

## **14- Mavzu: Delphi dasturlash tilida ma'lumotlar bazasi bilan ishlash**

### **Delphi dasturlash tili nazariyasi**

Delphi –Windows operatsion tizimida dastur yaratishga yo'naltirilgan dasturlash muhitidir. Delphida dastur tuzish zamonaviy visual loyihalash texnologiyalariga asoslangan bo'lib, unda dasturlashning obyektga mo'ljallangan g'oyasi mujassamlashgan. Delphida dastur Turbo Pascal dasturlash tilining rivoji bo'lgan Object Pascal tilida yoziladi.

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

Yuqori darajadagi mashinali kodda tuzilgan kompilyator;

Obyektga mo'ljallangan komponentalar modullari;

Dastur ilovalarini vizual tuzish;

Ma'lumotlar bazasini tuzish uchun yuqori masshtabli vosita.

Delphi - Windows muhitida ishlaydigan dastur tuzish uchun qulay bo'lgan vosita bo'lib, kompyuterda dastur yaratish ishlarini avtomatlashtiradi, xatoliklarni kamaytiradi ua dastur tuzuvchi mehnatini yengillashtiradi. Delphida dastur zamonaviy visual loyihalash texnologiyasi asosida obyektga mo'ljallangan dasturlash nazariyasini hisoblaydi.

### **Delphi tizimining oynasi va uning elementlari**

Delphi tizimida ishni boshlash uchun uni dasturlar menyusidan topib ishga tushiramiz.

Pusk>Programma> Borland Delphi> Delphi

Delphi oynasi ko'rinishi odatdagidan ancha boshqacharoq bo'lib, u o'zichiga beshta oynani oladi:

- bosh oyna - Delphi Projectl;
- forma oynasi - Form;
- obyekt xossalarini tahrirlash oynasi -Object Inspector;
- obyektlar ro'yxatini ko'rish oynasi - Object tree View;
- dastur kodlarini tahrirlash oynasi - Unit.pas.

Bosh oyna ekranning yuqori qismida joylashgan bo`lib, uning birinshi qatorida sarlavha ya'ni, proyektning nomi joylashgan. Ikkinchi qatorda buyruqlar menyusini gorizontol ko'rinishda joylashgan. Keyingi qatorning chap tarafida uskunalar paneli va o'ng tarafida komponentalar politrasi joylashgan.

## **Buyruqlar menyusi**

Buyruqlar menyusi quyidagilarni o'z ichiga olgan:

- File (fayl) bo'limi fayllar ustida ish bajarish uchun kerakli buyruqlari o'z ichiga olgan;
- Edit (tahrir) bo'limi faul ichidagi ma'lumotlarni tahrirlash uchun kerakli buyruqlarni o'z ichiga olgan:
- Seerch;
- View;
- Compile;
- Run formani ishga tushirish;
- Options;
- Tols servis xizmatidan foydalanish;
- Help yordam chaqirish.

Forma oynasida iloyalar yaratiladi. Object Inspector oynasi obyekt xossalarini tahrirlash uchun xizmat qiladi. Obyekt xossalari bu obyektga berilgan xarakteristika bo'lib, uning ko'rinishi, joylashishi va holatidir. Masalan, Width va Height xossalari forma o'lchamini, top va Lift esa formaning ekrandagi holati, Caption - sarlavha matnini aniqlaydi.

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

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

- dasturiy visual loyihalash jarayoni;
- dastur kodlarini kiritish (yozish) jarayoni.

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

Kodlarni yozish oynasi boshlanishda o'z ichiga holi bo'sh formani akslantiruvchi dastur matnini yozib chiqaradi. Dastur loyihasini ishlash mobaynida dasturchi kerakli dastur operatorlarini kiritib, formani loyiha bo'yicha akslantiradi. Delphida dasturlash forma oynasini tashkil etishdan boshlanadi.

## **Oddiy dastur iloyasini yaratish**

Oddiy dastur iloyasini yaratish ketma-ket File> New> Applisation buyrug'ini berish bilan boshlanadi va buyruqni berishdan oldin ikkita asosiy ishni bajarish lozim:

- papka tashkil etish;
- tizimni to'g'rilash.

Papka tuzing, masalan, My\_Delhp nomli. My\_Delhp papkasiichida yana o'z dasturingizni saqlash uchun papka ochish,

Delphi muhitining standart nastroykasiga o'zgartirish kiritish uchun Options buyrug'ini berish va muloqot darchasidan kerakli o'zgarishlarni bajarish lozim.

Delphi dasturlash muhitida ishlash jarayonida quyidagi kengaytmali fayllar ishlatiladi:

- loyiha fayli, kengaytmasi. dpr;
- paskal moduli fayli, kengaytmasi. pas;
- komponentalar joylashgan fayl, kengaytmasi ".de";
- formalar joylashgan fayl, kengaytmasi. dfm;
- ma'lumotlar bazasi fayli, kengaytmasi .dbf.

### **Delphi dastur uchta asosiy etapdan**

Tayyorlanadigan Delphi dastur uchta asosiy etapdan o'tadi:

- kompilyatsiya;
- komponovka;
- bajarish.

Kompilyatsiya etapida tayyorlangan dastur matni Object Pascal tiliga o'tkaziladi. Komponovka bosqichida esa kerakli qo'shimcha yordamchi dasturlar va ost dasturlar unga birlashtiriladi. F9 tugmasini bosish bilan Save UnitAs dialog oynasi paydo bo'ladi va sizdan Unit.pas moduli uchun fayl nomini va joylashadigan papkani ko'rsatishingizni so'raydi. Agar joyi ko'rsatilmasa Delphi avtomatik ravishda dasturingizni Vin papkasiga joylashtiradi. Yaxshisi siz bu papkani o'z ishchi papkangiz nomiga almashtiring, masalan, My \_Delph. Dastur kompilyatsiya qilinishi paytida Delphi sistemasi pas, dfm va dcu kengaytmali modullar tuzadi. pas kengaytmali faylkodlami yozish oynasiga kiritilgan dastur matnini. dfm forma oynasi tashkil etuvchilarini. dcu kengaytmali fayl esa pas va dfm kengaytmali fayllarning birgalikdagi mashina kodiga o'tkazilgan variantini saqlaydi. Bu dcu kengaytmali fayl kompilyator tomonidan tashkil qilinadi va yagona ishchi (bajariluvchi) .exe kengaytmali fayl tashkil qilishga baza yaratadi.

### **Delphi dasturlash tili muxitiga kirish**

DELPHI dasturlash tili tizimi-bu Windows uchun mo'ljallangan dasturlash muxiti bo'lib, 1995 yilda BORLAND kompaniyasi dasturtuzuvchilari guruhi Chak (Chuck) Va Denni (Danny) tomonidan yaratilgan.

Bu til o'zining keng qamrovli imkoniyatlariga egaligi bilan birga, boshqa dasturlash tillaridan o'zining ba'zi bir xususiyatlari bilan ajralib turadi. Borland Delphining paydo bo'lishi dasturlashni rivojlantirish tarixida yorqin ko'rinish bo'ldi.

### **Delphining tendensiyalar**

Delphining dunyoga kelishiga quyidagi tendensiyalar sabab bo'ldi;

- Windows uchun dasturlash va komponentalar texnologiyasi.
- Masalalarni yechish uchun obyektga yo'naltirilgan usul.
- Komponentalar texnologiyasiga asoslangan ilovalarni tez yaratishning visual muxitlari.

- Interpretasiyadan emas, kompilyasiyadan foydalanish. Bu shundan iboratki, interpretator bilan ishlashga qaraganda kompilyator bilan ishlash tezligi o'n martalab ustunlikka ega bo'ladi.
- Universal usullar yordamida ma'lumotlar bazas ibilan ishlash imkoniyatlarining mavjudligi.

Masalan, local va shu bilan bir qatorda server ma'lumotlari faylidan mijoz-server arxitekturasiga yoki ko'p bosqichli N-tier sxemasiga o'tishni ta'minlash.

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

Hozirgi kunda Delphi tilining bir necha xil Variantlari mavjud.

Agarda sizning kompyuteringizda Delphi dasturi o'rnatilgan bo'lsa uni ishga tushirish quyidagicha bo'ladi: «Pusk» tugmasidagi «Programma» dan Borland Delphi6 bo'limidan Delphi6 dasturi tanlanadi. Natijada Delphi dasturi ishga tushadi. Delphi dasturlash tilini ishga tushirilganda ekranda quyidagi muhit hosil bo'ladi:

### **Vizual komponentalar kutubxonasi**

Delphi sinflari murakkab iyerarxik tuzilmaga ega bo'lgan vizual komponentalar kutubxonasini tashkil qiladi. VKK tarkibiga kiruvchi yuzlab sinflar mavjud. Katta boshqa sinfga ajdod sinf bo'luvchi asosiy sinflarga quyidagilar kiradi.

Komponentalar - sinflarning nusxalari bo'lib, TComponent sinfnig davomchilaridir (avlodidir). Boshqa hamma sinf nusxalari obyektlar deyiladi.

Komponentalar bilan obyektlar orasidagi farq shundaki, formada komponentalar bilan (manipulatsiya qilish) ish ko'rish mumkin, lekin obyekt bilan ish ko'rish mumkin emas. Masalan, komponenta bo'lmagan TFont sinf obyektini qaraydigan bo'lsak uni formaga joylashtirib bo'lmaydi. Komponentalar Label yoki Edit larni formaga joylashtirish mumkin Va ularni joylashtirishda TFont sinf turiga kiruvchi Font xossasidan foydalaniladi.

### **Sinflar va obyektlar**

Object Pascal obyektga mo'ljallangan dasturlash tilidir. Obyektga mo'ljallangan tilda yig'ilgan imkoniyatlarga dasturlash tilining obyekt modellari deyiladi. Object Pascalda obyekt modellarini ishlatilishining amaliy natijasi komponentalarni yaratish va ularni qo'llab quvvatlashdir.

Ma'lumki, Delphi dasturlash yositasi Object Pascal obyektga yo'naltirilgan dasturlash tilini ishlatadi. Obyektga yo'naltirilgan dasturlash (OYD) - bu dastur ishlab chiqish usullari bo'lib, uning asosida real dunyo obyekt va uning holatini ifodalovchi ma'lum tuzilmaga ega obyekt tushunchasi yotadi. Object Pascalda obyekt modelining qo'llanilish natijasi bu komponentalarni qo'llash va yaratishdir. Object Pascal obyekt modelining asosi sinf va obyekt tushunchalaridir.

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

Pascal sinfiy tili dasturchiga o'zining murakkab ma'lumotlar turlari yozuvlar (records) kiritishiga imkon beradi. Obyektli dasturlash konsepsiyasiga asoslangan

Delphi tili sinflar kiritishiga imkon beradi. Sinf murakkab tuzilma bo'lib, ma'lumotlar ta'riflaridan tashqari, protsedura va funksiyalar ta'riflarini o'z ichiga oladi.

### **Asboblarni paneli va komponentalar ro'yxati**

Delphi dasturlash tilida komponentalar bajaradigan Vazifalariga qarab guruhlarga ajratilgan. Har bir guruh komponentalari o'zining nomiga ega bo'lgan alohida bo'limda joylashgan Windows uchun tuzilgan programmaning ichki tuzilishi konsol programmalarining ichki tuzilishidan farqlanadi. Dastur bajarilishida kalit so'zlar bo'lgan begin va end orasiga olingan operatorlarni windows boshqacha shaklda boshqaradi, yani ularni hodisa deb qaraydi. Har bir hodisaning o'ziga mos xossalari mavjud. Dasturlashda bu xossalardan keragini tanlab ishlatiladi. Bu tanlash obyektlar inspektorida qilinadi. Obyektlar inspektori ikki qismdan iborat Properties yani hodisalar Va Events xossalar. Delphi dasturlash tili siz yaratmoqchi bo'lgan dastur (prilojeniya)ni Windows amaliyot tizimi bilan bo'lgan holda boshqarishning keng va juda qulay imkoniyatlarini beradi. Bunday muxitda ishlash uchun eng avvalo siz o'zingizni Windows muxitida bemalol ishlayoladigan foydalanuvchidek xis qilishingiz lozim. Buning uchun esa Windows amaliyot tizimini mukammal darajada yaxshi bilishingiz darkor bo'ladi. Agarda siz Windows amaliyot tizimi bilan tanish bo'lsangiz Windows muxitida ishlovchi dasturlarni tuzish uchun yaxshilab tayyorgarlik qilishingiz lozim. Vizual dasturlash muxitida dastur tashkil qilgan termin va tushunchalarning butun assortimentidan foydalaniladi. Masalan, obyekt, xossa va hodisa tushunchalari Visual dasturlash muxitlarida standart obraz deb tushuniladi.

### **Delphi dasturlash tilida Paskal tili**

Delphi dasturlash tili uchun Paskaldasturlash tili asos qilib olingan. Xo'sh unday bo'lsa, bu ikkita tilning qanday o'xshashlik va farqli tomonlari bor? Biz bilamizki, Paskalda biror kattaroq dastur, masalan amaliy dasturlar majmuini tuzmoqchi bo'lsak, albatta biz modulli dasturlashdan foydalanamiz. Ya'ni qo'yilgan masalan, kichikroq bo'laklarga bo'lib olamiz va shular bilan ishlaymiz. Natijani olish uchun esa, bu bo'laklarni birlashtiruvchi bitta asosiy dastur qilinadi. Aynan shu narsa Delphi dasturlash tilining asosi xisoblanadi. Agar biz Delphida biror forma hosil qilsak, u holda Delphi bizga o'zi avtomatik tarzda shu formaga mos bo'lgan modullarni va bu modullarni boshqaruvchi dasturni tuzib beradi. Boshqaruvchi dasturni Delphida, agar unga o'zimiz nom bermasak, Project1.dpr deb nomlaydi. Bunda dpr fayl kengaytmasi. Avtomatik tarzda tashkil qilingan modulga biz nom bermasak, u holda Delphi uni Unit1.pas deb nomlaydi. O'zimiz esa pas kengaytmasini saqlagan holda xoxlagan nomimizni berishimiz mumkin. Tuzilgan dasturni EXM xotirasida saqlash uchun asosiy menyuning File bandidan foydalaniladi. Dasturni kompilyasiya qilish uchun asosiy menyuning RUN bandidan foydalaniladi. Agar dasturda xatoliklar bo'lsa, kompilyasiya qilish

jarayonida bu xatoliklar ko'rsatib boriladi. Agar xatolik bo'lmasa, dastur ishga tushadi

### **Obeykt tushunchasi**

Agarda hozirgi kunda obeykt so'zi obyektiv dasturlashda namoyon bo'lsa, oldingi davr bilan taqqoslanilganda dastur ta'minoti mashinasining sharofati bilan bu tushunchasi o'z doirasining bir qismini yo'qotadi. Ya'ni, hozirgi davrda Obeykt termini oldingiga qaraganda muhim kenglikda aniqlanadi. Obeykt tushunchasini aniqlash uchun qiyin bo'lgan dasturiy terminlardan foydalanish talab qilinmaydi. Delphi dasturlash tilida ham xuddi boshqa Visual dasturlash muxitlaridagidek Obeykt tugma, belgi, ma'lumot darchasi, maydon va boshqa bo'ladi. Obeyktni yaratishdan maqsad undan dasturda foydalanishdir. Ya'ni Obeykt bu- sizning dasturingizda qandaydir o'lchamda qurilgan elementdir. Obeyktni oddiy bir element deb qarash mumkin. Masalan, ramka, forma, va bir qancha elementlardan tashkil topgan forma.

Obeyktni yaratishda Obeyktni tanlab formada joylashtirilganda Obeykt o'zining standart o'lchamida qo'yilgan joyda paydo bo'ladi. Uning o'lchamlarini esa istagan darajada (formaning o'lchamlaridan kata bo'lmagan) o'zgartirishingiz mumkin. Siz Obeyktni formada joylashishingiz bilanoq, uning barcha atributlari darhol dastur kodida namoyon bo'ladi. Obeyktni ekranda joylashishingiz bilanoq unga tegishli bo'lgan dastur kodlari avtomatik tarzda bitta faylga yoziladi.

### **Xossa tushunchasi**

Har qanday Obeyktning o'ziga xos xususiyatlari bo'ladi. Obyekt hodisadan tashqari o'zining xossalariga ham ega bo'ladi. Xossa bu obyektning o'ziga xos xususiyatlarini ochib beruvchi xususiyatdir. Masalan rangini, balandligi, kengligi, ekranda joylashishini va xokazolarini. Obyekt xossasi bu xuddi proo'eduraning local o'zgaruvchisi kabidir. Ya'ni lokal o'zgaruvchilardan faqat prodsedura bajarilganda foydalaniladi. Xuddi shuningdek xossa ham obyekt bilan uzviy bo'lgan va ular obyekt atributlari bo'ladi, uning qurilish materiallarini namoyon qiluvchi local o'zgaruvchilar prodsedura ichida o'zgartirish uning bajarilishini o'zgartiradi. Shuningdek xossani o'zgartirilsa obyekt xam o'zgaradi.

### **Hodisa tushunchasi**

Delphi dasturlash tilida dastur va foydalanuvchi o'rtasida o'zaro aloqani tashkil qilish uchun hodisalarni boshqarish metodlaridan foydalanamiz. Delphida hodisalar bilan dinamik dasturlashni tushuntirish uchun qo'yidagi misolni keltiramiz. Ya'ni firma va uning ishchilari. Xar bir ishchining kompaniyadagi o'z o'rnini darajasini belgilab beruvchi o'zining majburiyati va vazifasi bo'ladi. Kompaniya tashqi dunyo bilan ro'y berib turadigan hodisalar orqali o'zaro aloqada bo'ladi. Masalan telefon qo'ng'irog'i, xat, faks, xizmat safari va hokazolar. Har bir hodisa mos ravishda birlashtirilgan ishchilar bilan ro'y beradi. Masalan Bobomurodda telefon qo'ng'irog'i bo'ladi, Bobomurod ushbu qo'n'iroqqa javob beruvchi javobgar hisoblanadi. Lekin Bobomurod boshqa

telefonga javobgar emas. Shuningdek boshqa ishchilar ham Bobomurodning telefonidagi qo'n'iroqqa javobgar hisoblanmaydilar. Bu keltirilgan misolda Bobomurod hodisani bajaruvchi bo'ladi. Ushbu misoldagi kabi Delphi sichqoncha, klaviatura va tizimli hodisalarga «hodisa-signal» shaklda javob beradi. Windows uchun tuzilgan programmaning ichki tuzilishi konsolprogrammalarining ichki tuzilishidan farqlanadi. Dastur bajarilishida kalit so'zlar bo'lgan begin va end orasiga olingan operatorlarni windows boshqacha shaklda boshqaradi, yani ularni hodisa deb qaraydi. Har bir hodisaning o'ziga mos xossalari mavjud. Dasturlashda bu xossalardan keragini tanlab ishlatiladi. Bu tanlash obyektlar inspektorida qilinadi. Obyektlar inspektori ikki qismdani borat Properties yani hodisalar va Events xossalar. Delphining xarakterli tomoni shundan iboratki, agar biz biror komponentadan foydalansak, obyektlar inspektori unga mos hodisalarni ajratibko'rsatib turadi. Masalan, biz biror yangi forma xosil qilgan bo'lsak, obyektlar inspektorida captionda Form1 yozuvi turadi. Agar biz xoxlasak Form1 ning o'rniga yangi o'zimizning programmaga mos nomimizni berishimiz mumkin. Har bir komponentaga obyektlar inspektorida unga tegishli hodisa va shu hodisaga mos xossalar bo'ladi.

### **Delphi dasturlash tilining xarakteri**

Delphi dasturlash tilining yana bir xarakterli tomoni shundan iboratki, agar biz dasturda biror tugmaga qandaydir vazifani yuklamoqchi bo'lsak, shu tugmaning ustida sichqonchani ikki marta chertilsa, shu tugmaga mos keluvchi proseduraning ichiga avtomatik ravishda kiritib qo'yadi. Bundan tashqari dastur tuzish jarayonida formadan modulning ichiga va moduldan formaga o'tishga extiyoj bo'ladi, buning uchun asboblardan panelidan ToggleForm-Unit tugmasi tanlanadi yoki F12 tugmasini bosish bilan amalga oshiriladi.

Formalar bilan ishlash.

Delphida tuziladigan dasturlar biror forma asosida qilinadi. Delphida har bir yangi hosil qilingan formaga unga mos bo'lgan modul avtomatik tashkil qilinib turiladi. Bu esa dasturchi uchun juda qulay imkoniyat, yani uning ishini tezlashtirishga yordam beradi.

Bu formaning shaklini tanlash, unda komponentalarni joylashtirish bizning ixtiyorimizda bo'ladi. Formaga biror komponentani qo'ymoqchi bo'lsak, shu komponentaning ustida sichqonchani chap tugmasini ikki marta bosiladi. Bu komponenta formaning o'rtasiga joylashadi. Biz uni hoxlagan joyimizga surib ko'chirishimiz mumkin.

Delphi dasturlash tilida Pascal tili asos qilib olingan. Chunki bunga sabab dasturlashni o'rganmoqchi bo'lganlarning ko'pchiligi birinchi navbatda Pascal dasturlash tili bilan tanishib chiqadi. Delphi dasturlash tili uchun Paskal dasturlash tili asos qilib olinibdi. Xo'sh unday bo'lsa, bu ikkita tilning qanday o'xshashlik va farqli tomonlari bor?

Delphi dasturlash tilida Pascalsiz foydalangan oldingi Pascal tiliday bo'lmaydi. Delphi dasturlash tilida yanada qulayroq Borland Pascal7 dasturlash tili tomonidan qo'llab quvatlangan yangi sintaksisdan foydalaniladi.

Bizbilamizki, Paskalda biror kattaroq dastur, masalan amaliy dasturlar majmuini tuzmoqchi bo'lsak, albatta biz modulli dasturlashdan foydalanamiz. Ya'ni qo'yilgan masalani kichikroq bo'laklarga bo'lib olamiz va shular bilan ishlaymiz. Natijani olish uchun esa, bu bo'laklarni birlashtiruvchi bitta asosiy dastur qilinadi. Aynan shu narsa Delphi dasturlash tilining asosi hisoblanadi. Agar biz Delphida biror forma hosil qilsak, u holda Delphi bizga o'zi avtomatik tarzda shu formaga mos bo'lgan modullarni Va bu modullarni boshqaruvchi dasturni tuzib beradi. Boshqaruvchi dasturni Delphida, agar unga o'zimiz nom bermasak, Project1.dpr deb nomlaydi.

Bunda dpr fayl kengaytmasi. Avtomatik tarzda tashkil qilingan modulga biz nom bermasak, u holda Delphi uni Unit1.pas deb nomlaydi. O'zimiz esa pas kengaytmasini saqlagan holda xoxlagan nomimizni iberishimiz mumkin. Boshqa dasturlash tili Basic Microsoft firmasi tomonidan ishlab chiqarilgan Vishual basic dasturlash tilida qo'llaniladi. Bu dasturlash tili Delphi dasturlash tiliga yaqin, ammo obyektga mo'ljallangan xossalariga muvofiq emas. Shunday bo'lsa ham Visualbasic hodisalar bilan boshqariladi Va Windows asosida professional ishlab chiqarishda zarurligini ko'rsatdi. Uning asosiy imkoniyatlari qator yillar davomida o'z holicha qolib kelmoqda. Delphi vujudga kelgunga qadar Visualbasic haqiqatdan hamma Windows muxiti uchun yagona visual ishlab chiqarish instrumenti bo'lgan.

### **Dasturning formalari**

Delphida tuziladigan dasturlar biror forma asosida qilinadi. Delphida har bir yangi hosil qilingan formaga unga mos bo'lgan modul avtomatik tashkil qilinib turiladi. Bu esa dasturchi uchun juda qulay imkoniyat, yani uning ishini tezlashtirishga yordam beradi. Bu formaning shaklini tanlash, unda komponentalarni joylashtirish bizning ixtiyorimizda bo'ladi. Formaga biror komponentani qo'ymoqchi bo'lsak, shu komponentaning ustida sichqonchanning chap tugmasini ikki marta bosiladi. Bu komponenta formaning o'rtasiga joylashadi. Biz uni xoxlagan joyimizga surib ko'chirishimiz mumkin.

### **Delphining forma komponentalari**

Forma komponentalari, bu 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 - dastur bosh menyusi. Komponenta murakkab iyerarxik tuzilmali yaratish uchun xizmat qiladi.

Pop menu - yordamchi yoki lokal menyusi. Bu menyu oynada sichqonchanning o'ng tugmasini bosish bilan chiqadi.

Label - metka (belgi), Bu komponenta [orta oynasiga uncha uzun bo'lmagan Ir qatorli yozuyni chiqarishda ishlatiladi Ya uning piktogrammasi panelda «A» ko'rinishda berilgan.

Edit - kiritish qatori. Forma oynasida matnli qator kiritish va tahrirlashda ishlatiladi.

Memo - ko'p qatorli matn muharriri. Ko'p qatorli matnlarni kiritish yoki chiqarishda ishlatiladi.

Button - buyruq tugmasi). Bu komponent dasturchi tomonidan berilgan bir necha buyruqlarni bajarishda ishlatiladi.

CheckBox - bog'liq bo'lmagan tanlash tugmasi. Dasturda bu komponentning asosiy mantiqiy xossasi (Checked) o'zgartiriladi.

RadioButton - bog'liq bo'lgan tanlash tugmasi. y angli tanlash tugmasi bosilganda, oldin tanlangan tugmani avtomatik rayishda ozod etadi.

## **Yangi komponentning yaratilishi**

Delphi dasturlash muhiti bilan tanishishni davom ettirar ekanmiz, komponentlar to'plami haqida gapirishga to'g'ri keladi.

Komponentlar to'plamining birinchi sahifasida 14ta obyektlar joylashgan bo'lib, hamma dasturchilar albatta ushbu obyektlardan foydalanishgan. Chunki, juda ham kamchilik tugma, ro'yxat, kiritish oynalari va ho kazolardan foydalanmasdan dastur tuza oladi. Bu obyektarning barchasi Windowsdagi sichqoncha va muloqot oynasi kabi muhimdir.

Har bir sahifadagi komponentlar to'plami va joylashishini o'zgartirish mumkin.

Demak, siz mavjud komponentlarni o'rnini o'zgartirish bilan birgalikda ularning tarkibiga yangilarini qo'shishingiz mumkin.

Delphining standart komponentlari va ularni ishlatish haqidagi qisqacha izoh quyida keltirilgan. Bu komponentlarni o'rganayotgan vaqtda qo'l ostingizda kompyuter bo'lgani ma'qul, chunki birvarakayiga ularning ishlash printsipli bilan ham tanishishingiz mumkin.

## **Obyektlar Inspektori haqida**

Oldinroq biz Obyektlar Inspektori (Object Inspector) bilan bir oz tanishib o'tdik. Endi u bilan yaqinroq tanishish vaqti etdi. Obyektlar Inspektorining asosiy vazifasi formaga tashlangan ixtiyoriy obyektning xususiyatlarini tahrirlash hisoblanadi.

Bundan tashqari u yordamida formaning xususiyatlarini ham o'zgartirish mumkin.

Obyektlar Inspektorini o'rganishning eng qulay usuli – u bilan ishlashdir. Avval yangi loiha ochamiz File | New Project. So'ngra formaga TMemo, TButton va TListBox obyektlarini 1-rasmda ko'rsatilganidek joylashtiring. Xususiyatlar bilan ishlashni boshlashda Ctl3D xususiyatidan boshlaymiz (jimlik qoidasi bo'yicha faol).

Formaga sichqon tugmasini bosing, Obyektlar Inspektoriga o'ting va Ctl3D xususiyati qiymatini sichqoncha yordamida bir necha marta o'zgartirib ko'ring. Bu amal formaning tashqi ko'rinishini o'zgartirayotganligiga e'tibor qarating.

Formaning Ctl3D xususiyati qiymatini o'zgartirish shu formaga tashlangan hamma obyektarning shu xususiyati qiymatini o'zgartiradi.

Formaga dastlabki holatini qaytaring va undagi TMemo obyektini belgilang,

so'ngra <Shift> tugmasini bosib turib, TListBox obyektini belgilang. Endi siz

formadagi bir nechta obyektarni belgilay olasiz. Obyektlar guruhini belgilaganda

Obyektlar Inspektorida faqat shu guruh obyektlari uchun umumiy bo'lgan xususiyatlar ko'rinadi va ularning qiymati o'zgartirilishi belgilangan obyektlarning barchasi uchun amal qiladi. Agar bu guruhda umumiy xususiyatlarga ega bo'lgan obyektlar bo'lmasa, Obyektlar Inspektorida hech qanday xususiyat ko'rinmaydi.

### **Tashqi dasturlarni ishga tushirish**

Ma'lumotlar omborining maqsadi ma'niga ega bo'lgan ma'lumotlarni bir yoki bir nechta joyda ortiqcha sarflarsiz (xotira va zaxiralarni iqtisod qilib) saqlashdir. Shunday qilib ma'lumotlar ombori ikkita asosiy maqsadlarni ko'zda tutadi: ma'lumotlarni yoo'qolishdan saqlash va ularning ishonchliligini oshirish. Demak, yaxshi loihalangan ma'lumotlar ombori:

- foydalanuvchilarning ma'lumotlar omboriga bo'lgan barcha talablarini qondiradi. Shuning uchun ma'lumotlar omborini loihalashdan oldin foydalanuvchilarning ma'lumotlar omboriga bo'lgan talablarini keng miqyosda o'rganib chiqish zarur.
- Ma'lumotlarning mosligi va ishonchliligini kafolatlaydi. Jadvallarni tuzishda foydalanuvchi tomonidan mumkin bo'lmagan ma'lumotlar kiritilashining oldini olish uchun ma'lum shartlarni kiritish kerak.

### **Delphida ma'lumotlar ombori bilan ishlovchi komponentlar sharhi**

Delphida ma'lumotlar ombori bilan ishlovchi juda ko'p komponentlar bo'lib, biz ulardan asosan TTable, TDbgrids, TDataSource, TDbNavigator komponentlari bilan tanishamiz.

- TTable obyekt ma'lumotlar omboridagi mavjud jadval bilan mulokot o'rnatish uchun xizmat qiladi. TTable ixtiyoriy tipdagi (FoxPro, ODBC, SQL ...) ma'lumotlar omborining har bir yozuviga va maydoniga to'g'ridan to'g'ri murojaat qila oladi. Bu komponent shuningdek, alohida hisobotlar bilan ham mulokot o'rnatadi.

TTable obyektidan foydalanishdan oldin unga ma'lumotlar ombori alyasini ulash kerak, ya'ni shu komponentning DatabaseName xususiyatida chiqadigan ro'yxatdan kerakli alyasni tanlash va TableName xususiyatidagi ro'yxatdan kerakli jadval nomini tanlash kerak. TTable obyektini faollashtirish uchun Active xususiyati qiymatini *true* ga o'tkazish kerak.

### **Ma'lumotlar omboriga murojaat qiluvchi dastur**

Endi yuqorida aytilganlarni umumlashtirib, Delphida ma'lumotlar ombori bilan ishlovchi dasturga misol ko'rib o'tamiz. Ushbu dasturda biz standart ma'lumotlar ombori bo'lgan DBDemos dan foydalanamiz (Siz o'zingiz tuzgan ma'lumotlar omborini ham alyas sifatida dasturlarga kiritishingiz mumkin).

Demak, yangi loiha hosil qilamiz va Formalar Dizayneri rejimida formaga BDE sahifasidan TTable, DataAcces sahifasidan TDataSource, Data Controls sahifasidan TDBGrid va TDbNavigator komponentlarini tashlaymiz Demak hosil bo'lgan tasvir taxminan rasmdagidek bo'lishiga erishamiz. So'ngra, Table1 (TTable) obyektinig DatabaseName xususiyatidagi ro'yxatdan DBDemosni tanlaymiz,

TableName xususiyatidagi ro'yxatdan Animals.dbf jadvalini tanlab, Activ xususiyati qiymatini Truega o'rnatamiz. Endi DataSours komponentining DataSet xususiyatidan Table1ni tanlaymiz. Oxirgi ishimiz DBGrid1 va DBNavigator1larning DataSours xususiyatidan DataSours1 ni tanlab, dasturni ishga tushiramiz. Buning uchun biz hech qanday dastur yozmadik va bu komponentlarning o'zi ma'lumotlar bilan oddiy amallarni bajarish imkonini beradi.

### **Ismlar va identifikatorlar**

Ma'lumki, ma'lumotlarning tahlili jarayonini ifodalovchi algoritm turli xil ob'ektlar (o'zgarmlar, o'zgaruvchi miqdorlar, funksiyalar Va hokazo) ustida ish olib boradi. Bu ob'ektlarga ularning Vazifasi Va qabul qiladigan qiymatlariga qarab maxsus ismlar beriladi. Shu ismlarni odatda, identifikatorlar deb ataladi. Identifikator deb harf yoki «\_» belgisidan boshlanuvchi harf, raqam Va «\_» belgisining ixtiyoriy ketma-ketligiga aytiladi: Xizmatchi so'zlardan identifikator sifatida foydalanish mumkin emas. Odatda identifikator so'zining o'rniga qulayroq Va qisqaroq qilib ism deyish mumkin. Dasturda qatnashuvchi ob'ektlarga ismlarni dasturchi o'z ixtiyoriga ko'ra tanlab olishi mumkin. Bir xil ism bilan bir necha xil ob'ektlarni nomlash mutlaqo mumkin emas. Turbo Paskal muhitida ismda qatnashuvchi belgilar soni (ism uzunligi) 63 ta belgidan oshmasligi kerak.

### **O'zgaruvchilar**

O'zgaruvchi dastur obyekti bo'lib, turli xil qiymatlarni xotirada ma'lum nom bilan saqlab turish uchun ishlatiladi. O'zgaruvchi o'z qiymatini dasturning bajarilish davomida o'zlashtirish operatori yordamida qabul qiladi. Qabul qilingan qiymat o'zgaruvchiga boshqa yangi qiymat berilmaguncha saqlanib turiladi Va yangi qiymat berilishi bilan eski qiymat butunlay o'chib, yo'q bo'lib ketadi. Har bir o'zgaruvchiga ma'lum bir tipga tegishli qiymatlarnigina qabul qilish huquqi beriladi. Boshqa tipdagi qiymatlarni o'zlashtirishga urinish dasturning xatoligini ta'minlaydi.

O'zgaruvchi — bu identifikatordir. Uning ismi o'zgaruvchining qiymatiga murojaat qilishda ishlatiladi. Boshqacha aytganda, dastur matnidagi ism shu o'zgaruvchining qiymatini ifodalaydi.

kursidan funksiya tushunchasi bizga yaxshi ma'lum. Algoritmik tillarda faqat qiymatini hisoblash algoritmlari ma'lum bo'lgan funksiyalargina ishlatiladi. Dastur tuzuvchi dastur uchun lozim bo'lgan kerakli funksiyalarni o'z dasturiga kiritishi mumkin.

Xuddi funksiyalar kabi hal qilinayotgan masalaning ma'lum bir tugallangan bosqichlarini hisoblash Vazifasini protseduralar zimmasiga yuklasa ham bo'ladi. Funksiyani hisoblash natijasida faqat, yagona natijaviy qiymatga erishiladi, protseduradan foydalanganda esa, natijaviy qiymatlar soni yetarlicha ko'p bo'lishi mumkin.

Dasturda aniqlangan funksiya Va protseduralar o'zgaruvchilarning e'loni bo'limida e'lon qilinib qo'yilishi kerak. Bunda har bir funksiya Va protseduraga ularning

bajaradigan Vazifasiga mos ismlar berib qo'yiladi. Ularni aniqlashda formal parametrlardan foydalaniladi. Bu parametrlarning tiplari o'z navbatida, funksiya Va protseduraning ichida aniqlanilib, e'lon qilinadi.

Dasturda aniqlangan funksiya Va protseduralardan foydalanish uchun dastur matnida ularning ismlari Va formal parametrlarga mos bo'lgan faktik parametrlari berilishi kerak.

## **Foydalanilgan adabiyotlar**

1. Informatika va informatsion texnologiyalar, M. Aripov va boshqalar. Oliy o'quv yurti talabalari uchun darslik. Toshkent-2019 y.
2. Axborot texnologiyalari, M. Aripov va boshqalar. Oliy o'quv yurti talabalari uchun o'quv qo'llanma. Toshkent-2019 y.
3. Delphi tilida dasturlash asoslari, Sh. Nazirov. Toshkent-2018 y.