  ko`rinishdagi piktogrammaga ega. Bu tugmachani bosib formadan rasm uchun joy ajratiladi va keyin esa xossalar bo`limidan Picture xossasi tanlanib, u erdan uch nuqtali tugmacha bosiladi.

Natijada ekranda rasmni aniqlash va joylash uchun muloqat darchasi ochiladi. Muloqat darchasi quyidagi tugmachalarga ega:

Load -fayldan rasmni chaqirish;
Save -rasmni faylga saqlash;
Clear -tanlangan rasmni olib tashlash;
Ok -tanlangan rasmni ajratilgan joyga yozish;
Cancel -qilingan o`zgartirishlarni bekor qilish.

Shape komponentasi formaga aylana, to`rtburchak, ellips va boshqa shakllarni joylashtirish uchun ishlatiladi. Uning quyidagi xossalari mavjud:

Brush -shaklni bo`yash uchun cho`tkacha;
Pen -shakl chetini chizish uchun qalam;
Shape -ekranga chiqadigan shaklni aniqlaydi:
StRectangle -to`rtburchak;
StSquare -kvadrat;
StRoundRect -chetlari aylana to`rtburchak;
StRoundSquare -chetlari aylana kvadrat;
StEllipse -ellips;
StCircle -aylana.

Shape komponentasi ham Additional palitrasida joylashgan bo`lib, u  ko`rinishdagi piktogrammaga ega. Bu tugmachani bosib formadan shakl uchun joy ajratiladi va keyin esa xossalar bo`limidan Shape xossasiga kirilib kerakli shakl tanlanadi.

PaintBox komponentaci formaga chegaralangan maydonda shakllarni chizish imkonini beradi.

PaintBox komponentasi System palitrasida joylashgan bo`lib, u  ko`rinishdagi piktogrammaga ega. Bu tugmachani bosib formadan shakl uchun joy ajratiladi va keyin esa xossalar bo`limidan Shape xossasiga kirilib kerakli shakl tanlanadi.

Timer vizual bo`lmagan komponenta bo`lib, formada bajariladigan ma’lum bir operatsiyalarni vaqt bo`yicha boshqaradi.

Timer komponentasi System palitrasida joylashgan bo`lib, u  ko`rinishdagi piktogrammaga ega. Bu tugmachani bosib formaga olib kelib qo`yiladi.

U quyidagi xossalarga ega:

Enabled -true qiymati o`rnatilgan bo`lsa u bo`ladigan jarayonga ta’sir qiladi;

Interval -millisekundlarda vaqt intervalini aniqlaydi va jarayonning ekranga chiqishiga ta’sir ko`rsatadi. Tegmagan holda 1000 (1 sekund) ko`rsatadi.

M i s o l. Ilova uchun “zastavka” yaratish.

Zastavka grafik tasvirlar ko`rinishida bo`lib, programmalar ishga tushirilganda bir necha sekunddan so`ng ekranda paydo bo`ladi. Unda programma nomi va uning avtorlari haqida ma’lumot bo`lishi mumkin.

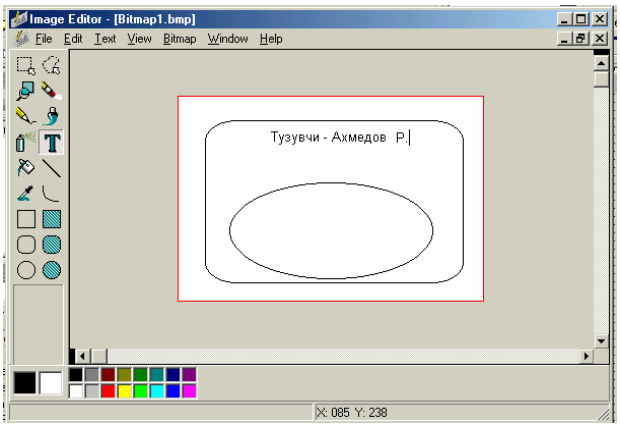
Grafik tasvirni, ya'ni .bmp kengaytmaga ega bo`lgan faylni grafik muharriri yordamida tayyorlaymiz. Delphi sistemasini ishga tushirishdan avval tuziladigan ilovani saqlash uchun o`zimizga papka tashkil qilamiz.

1.Oldin tuzilgan biror bir ilovani ochamiz yoki yangi ilova tashkil etamiz.

2.Bosh menyudan grafik muharrirni ishga tushiramiz: Tols=>Image Editor. (Bu Delphi grafik muxariri oddiy Paint grafik muharriridan uncha katta farq qilmaydi)

3.Delphi grafik muxariri Image Editor menyusidan File=>New=>Bitmap File(.bmp) buyrug`i beriladi. Natijada ekranda rasm paramertlarini berish uchun muloqat darchasi paydo bo`ladi. Muloqat darchasidan kerakli parametrlar tanlanib Ok tugmasi bosiladi. Tayyor mavjud rasm foyllaridan ham foydalashish mumkin.

4.Grafik muhariri oynasidan ajratilgan joyga ixtiyoriy rasm chizilib, u saqlanadi. Masalan, aylana va unga tashqi chizilgan rasm chizib, ichiga "Tuzuchi – Axmelov R." so`zi yozib qo`yilsin. Matnni yozish uchun uskunalar panelining "T" (Text) tugmachasidan foydalaniladi.



5.Grafik fayli saqlanadi va undan chiqiladi.

6.System palitrasidan Timer komponentasining tugmachani bosib formaga olib kelib qo`yiladi va u Timer1 nom oladi. Interval xossasini 3000 ga tenglashirib olamiz.

7.Additional palitrasidan Image komponentasi tugmachasini bosib formadan rasm uchun joy ajratiladi va keyin esa xossalar bo`limidan Picture xossasi tanlanib, u erdan uch nuqtali tugmacha bosiladi. Natijada ekranda rasmni aniqlash va joylash uchun muloqat darchasi ochiladi. Muloqat darchasidan Load buyrug`i berilib, saqlangan rasm faylimiz tanlanadi va Ok tugmasi bosiladi. Rasm to`liq formaga joylashishi uchun Autosize xossasiga True qiymatini o`rnatamiz.

8.Timer1 komponentini aktivlashtiramiz, ya'ni uni ikki marta tez-tez bosamiz va kodlarni yozish oynasiga quyidagi qora yozilgan kodlarni kiritamiz.

```
procedure TForm1.Timer1Timer(Sender: TObject);
  begin
        Image1.Free;
        Timer1.Free;
  end;
```

Bu shuni bildiradiki programma ishga tushgandan so`ng 3000 millisekunddan o`tishi bilan Image1 va Timer1 komponentalari kompyuter xotirasidan va mos ravishda ekrandan o`chiriladi.

9.Tuzilgan loyiha (proekt) ya’ni Project1 va Unit1 standart modul nomlarini mos nomlar bilan almashtirib saqlanadi.

10.YAngi nom bilan saqlangan proekt, ya’ni ilova F9 tugmachasini bosish bilan ishga tushuriladi.

Ilova ishga tushirilganda ekranda yuqoridagi 4 punktdagi rasm “zastavka” ko`rinishida namoyon bo`ladi.

Tashkil qilingan modulning to`liq ko`rinishini keltiramiz.

```
unit px1;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,  Forms, Dialogs,
   StdCtrls, ExtCtrls;
type
  TForm1 = class(TForm)
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
  implementation
 {$R *.dfm}
procedure TForm1.Timer1Timer(Sender: TObject);
 begin
   Image1.Free;
   Timer1.Free;
 End;
end.
```

YAratilgan “zastavka”miz programma ishlash davomida o`chib-yonib turishi uchun quyidagi OnTimer xodisasini qayta ishlash kodini yozishimiz kerak bo`ladi.

```
procedure TForm1.Timer1Timer(Sender: TObject);
 begin
    If Image1.visible=true then Image1.hide Else Image1.Show;
 End;
```

M i s o l 3. Oyning Er atrofida aylanishini namoyish etuvchi ilova yaratish.

1.YAngi ilova yaratamiz.

2.Formaga Timer komponentasini Timer1 nom Bilan joylashtiramiz.Uning Interval xossasini 55 qilib o`rnatamiz. Jarayon ya’ni xodisa55 millisekundda paydo bo`ladi (uyg`onadi).

3.Additional palitrasidan Shape komponentasini Shape1 nom bilan formaga joylashtiramiz va uning quyidagi xossalarini o`rnatamiz.

Shape -StCircle;
Height -121;
Width -121;
Left -240;
Top -104.

Brush xossasini tanlab ikki marta sichqonchada bosamiz, natijada ikkita yana qo`shimcha xossalar paydo bo`ladi: Color va Style. Color xossasini tanlab unga slBlue qiymatni o`rnatamiz.

4.Formaga ikkinchi Shape komponentasini Shape2 nom bilan joylashtiramiz va uning quyidagi xossalarini o`rnatamiz.

Shape -StCircle;
Height -41;
Width -41;
Left -400;
Top -152.

Brush xossasiga clYellow rangini o`rnatamiz.

5. Formaning yuqori qismiga Label komponentasini Label1 nom bilan joylashtiramiz va uning Caption xossasini “Oyning Er atrofida aylanishi” qiymatiga o`zgartiramiz. Font xossasiga kirib kerakli shriftni va uning o`lchamini aniqlaymiz (agar kerak bo`lsa). Masalan,

SHrift -Courier New,
Nachertanie -polujirnыy,
Razmer -16,
Nabor simvolov -krilitsa.
Transparent xossasi qiymatini True qilib o`rnatamiz.

6.Timer1 komponentini aktivlashtiramiz, ya’ni uni ikki marta tez-tez bosamiz va kodlarni yozish oynasiga quyidagi qora yozilgan kodlarni kiritamiz.

```
procedure TForm1.Timer1Timer(Sender: TObject);
  begin
        x:=x+0.1;
        Shape2.Left:=265+Trunc(50*Cos(x));
        Shape2.top:=150-Trunc(50*Sin(x));
  end;
```

7.Tuzilgan loyiha (proekt) ya’ni Project1 va Unit1 standart modul nomlarini mos nomlar bilan almashtirib saqlanadi.

8.YAngi nom bilan saqlangan proekt, ya’ni ilova F9 tugmachasini bosish bilan ishga tushuriladi.

Ilova ishga tushirilganda ekranda Oyning Er atrofida aylanishi namoyish qilinadi.

Tashkil qilingan modulning to`liq ko`rinishini keltiramiz.

```
unit px2;
interface
uses
```

```
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,  Forms, Dialogs,
  StdCtrls, ExtCtrls;
type
  TForm1 = class(TForm)
         Timer1: Ttimer;
         Shape1: TShape;
        Shape2: TShape;
         Label1:Tlabel;
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form1: TForm1;
  X: Real;
 implementation
 {$R *.dfm}

procedure TForm1.Timer1Timer(Sender: TObject);
 begin
   x:=x+0.1;
  Shape2.Left:=265+Trunc(50*Cos(x));
  Shape2.top:=150-Trunc(50*Sin(x));
 End;
Initialization
  x:=0
end.
```

Programmada o`zgaruvchi x global holda e'lon qilingan. SHu tufayli uning boshlang`ich qiymati Initialization sektsiyasida berilgan.

Savollar

1.Dastur grafikani qanday ob'ektlar yuzasiga chiqaradi?
2.Ob'ekt yuzasiga qanday xossa moc keladi?
3.Tren (qalam) ob'ektining qanday xossalari bor?
4.Muyqalam qanday xossalarga ega?
5.Grafik primitivlarni chizishning qanday usullarini bilasiz?
6.To`g`ri chiziq qanday usul orqali amalga oshiriladi?
7. To`rtburchak, aylana yoki ellips qanday usullar orqali amalga oshiriladi?
8.Grafik ob'ekt yuzasiga matn chiqarish qanday usul orqali amalga oshiriladi?

15.ILOVALAR UCHUN MENYU YARATISH VA BIR NECHA FORMALAR BILAN ISHLASH

Ko`pchilik ilovalar bosh menyuga ega bo`lib, bajariladigan operatsiyalar ro`yxatini o`z ichiga oladi. Bosh menyu punktlari nulinchi darajadagi menyu elementlari deyiladi. Ularning har biri bog`liq birinchi darajali menyu elementlarini o`z ichiga olishi mumkin. Delьfida ilovalar uchun kontekst menyularini ham yaratish mumkin. Ilovalarda kontekst menyusi sichqoncha o`ng tugmasini bosish bilan chaqiriladi va unga kerakli buyruqlarni joylashtirigsh mumkin.

Delьfida yaratiladigan dasturlarni ishlatitshda bir necha bog`liq formalardan foydalanishga to`g`ri keladi. Delьfi dastur ilovasini loyihalashtirayotganda formalarni bir biriga bog`lash imkonini yaratib beradi.

15.1.MainMenu komponentasi

Delphida bosh menyu tashkil qilish uchun maxsus vizual bo`lmagan MainMenu komponentasi mavjud. Bu komponenta Standart komponentalar palitrasida joylashgan.

MainMenu komponentasining asosiy xossasi Items xossasidir. U o`zida ilova bosh menyusining nulinchi darajali elementlarini saqlaydi.

Misol. File, Edit va Run strukturaga ega bosh menyu va ular tanlanganda mos punktlarga o`tish dastur ilovasini yarating.

Bajarish tartibi:

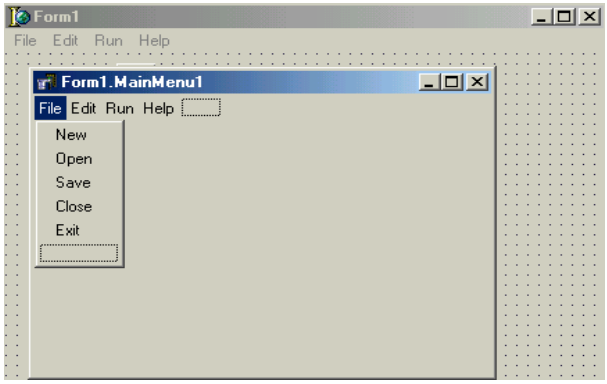
1.Formaga Standart komponentalar palitrasidan MainMenu komponentasini o`rnatamiz.

2.MainMenu1 komponentasi Items xossasini o`rnatamiz. Buning uchun Items xossasiga kirib, uning uch nuqtali tugmasini bosamiz. Natijada ekranda menyu konstruktori muloqot oynasi chiqadi.

3.Menyu konstruktori yordamida file, Edit, Run bosh menyu nomlarini va mos ularning buyruqlarini (New, Open, Save va boshqa elementlarini) kiritamiz. Buning uchun Caption xossasiga o`tilib va kerakli buyruq yoziladi.

4.Menyu konstruktor oynasidan chiqib
har bir Menyu punkti uchun ularning
har birini sichqonchada bir marta
chiqillatib kerakli dastur kodlarini
kiritamiz. Masalan:

```
procedure TForm1.New1Click(Sender:TObject);
  begin
   ShowMessage('New punkti');
  end;
```



15.2.PopupMenu komponentasi

Delphida kontekstli menyu tashkil qilish uchun maxsus vizual bo`lmagan PopupMenu komponentasi mavjud. Bu komponenta Standart komponentalar palitrasida joylashgan.

Misol. Open, Save va Font strukturaga ega kontekstli menyu yarating.

Bajarish tartibi:

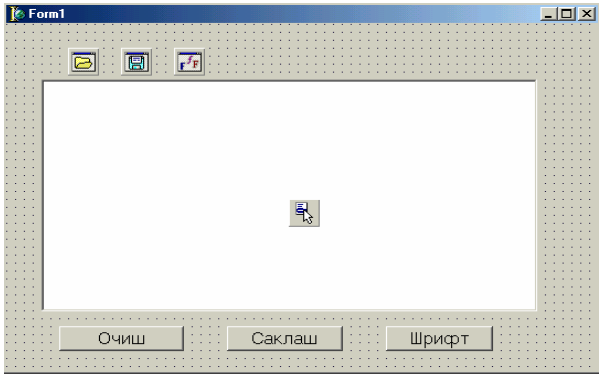
1.Formaga Standart komponentalar palitrasidan PopupMenu komponentasini o`rnatamiz.

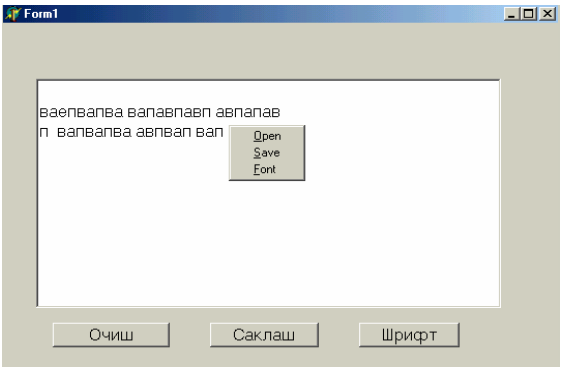
2. PopupMenu komponentasi Items xossasini o`rnatamiz. Buning uchun Items xossasiga kirib, uning uch nuqtali tugmasini bosamiz. Natijada ekranda menyu konstruktori muloqot oynasi chiqadi.

3.Menyu konstruktori yordamida Kontekst menyu elementlarini kiritamiz: Open, Save va Font.

4.Menyu konstruktor oynasidan chiqib har bir Menyu punkti uchun ularning har birini sichqonchada bir marta chiqillatib kerakli dastur kodlarini kiritamiz. Masalan:

```
procedure TForm1.Open1Click(Sender: TObject);
  begin
   Button1Click(Button1);
 end;
 procedure TForm1.Save1Click(Sender: TObject);
  begin
   Button2Click(Button2);
  end;
 procedure TForm1.Font1Click(Sender: TObject);
  begin
   Button3Click(Button3);
 end;
```





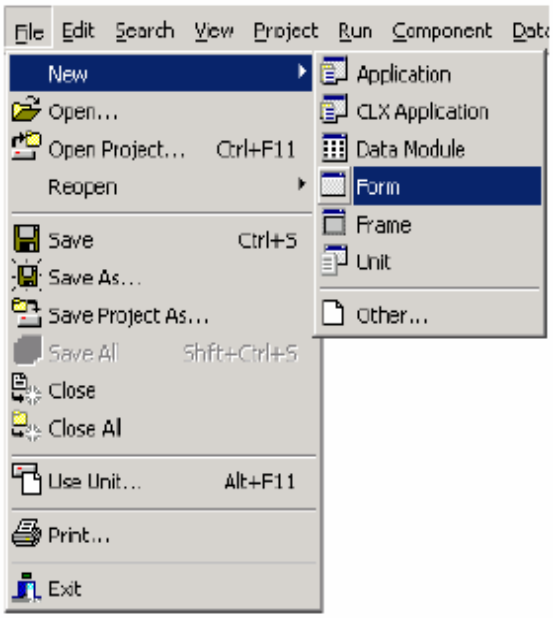
15.3.Bir necha formada ish yuritish

Ko`p hollarda bir necha formalar bilan ishlashga to`g`ri keladi. Buni misolda ko`rib chiqamiz. YAngi loyiha yaratib, formaga boshqarish tugmasi (Button1) va ilova komponentasi (Label1) joylashtiramiz. Boshqarish tugmasining Click hodisasiga quyidagi kodni kiritamiz:

```
procedure TForm1.Button1Click(Sender: TObject);
  begin
      Form2.Show;
      Label1.Caption:='Ko`p formali loyiha';
  end;
```

Endi loyihaga yangi forma qo`shamiz.

Buning uchun Menyu File bo`limidan New punktini, so`ngra Form punktini tanlaymiz. Loyihalar menejerini (View->Project Manager) ochib Project1.exe loyihamizda ikki forma *Unit1* i *Unit2* borligini ko`rishimiz mumkin. Formalarning biriga ikki marta chertilsa, tizim oynasida shu formaning o`zgartirish mumkin bo`ladi.

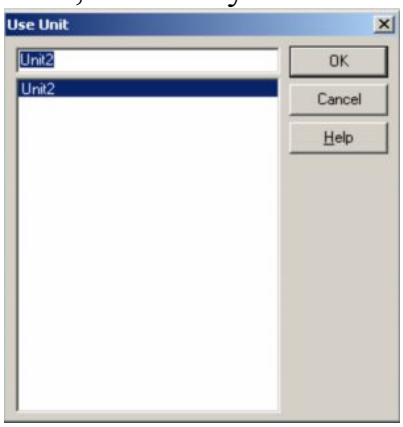


Ikkinchi formaga boshqarish tugmasi (Button1) va tahrirlash qatorini (Edit1) joylashtiramiz. Boshqarish tugmasining Click hodisasiga quyidagi kodni kiritamiz.

```
procedure TForm2.Button1Click(Sender: TObject);
```

```
begin
    Form1.Label1.Caption:=Edit1.Text;
    Form2.Close;
end;
```

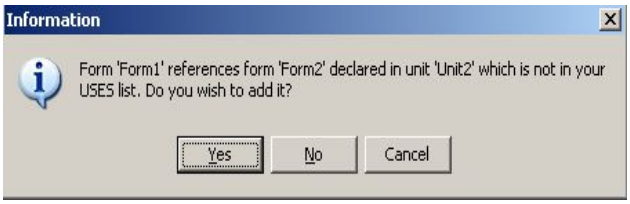
Agar loyihani kompilyatsiya qilsak, xato haqida ma'lumot chiqadi. CHunki ikkinchi forma Unit2 da ta'riflangan, biz uni birinchi formada ishga tushirayapmiz. Birinchi formaga o`tib, *File* menyusidan *Use Unit* punktini tanlaymiz. Quyidagi oyna ochiladi:



Bu oynada kerakli Unit tanlab, OK tugmasini bosamiz va quyidagi kodni qo`shamiz:

```
var
  Form1: TForm1;
  implementation
 uses Unit2;
```

Bu kodni qo`lda kiritish ham mumkin. Agar modul qo`shilmagan bo`lsa, loyihani ishga tushirishda quyidagi ma'lumot oynasi chiqadi: Agar Yes tugmasi bosilsa, modul avtomatik qo`shiladi.



Agar dastur ishga tushirilsa, ekranda birinchi formani aktiv holda ko`ramiz. Agar boshqarish tugmasi bosilsa, ekranda ikkinchi forma aktiv holda paydo bo`ladi. Ikkinchi formada taxrirlash qatoriga biror satr kiritib, boshqarish tugmasini bossak, bu forma berkiladi va Label1 komponentasi Caption xossasiga biz kiritgan satr qiymat sifatida beriladi.

Birinchi formadagi boshqarish tugmasiga quyidagi kodni kiritamiz:

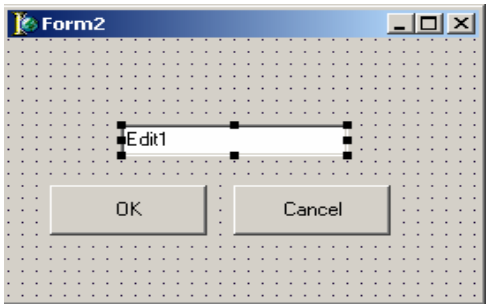
```
procedure TForm1.Button1Click(Sender: TObject);
  begin
        Form2.ShowModal;
        Label1.Caption:='Kup formali loyixa';
  end;
```

Ikkinchi formadagi boshqarish tugmasiga quyidagi kodni kiritamiz

```
procedure TForm2.Button1Click(Sender: TObject);
  begin
      Form1.Label1.Caption:=Edit1.Text;
  end;
```

Agar dasturni ishga tushirsak birinchi forma aktiv shaklda ekranda paydo bo`ladi. Agar boshqarish tugmasini bossak, ikkinchi forma aktiv shaklda paydo bo`ladi, lekin Label1 ustidagi yozuv o`zgarmaydi. CHunki biz ikkinchi formani modul rejimda ochdik.

Boshqarish tugmalarining ModalResult xossalari mavjud bo`lib, ma'lumot almashishda foydalidir. Ikkinchi formaga yangi boshqarish tugmasi qo`shib, tugmalar nomlarini o`zgartiramiz.

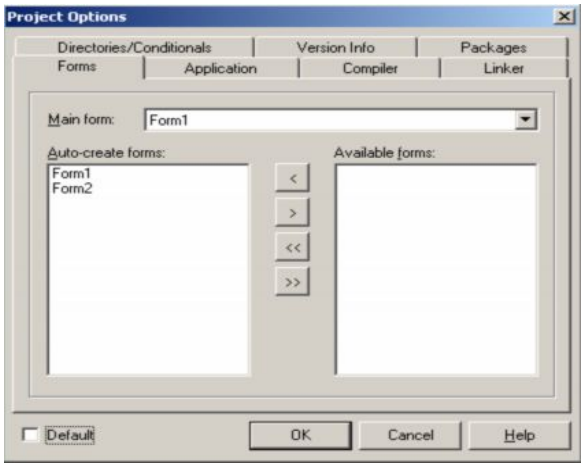


Birinchi tugmaning ModalResult xossasiga *mrOk* qiymati, ikkinchi tugmaning ModalResult xossasiga *mrCancel* qiymati beramiz. Bu tugmalarning Click hodisasiga kiritilgan kodni tozalaymiz. CHunki ModalResult xossasiga qiymat berilishi, bu tugmalarni bosganda forma berkilishiga olib keladi.

Endi birinchi formadagi boshqarish tugmasining Click hodisasiga quyidagi kodni kiritamiz:

```
procedure TForm1.Button1Click(Sender: TObject);
  begin
      Form2.ShowModal=mrOK then
      Label1.Caption:= Form2.Edit1.Text;
  end;
```

Endi ikkinchi formada OK tugmasi bosilsa birinchi formada satr aks etadi.
Loyiha ishga tushganda har doim birinchi forma ishga tushadi. Formalarning ishga tushirish ketma ketligini o`zgartirish uchun Menyuning Project bo`limidagi Options punktini tanlash lozim. Quyidagi dialog oynasi paydo bo`ladi:



Bu oynada Main form qatorida ixtiyoriy formani asosiy forma sifatida tanlash mumkin. Mana shu tanlangan forma birinchi bo`lib ishga tushadi.

Savollar

1.Delphida ilova uchun bosh menyu qanday tashkil qilinadi?
2.Delphida ilova uchun kontekst menyu qanday tashkil qilinadi?
3.MainMenu komponentasining asosiy xossasi nima?
4.Bir necha formalar bilan ishlash qanday amalga oshiriladi?

FOYDALANILGAN ADABIYOTLAR RO`YXATI

1.Faysman A. Professionalьnoe programmirovanie na Turbo Pascale. 1992.

2.Kulьtin M.B. Programmirovanie v Turbo Pascal i Delphi, Sankt-Peter-burg, 2002 g.

3.Kondzyuba S.P., Gromov V.N. Delphi 6/7. Baza dannыx i prilojeniya. M.- Sankt-Peter-burg - Kiev, 2002 g.

4.WWW.Intuit.ru. Internet-Universitet informatsionnыx texnologiy. Moskva.

5.Abramyan M.E. Elektronnыy zadachnik po programmirovaniyu. Versiya 4.6. Rastov-na Donu, 2007.

6.Bobrovskiy S. Delphi 5. Uchebno`y kurs SPb, M.: 2000.

7.Dantemann Djef, Mnshel Djim, Programmirovanie v srede Delphi. K.: NIPF DiaSoftltd,1995.

8.Nazirov Sh., Musaev M., Ne'matov A., Qobulov R. Delphi tilida dasturlash asoslari. G`fur G`ulom nashriyoti, Toshkent, 2007.


Delphi 10 Lite
delphi10lite@gmail.com
v3.0 [SP1 Integrated]