

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

**AJINIYOZ NOMIDAGI
NUKUS DAVLAT PEDAGOGIKA INSTITUTI**

Avezov Mardonbek Abdullayevich

**MAVZU: DELPHI DASTURLASH TILIDA YARATILGAN PEDAGOGIK
DASTURIY VOSITALARNING MA'LUMOTLAR BAZALARI BILAN
INTEGRATSIYASINI TA'MINLASH**

5A110701 - «Ta'limda axborot texnologiyalari»

Magistr akademik darajasini olish uchun yozilgan

D I S S E R T A T S I Y A

MAK da himoyalashga ruxsat

Magistratura bo'limi boshlig'i:

_____ dots. B. Atashov

Kafedra mudiri: f.-m.f.n _____

dots. Turenliyazova.A.

Ilmiy rahbar: f.-m.f.n _____

dots.Izimbetov.E.

Nukus-2016

KIRISH

I-BOB. DELPHI DASTURLASH MUHITIDA ISHLASH TAMOIYILLARI

1.1. DELPHI DASTURLASH MUHITINING ASOSIY INTERFEYSI.....

1.2. MA'LUMOTLAR BAZALARI VA ULARNI BOSHQARISH.....

1.3. DELPHI MUHITIDA MA'LUMOTLAR BAZALARI BILAN ISH
YURITISH TEHNOLOGIYSI.....

II-BOB. DELPHI MUHITIDA PEDAGOGIK DASTURIY VOSITALAR YARATISH

2.1. MAKTAB MATEMATIKA KURSIGA BOG'LIQ MASALALARNI
YECHISHGA MO'LJALLANGAN DASTURIY VOSITALAR YARATISH

2.2. O'QUVCHILAR BILIM DARAJASINI ANIQLOVCHI DASTURIY
VOSITALAR YARATISH.....

III-BOB. PEDAGOGIK DASTURIY VOSITALARNI MA'LUMOTLAR BAZALARINI BOSHQARISH TIZIMLARI BILAN INTEGRATSIYALASH

3.1. PEDAGOGIK DASTURIY VOSITALARNING MA'LUMOTLAR
BAZASINI YARATISH.....

3.2. ALGORITMLASH BO'YICHA MA'LUMOTLAR BAZASINI
YARATISH.....

XULOSA.....

FOYDALANILGAN ADABIYOTLAR TIZIMI.....

KIRISH

Xozirgi davr – axborot texnologiyalari rivoj topgan davr, shu sababli biz ta'lim tizimini o'zgartirib, ta'lim tizimiga yangiliklar kiritishimiz lozim. Ushbu yangiliklar nimalardan iborat bo'lishligi lozim? Ular birinchi navbatda kimlarga moljallangan bo'lishi kerak? Ushbu savollarni biz magistrlik dissertatsiya borisida ko'rib chiqishga harakat qilamiz. Ushbu magistrlik dissertatsiya kirishuv bo'limidan, 3 bobtan, xulosalardan va foydalanilgan adabiyotlar ro'yxatidan iborat.

Mavzu dolzarbligi. Bizning fikrimizcha, dissertatsiya mavzusi xozirgi zamonga mos, actual bo'lib topiladi, sababi xozirgi zamonimizning ta'lim tizimiga qo'yadigan talablari shunday-ki, ularga javob berish uchun ta'lim tizimiga yangi pedagogic texnologiyalar kiritilish lozim va ushbu texnologiyalar, albatta, axborot texnologiyalariga asoslangan bo'lishi kerak.

Birinchi bobda Delphi dasturlash muhiti va undagi ma'lumotlar bazalari bilan ish yurutuvchi texnologiyalar haqida tushunchalar berilgan.

Ikkinchi bobda Delphi muhitida pedagogic o'rgatuvchi dasturiy vositalar va o'quvchilar bilimini tekshiruvchi dasturiy vositalar yaratilishi ko'rib chiqilgan.

Uchinchi bobda esa ushbu yaratilgan dasturiy vositalar kompleksini ma'lumotlar bazasi bilan bog'lash yoki integratsiyalash usullari ko'rib chiqilgan. Barcha dasturiy vositalar maktab kursidagi matematika masalalarini yechish uchun muljollangan.

I-BOB. DELPHI DASTURLASH MUHITI

1.1. DELPHI DASTURLASH MUHITI HAQIDA TUSHUNCHALAR

Delphi tizimi - bu Windows uchun yaratilgan dastur muhiti va dasturlash tili bo'lib 1995 yilda Borland kompaniyasi guruhi dastur tuzuvchilari Chak (Chuck) va Denni (Danny) tomonidan yaratilgan.

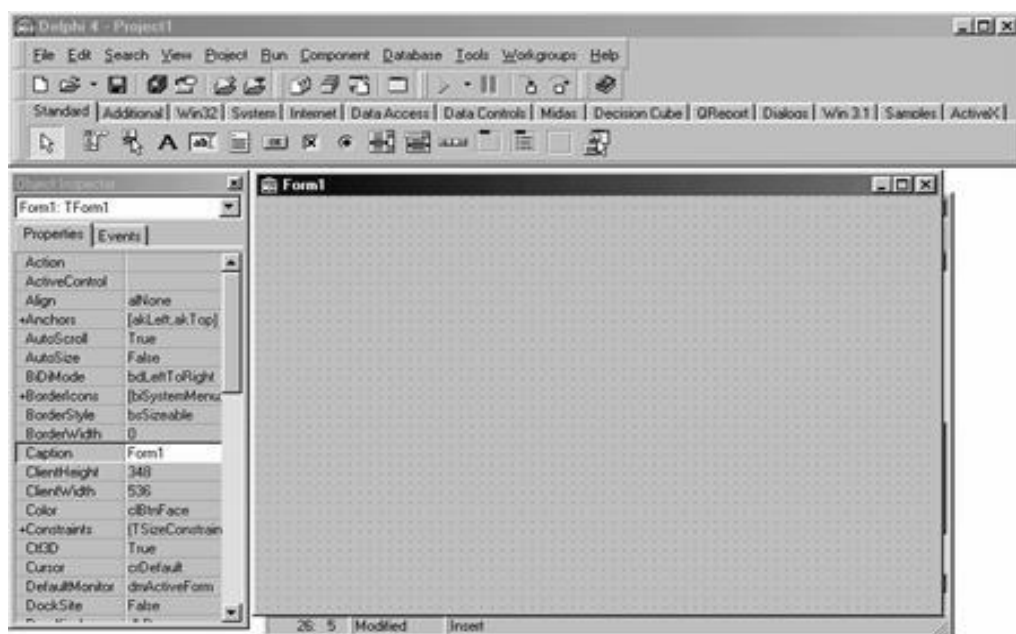
Bu til o'zining keng qamrovli imkoniyatlariga egaligi bilan birga, boshqa dasturlash tillaridan o'zining ba'zi bir xususiyatlari bilan ajralib turadi. Delphi dasturlash tilida dasturlar visual loyihalashning hozirgi zamon texnologiyalari asosida yaratiladi, o'z navbatida ob'eykga mo'ljallangan dasturlash g'oyasiga asoslangan. Borland Delphining paydo bo'lishi dasturlashni rivojlantirish tarixida yorqin ko'rinish bo'ldi. Delphi dasturlash tilidagi dasturlar Object Pascal tilida yoziladi, bu til Turbo Pascal tilining davomi va takomillashtirilganidir. Uning soddaligi, yuqori sifatli kompyatorga ega ekanligi va dasturlar yaratishning qulay muhitiga ega bo'lishi sabab bo'lgan. Delphining dunyoga kelishiga quyidagi tendensiyalar sabab bo'ldi:

- * Windows uchun dasturlash va komponentlar texnologiyasi;
- * Masalalarni yechish uchun obyektga yo'naltirilgan usul;
- * Komponentlar texnologiyasiga asoslangan ilovalarni tez yaratishning vizual muhitlari;
- * Interpretatsiyadan emas, kompilatsiyadan foydalanish. Bu shundan iboratki, interpretator bilan ishlashga qaraganda kompilator bilan ishlash tezligi o'n martalab ustunlikka ega bo'ladi;
- * Universal usullar yordamida ma'lumotlar bazasi bilan ishlash imkoniyatlarining mavjudligi;

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

Dasturiy texnologiyalar bir joyda turib qolmagani kabi Borland (1998 yil apreldan Inprise Corporation) firmasining navbatdagi kuchli mehnatlari natijasida Turbo Pascal tili o'rniga Object Pascal tili – ob'eykga mo'ljallangan dasturlash konsepsiyasini o'zida mujassamlashtirgan tili yuzaga keldi.

Delphi va Object Pascal uzoq evalyusion rivojlanish natijasi va hozirgi zamon kompyuter texnologiyalarining o'zida akslantiruvchi mahsulotdir. Xususan, Delphi yordamida har xil dasturlar – konsol ilovalardan tortib, ma'lumotlar bazasi bilan, internet bilan ishlash dasturlarigacha tuzuish mumkin. Shunday ekan Delphi dasturini o'rganishni nimadan boshlash kerak degan savol tug'iladi? Delphi dasturlash tili bilan tanishishni Object Pascal tilini o'rganish va bu tilni konsol ilovalarni yaratish uchun dasturkash tuzishda qo'llashdan boshlasa bo'ladi.



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

Delphi ning ba'zi bir o'ziga xos xususiyatlariga to'xtab o'tamiz. Delphining ixtisoslashtirilgan dasturlash muhiti quyidagilarni o'z ichiga oladi:

- AppBrowser tahrirlagichi (redaktori)

- Formalar Konstruktori loyiha (proyekt) boshqaruvchisi (menedjeri)

- Delphi fayllari

- Obyektlar bazasi (Object Repository)

Dasturlashtirishning Delphi ga o'xshash vizual asboblarida dasturlash muhiti ko'pincha shu muhitda ishlatiladigan dasturlash tilidan muhimroq ahamiyatga ega. Delphi 4 da vizual dasturlashning butunlay yangi ko'rinishi qo'llanilgan va biz hozir uni batafsil o'rganib chiqamiz.

Delphi 4 tarkibiga kiruvchi tahrirlagich hamma imkoniyatlarini saqlab qolgan holda dastur kodini yozishga butunlay yangicha yondoshadi. Bu yangiliklarning uchta asosiylari: Code Explorer (kod yetakchisi—modulda e'lon qilingan barcha obyekt va identifikatorlarni ko'rsatadi), kod bo'yicha erkin harakat ta'minoti (huddi WEB brouzyerdagiday) va kodlarni avtomat generatsiyalash (hosil qilish)ning yangi texnologiyasi Class Completionlardir.

Code Explorer darchasi modulda e'lon qilingan tiplar, o'zgaruvchilar va modulda ishlatilayotgan boshqa modullarnig ro'yxatini ko'rsatadi. Code Explorer Klassga o'xshash murakkab turlarning xossalari (свойств), (metodlar)i haqida batafsil ma'lumot berishi mumkin. Code Exploryerdagi ma'lumotlar kod yozish davomida avtomat ravishda yangilanib turadi. Bundan tashqari Code Explorer yordamida siz kerakli obyekt yoki o'zgaruvchilarni topishda foydalanishingiz mumkin. Buning uchun sichqoncha ko'rsatkichi bilan kerakli obyekt nomini ikki marta chertish kifoya.

Code Insight. Delphi 4 da Code Insight («Kod—ichidan») texnologiyasining yangilangan shakli qo'llaniladi. Code Insight ning hamma imkoniyatlari yozilayotgan kodni va unga qo'shilgan VCL metodlarining kodlarini to'xtovsiz sintaksik analiz qilishshga asoslangan. Bu texnologiyaning asosiy mezoni boshlovchilar va tajribali dasturchilarning ishini yengillatishdan iborat. Code Insight quyidagi imkoniyatlarga ega:

Code Completion (kodni to'ldirish) Dasturlashtirishni yangi o'rganayotganlar uchun juda qo'l keladi. Bunda biror obyektning nomi terilib nuqta qo'yilgandan keyin metod yoki o'zgaruvchilarning bosh harflari terilishi bilan o'sha harflar bilan boshlanadigan barcha elementlar ro'yxati chiqariladi. Bu ruyxatni Ctrl Q probel ni bosib ham chiqarish mumkin.:q belgisi qo'yilgandan esa,

mana shu obyekt yoki o'zgaruvchining tipiga mos barcha obyekt yoki o'zgaruvchilar ro'yxati chiqariladi.

Code Templates (Kod andozasi) kodlarni kompyuterga kiritishda bir xil qaytariladigan kodlarni avtomatik ravishda kiritish imkonini beradi. Masalan, kodlarda **if... then begin ... end else begin ... end;** konstruksiyasi juda ko'p uchraydi. Bunday konstruksiyani qayta-qayta terib o'tirmaslik uchun Ctl+Q ni bosish va chiqarilgan ro'yxatdan kerakli konstruksiyani tanlash kifoya. Bunday andozalarni istagancha hosil qilish mumkin.

Code Parameters (funksiya parametrlari) funksiya nomi terilib, qavs belgisi qo'yilgandan keyin ushbu funksiyaning parametrlari ro'yxatini suzib chiquvchi xabar ko'rinishida chiqaradi. Bunda joriy parametr ajratib ko'rsatiladi. Bundan tashqari Delphi 4 da Code Parametrs o'zgaruvchilarning indalmagandagi, berilmagan qiymatlarini va qayta yuklanuvchi (overloading) usullarini ham ishlatishni ta'minlaydi.

Formalar (shakllar) konstruktori

Delphining asosiy darchalaridan birinchisi kod tahrirlagichi bo'lsa, ikkinchisi shubhasiz, formalar konstruktorigidir. Delphining formalar konstruktori komponentlar palitrasidan foydalanib, formaga istalgan obyektни qo'yishi mumkin (komponentlar palitrasidan kerakli obyekt tanlanadi, formaga chertiladi, formada kerakli obyekt standart o'lchamlari bilan paydo bo'ladi.). Bunda yangi hosil bo'lgan obyektning kodi avtomatik ravishda kod tahrirlagichi yordamida modulga qo'shib qo'yiladi.

Formaga joylashtirilgan komponentlarni Projects sahifasidagi Object Inspector darchasi yordamida boshqarish imkonini beradi.

Boshlang'ich koddan hodisaga javob beruvchi usulni olib tashlashning eng yaxshi yo'li bu usha amalning sarlavhasi va kalit so'zlar begin va end dan boshqa hamma kodni olib tashlashdir.

Proyektlarni boshqarish

Object Inspector ning Events sahifasi yordamida tanlangan komponentlar bilan hodisalar bog'lanadi. Buning uchun bir qancha usullardan foydalanish mumkin.

Delphi 4 ning yangiliklaridan biri- bu bir vaqning o'zida bir necha projekt (loyiha)lar bilan ishlash imkoniyatining mavjudligidir. Bu View ➤ Project Manager buyrug'i yordamida bajariladi.

Projektni kompilatsiya qilish va yig'ish

Projektni bir necha usul bilan kompilatsiya qilish mumkin. Agar Run ➤ Run buyrug'i ishga tushirilsa, projekt bajarilishidan oldin kompilatsiya qilinadi. Bunda faqat o'zgartirilgan modullargina kompilatsiya qilinadi.

Agar Complite ➤ Build All tanlansa, projektga qo'shilgan modullar o'zgargan-o'zgarmaganidan qat'iy nazar, kompilatsiya qilinadi.

Projekt yig'ilishi uchun undagi har bir boshlang'ich fayl Delphining kompilatsiya qilingan moduli (Document Complited Unit, DCU)ga aylantiriladi. Bunda fayl Pascal boshlang'ich faylining nomi bilan .dcu kengaytmasi bilan saqlanadi. Projektning boshlang'ich kodi kompilatsiya qilinganda projekt modullarining kodlari VCL kutubxonasi kodlari bilan birgalikda bajariluvchi faylga kompilatsiya qilinadi.

Complite buyrug'i AppBrowser tahrirlagichiga projekt yuklangan paytdagina bajarilishi mumkin. Agar faol projektlar bo'lmasa, Open Project komandasi yordamida .PAS kengaytmali faylni ochish va uni sintaksisini tekshirib, DCU ga kompilatsiya qilish mumkin.

Delphining boshlang'ich fayllari

Formaning Pascal fayli forma turkumi (sinfi) ning tacnifini va qayta ishlovchilarning boshlang'ich kodlarini saqlaydi. Object Inspector yordamida o'rnatiladigan xossalar va formaning o'zining ta'rifi alohida .DFM kengaytmali fayllarda saqlaydi.

Bu ikki xil fayldan tashqari projekt uchun o'ta zarur bo'lgan, Pascal kodlaridan iborat Delphi ning loyiha fayli (DPR) ham mavjud. Bu fayl loyiha

yaratilayotganda avtomat ravishda hosil qilinadi. Uni View ➤ Project Source buyrug'i yordamida ko'rish mumkin.

Obyektlar Repozitariysi (bazasi)

Delphida yangi forma, loyiha, ma'lumotlar moduli (Data module) hosil qilishga imkon beradigan bir necha buyruqlar mavjud. Bu buyruqlar File ➤ New tanlanganda chiqariladi. Bunda Delphi obyektlar repozitariysini ochadi. Repozitariy ixtiyoriy turdagi yangi elementlar: formalar (shakllar), ma'lumotlar moduli, oqim obyektlari (thread objects), komponentlar, automation obyektlar va boshqalarni yaratish imkonini beradi.

Obyektlar repozitariysiga bundan keyingi loyihalarda ishlatilishi mumkin bo'lgan yangi elementlarni qo'shish mumkin. Buning uchun Project ➤ Add Repository buyrug'ini bajarish va hosil bo'lgan muloqot darchasini to'ldirish kerak.

1.2. DELPHI MUHITIDA MA'LUMOTLAR BAZALARI BILAN ISH YURITISH TEHNOLOGIYALARI

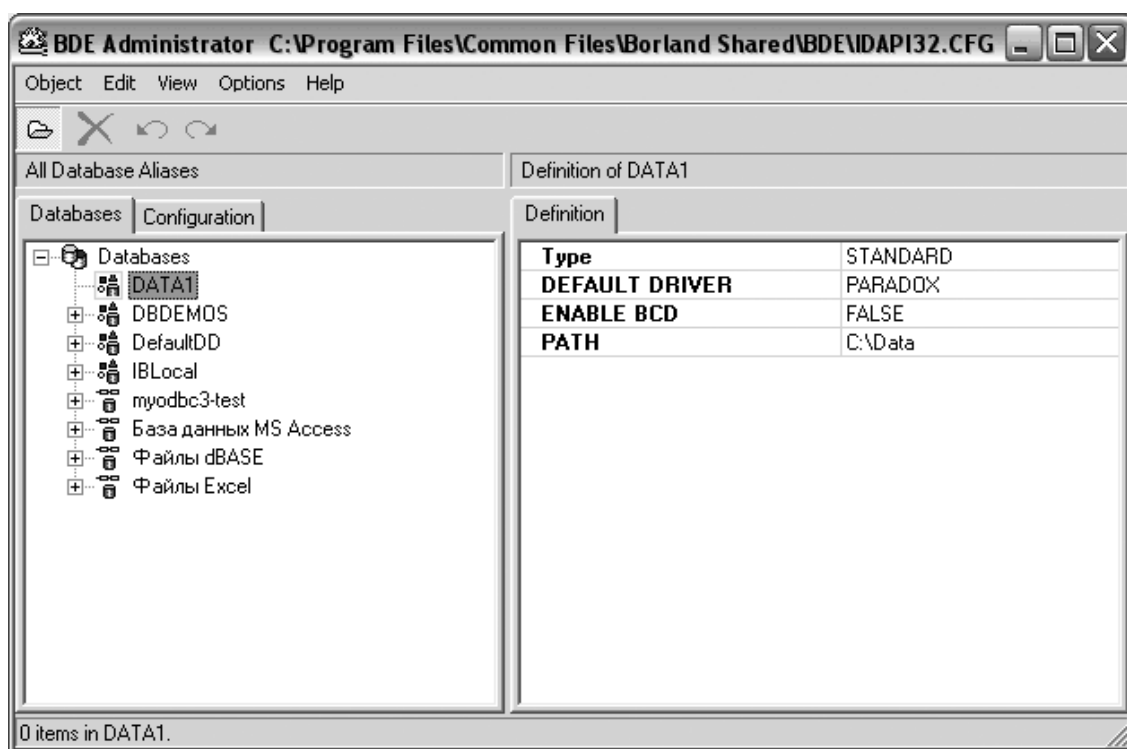
Delphi muhitida ma'lumotlar bazalari bilan ishlash texnologiyalaridan biri - Microsoft ActiveX Data Objects texnologiyasi ma'lumotlar bazalari bilan ish olib borishni oson va tezkor amalgam oshiradi. ADO texnologiyasi va OLE DB interfeyslari Delphida yaratilgan dasturlarga hohlagan turdagi ma'lumotlar bazalari bilan bog'lanish imkoniyatini beradi. Masalan, ADO dan foydalanuvchi dastur SQL korporativ serveridagi ma'lumotlarga ham, electron jadvallarga ham, local MBBT larga ham murakkab so'rovlarni bajara oladi. ADO orqali jo'natilgan xoxlagan SQL so'rovi o'rinlanadi.

Bu qanday qilib amalgam oshirilib turadi? Buning hammasini ADO mexanizmi va OLE DB interfeyslari amalgam oshirib turadi. OLE DB - bu ma'lumotlar ustida bajariluvchi standart amallar va ma'lumotlar olmashishni amalgam oshiruvchi maxsus funktsiyalar majmuasi bo'lib topiladi. ADO terminologiyasi bo'yicha, xoxlagan ma'lumotlar manbayi (ma'lumotlar bazasi, electron jadval, fayl) ma'lumotlar ombori deb tushuniladi, ushbu ma'lumotlar ombori bilan dastur ma'lumotlar provayderi yordamida ish olib boradi. Dasturning minimal komponentlar majmuasi o'z ichiga bog'lanish obyektini, ma'lumotlar majmuasi obyektini va so'rovlar protsessori obyektini oladi.

BDE va BDE Administrator

So'nggi paytda Borland kompaniyasi ma'lumotlar bazalari bilan osh olib borishda SQL texnologiyalarini oldiga chiqarib atirganligiga qaramasdan, BDE (Borland Database Engine – Borland ma'lumotlar bazalari mehanizmi) ancha murakkab bo'lmagan ma'lumotlar bazalarini yaratishda haliyam keng ravishda qo'llanilmoqda. Bunga sabab faqat ushbu texnologiyaning osonligi emas-ki, ammo xozirgi kundayam BDE ni foydalanuvchi dasturlar ishlab kelmoqda va ushbu dasturlarni sozlab turish lozim.

BDE qo'llaydigan ma'lumotlar bazalari parametrlarini sozlash uchun va ushbu ma'lumotlar bazasiga psevdonim (Alias) yaratish uchun Delphi muhitiga kiruvchi maxsus utilita – BDE Administrator qo'llaniladi. Bu utilitaning ishchi maydoni 2 sahifadan iborat – Databases va Configuration (1-rasm).



1-rasm. BDE Administrator Databases sahifasi oynasi

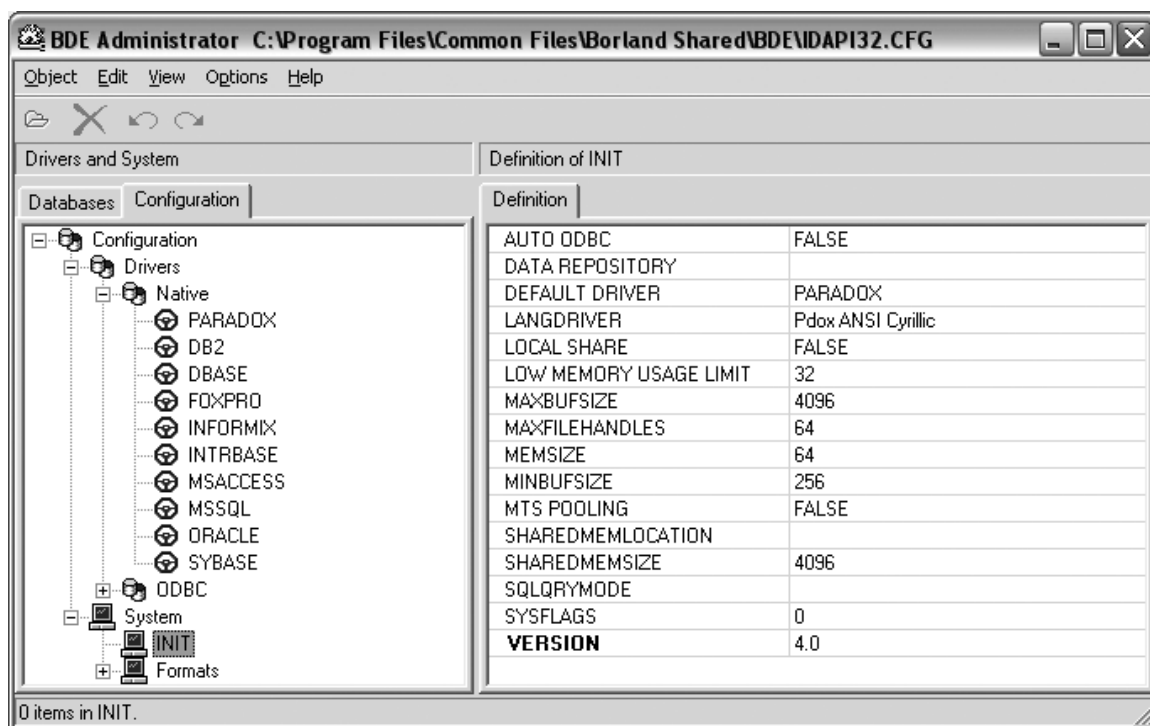
Databases sahifasida ma'lumotlar bazasiga oid alias larning tizimi joylashgan. Delphi muhiti va BDE o'rnatilgach birnechta aliaslar yaratiladi, masalan, DBDEMOS, DefaultDD va IBLocal. Yangi alias yaratish uchun Object menyusidan New bo'limini tanlash lozim. Natijada biz MBBT tipini tanlay oladigan oyna xosil bo'ladi. Ko'pchilik vosiyatlarda Standard tipi tushuniladi, bu ma'lumotlar bazasini yaratishda Paradox drayveridan qo'llanishni anglatadi. Undan keyin tizimda Standard1 deb nomlanuvchi yangi element xosil bo'ladi, o'ng tomonda esa, xossalarni aniqlash maydonida (Definition) barcha tahrirlash mumkin bo'lgan parametrlar paydo bo'ladi. Delphi uchun standart tipi uchun bu xossalar quyidagicha bo'ladi: tip (Type), MBBT drayveri (Default driver), haqiqiy sonlarni turlandirish rejimi (Enable BCD) va ma'lumotlar bazasi joylashgan

katalog (Path). Yangi aliasni duzgach uning ismini ham o'zgartirshga imkoniyat mavjud – buning uchun Object menyusidan Rename buyrug'I tanlanadi.

Aytaylik, bizning aliasimiz DATA1, ma'lumotlar bazasi esa C diskida Data katalogida joylashgan bo'lsin. Dasturimiz C:\Data katalogidagi Paradox ma'lumotlar bazasiga so'rov jonatish uchun faqat ushbu aliasni ko'rsatsak bo'ladi.

BDE Administratorning ikkinchi sahifasi - Configuration – Drivers guruhidagi MBBT drayverlarin va System guruhidagi sistemalik o'zgaruvchilarni sozlash imkoniyatini beradi. Drayverlar BDE o'zi boshqaruvchi (Native) va ODBC orqali boshqariluvchi bo'lib ikkiga bo'linadi (Open Database Connectivity – ochiq ma'lumotlar bazalari biriktirilishi).

Sozlamalar ichida Langdriver opsiyasini ko'rib chiqamiz, bu opsiya ham sistemalik sozlamalarda (INT guruhi, 2-rasmga qora), ham har bir MBBT uchun alohida mavjud bo'ladi. Kirill harflari bilan to'g'gri ish olib boorish uchun "Pdow ANSI Cyrillic" ma'nosini o'rnatish kerak, u Windows-1251 kodirovkasiga mos keladi, bu ma'no ham umumiy sozlamalarda (INIT), ham qo'llaniluvchi MBBT drayveri (masalan, Paradox) uchun ham o'rnatildai.



2-rasm. BDE Administrator da sistemalik sozlamalar

Database Desktopda jadvallar yaratish

Ma'lumotlar bazalari bilan ish olib boruvchi dastur yaratganda jadvallar bilan ish olib boorish uchun Delphi muhiti ichiga kiruvchi Database Desktop dasturidan foydalansa bo'ladi. Database Desktop yordamida jadvallar hosil qilish, o'larga yangi ma'lumotlar kiritish, ularni o'zgartirish va so'rovlar bilan ish olib borishga bo'ladi.

Yangi jadval yaratish uchun File > New > Table buyrug'ini ishga tushirish kerak. Bu yerda hosil qilinuvchi sahifa turini ko'rsatish mavjud bo'ladi, ko'pchilik vasiyatda Paradox 7 formati tanlanadi, biz ham ushbu formatdan foydalanamiz. Keyin esa jadval strukturasini aniqlash oynasi paydo bo'ladi (3-rasm), bu yerda jadval parametrlari yaratiladi va ular sozlanadi, masalan, jadval maydonlari, indekslari, parollar, shartlar va hokazo.

	Field Name	Type	Size	Key
1				

Enter a field name up to 25 characters long.

Table properties:
Validity Checks
Define...
☐ 1. Required Field
2. Minimum value:
3. Maximum value:
4. Default value:
5. Picture:
Assist...
Borrow... Save As... Cancel Help

3-rasm. Jadval strukturasini aniqlash

II-BOB. DELPHI MUHITIDA PEDAGOGIK DASTURIY VOSITALARNI YARATISH

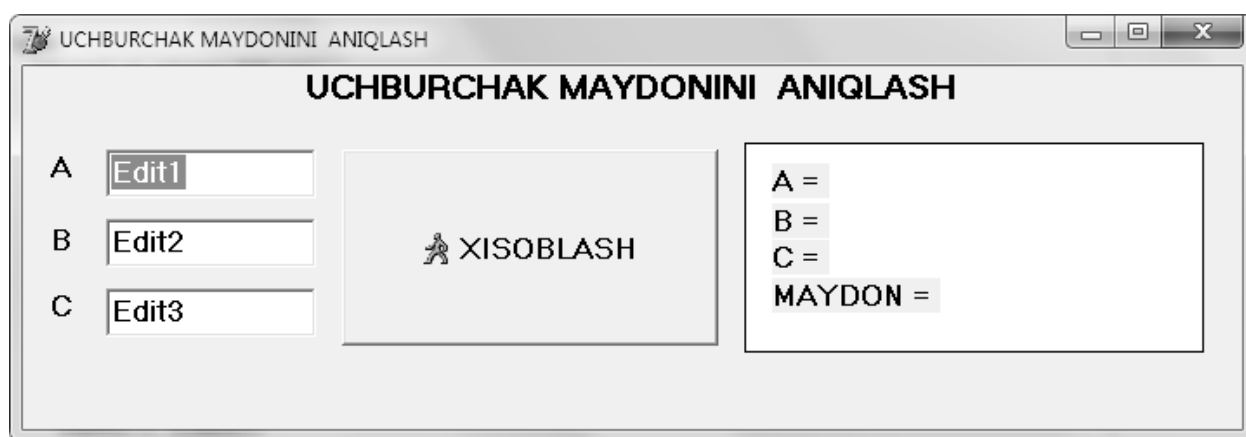
2.1. MAKTAB KURSI MATEMATIKASIGA BOG'LIQ MASALALARNI YECHISHGA MULJALLANGAN DASTURIY VOSITALAR YARATISH

Ushbu bo'limda biz maktab matematika kursiga oyid o'rgatuvchi dasturiy vositalarni Delphi muhitida yaratilishni bosqichma-bosqich ko'rib chiqamiz. Magistrlik dissertatsiya ishimizning qo'shimchalar sahifalarida barcha yaratiladigan dasturlar listinglari berilgan.

Uchburchak maydonini aniqlash dasturi

Dastur o'quvchilardan uchburchakning uchta tomonini kiritishini so'raydi. Ushbu kiritilgan uchta tomon qiymatlariga ko'ra dastur uchburchak maydonini chiqarib beradi.

Dasturning tashqi ko'rinishi 1-rasmda ko'rsatilgan. Formaga 3 ta Edit, 8 ta Label, 1 Bitbtn va 1 Shape joylashtirilgan (1-rasm).

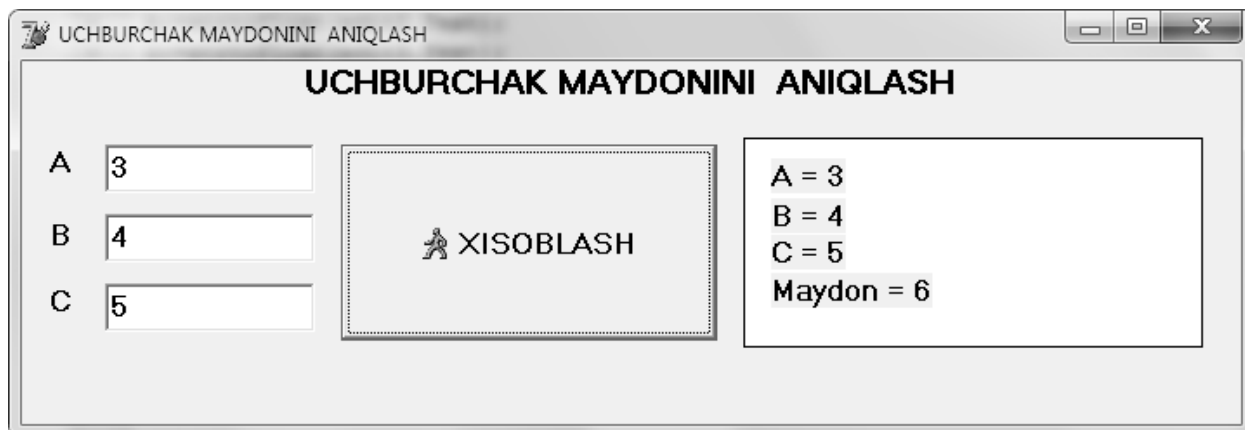


1-rasm. Dasturning tashqi ko'rinishi

Dastur algoritmi juda oddiy. Lekin ushbu dastur 5-7 sinf o'quvchilari uchun foydali bo'ladi degan fikrdamiz.

Foydalanuvchi uchta Edit larga qiymatlar kiritgach, “Xisoblash” tugmasi bosiladi va natijada oq to’rtburchak ichida 4 ta natijja – foydalanuvchi tomonidan kiritilgan 3 ta qiymat va maydon chiqadi.

Maydon bizga maktab kursidan belgili Geron formulasiga muvofiq xisoblanadi. Barcha xisoblashlar, albatta, Bitbtn ning onClick voqeasida olib boriladi. 2-rasmda dasturning ishlash natijasini ko’ra olasiz (2-rasm).



2-rasm. Dastur ishi natijasi

Bitbtn1 ustiga ikki marta bosib, uning onClick voqeasiga quyidagi kodni yozamiz:

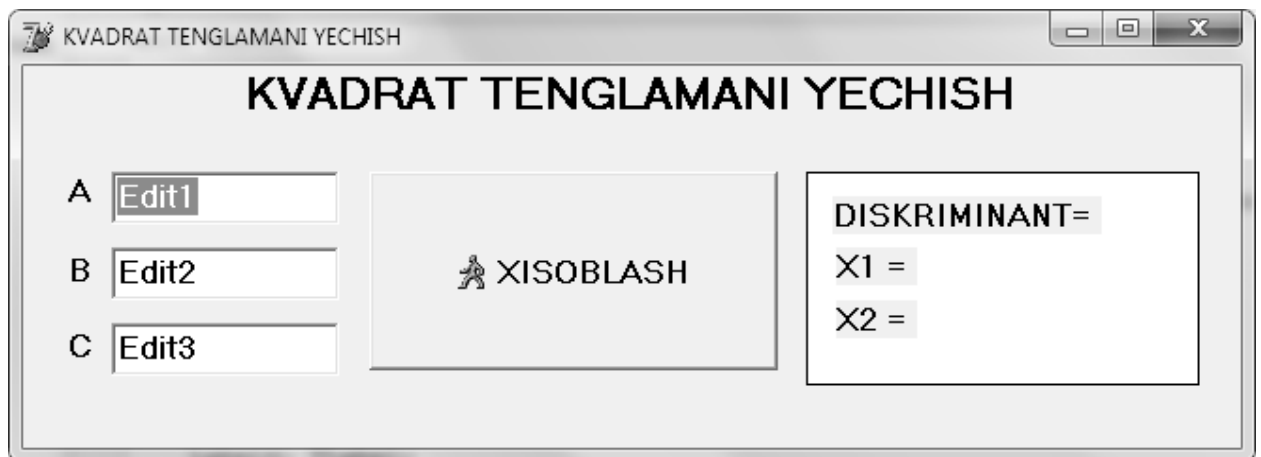
```
procedure TForm1.BitBtn1Click(Sender: TObject);
var
  a,b,c,s,p:real;
begin
  a:=strtofloat(edit1.Text);
  b:=strtofloat(edit2.Text);
  c:=strtofloat(edit3.Text);
  p:=(a+b+c)/2;
  s:=sqrt(p*(p-a)*(p-b)*(p-c));
  label5.Caption:='A = '+edit1.Text;
  label6.Caption:='B = '+edit2.Text;
  label7.Caption:='C = '+edit3.Text;
  label8.Caption:='Maydon = '+floattostr(s);
```

end;

Kvadrat tenglamani yechish dasturi

Dastur o'quvchilardan $ax^2+bx+c=0$ ko'rinishidagi kvadrat tenglamaning uchta koeffitsiyentini kiritishini so'raydi. Ushbu kiritilgan uchta qiymatlarga ko'ra dastur kvadrat tenglamaning (agar mavjud bo'lsa) ildizlarini chiqarib beradi.

Dasturning tashqi ko'rinishi 3-rasmda ko'rsatilgan. Formaga 3 ta Edit, 8 ta Label, 1 Bitbtn va 1 Shape joylashtirilgan (3-rasm).

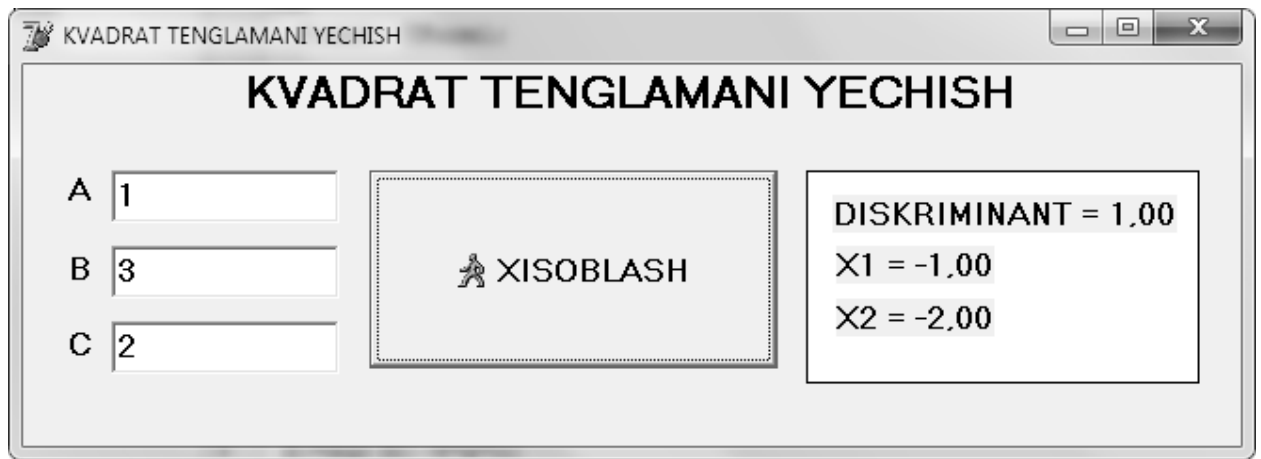


3-rasm. Dasturning tashqi ko'rinishi

Dastur algoritmi juda oddiy. Lekin ushbu dastur 5-7 sinf o'quvchilari uchun foydali bo'ladi degan fikrdamiz.

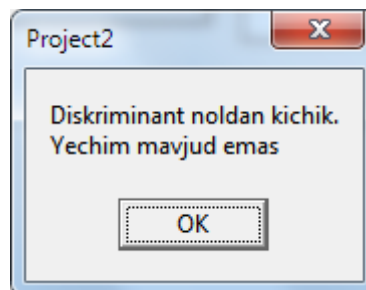
Foydalanuvchi uchta Edit larga qiymatlar kiritgach, "Xisoblash" tugmasi bosiladi va natijada oq to'rtburchak ichida 3 ta natijja – diskriminant va (agar mavjud bo'lsa) tenglamaning X1 va X2 ildizlari qiymatlari chiqadi.

Tenglamani yechish algoritmi bizga maktab kursidan belgili. Barcha xisoblashlar, albatta, Bitbtn ning onClick voqeasida olib boriladi. 4-rasmda dasturning ishlash natijasini ko'ra olasiz (4-rasm).



4-rasm. Dastur ishi natiyjasi

Agarda tenglama yechimga ega bo'lmasa unda quyidaicha oyna paydo bo'ladi (5-rasm):



5-rasm. Tenglama yechimga ega emas.

Bitbtn1 ustiga ikki marta bosib, uning onClick voqeasiga quyidagi kodni yozamiz:

```
procedure TForm1.BitBtn1Click(Sender: TObject);
var
  a,b,c,d,x1,x2:extended;
begin
  a:=strtofloat(edit1.Text);
  b:=strtofloat(edit2.Text);
  c:=strtofloat(edit3.Text);
  d:=sqr(b)-4*a*c;
  if d>=0 then
  begin
    x1:=(-b+sqr(d))/(2*a);
```

```

x2:=(-b-sqrt(d))/(2*a);
label5.Caption:='DISKRIMINANT = '+FLOATTOSTRf(D,ffFixed,4,2);
label6.Caption:='X1 = '+FLOATTOSTRf(X1,ffFixed,4,2);
label7.Caption:='X2 = '+FLOATTOSTRf(X2,ffFixed,4,2);
end
else
showmessage('Diskriminant noldan kichik.'+#13+'Yechim mavjud emas');
end;

```

2.2. O'QUVCHILAR BILIM DARAJASINI ANIQLOVCHI DASTURIY VOSITALAR YARATISH (MAKTAB KURSI MATEMATIKASIGA OID)

Ta'lim tizimiga axborot texnologiyalardan foydalanuvchi qismlarni kiritish protsessida eng qiziqarli bo'limlardan biri – bu o'quvchilarning egallagan bilim darajasini aniqlovchi dasturiy vositalarni yaratish bo'lib topiladi. Ko'pchilik xolatlarda bunday vosiyatlar deganimizda biz o'quvchilardan test topshiriqlarini avtomat ravshda o'tkazish imkoniyatini beruvchi dasturlarni tushunamiz.

Bunday dasturiy vositalarni maktab kursi matematikasiga oid yaratish protsessini ko'rib chiqaylik. Biz ushbu bo'limda maktab geometriyasi va algebrasi kurslaridan besh-olti savoldan berib, o'quvchining bilim darajisini aniqlaydigan dasturiy ta'minotni yaratishni bosqichma-bosqich ko'rib chiqamiz.

Bunday dasturiy ta'minot yaratilgan paytda biz oldin birnechta savolga javob berishimiz kerak. Ushbu savollar quyidagilardan iborat:

1. Savollar qanaqa formatda bo'ladi? (faqat tekst, yoki grafika ham aralashganma)
2. Savolni “o'tkazib yuborib”, so'ng ushbu savolga qaytib kelish imkoni bormi yoki yo'qmi
3. Baxolash natijalari qanday formada bo'ladi? (html, excel fayl yoki bloknotda)

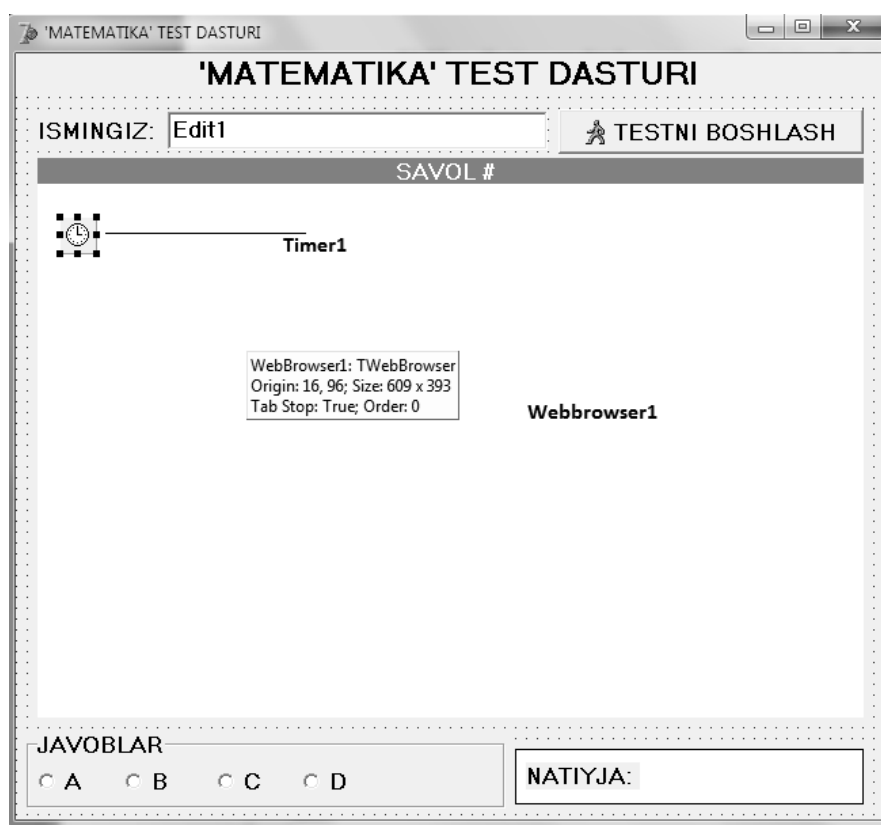
Ushbu savollarga asoslana o'tirib, biz ikkita test olis dasturini ishlab chiqamiz. Birinchi test olish dasturi maktab matematikasiga oid bo'ladi, ikkinchi dasturimiz esa maktab geometriyasiga oyid bo'ladi. Ikki dastur ham grafikadan foydalanish imkoniyatini beradi, natijalar esa html formada saqlanadi. Testdan o'tish vaqt bo'yicha chegaraga ega emas, shu sababli har bir savol ketma-ket yechilishi lozim.

Savollar oldinnan Word tekst muharririda ishlab chiqilgan, har bitta savol – bo'lak hujjat, barchasi web-hujjat formatida saqlangan (Word muharririda Fayl-Soxranit' kak Web-stranitsu buyrug'i orqali).

Har bir dasturimiz 10 ta savoldan iborat.

'MATEMATIKA' test olish dasturi

Ushbu dastur maktab matematikasining 5-sinfi bo'yicha 10 ta savoldan iborat. Dastur tashqi ko'rinishi 1-rasmda ko'rsatilgan:



1-rasm. Dastur tashqi ko'rinishi

Dasturni xosil qilishni bosqichma-bosqich ko'rib chqamiz. Delphi muhitini ishga tushuramiz va quyidagilarni bajaramiz:

1. Formaga Label componentasin o'rnatamiz.
2. Uning xossalarida "Align=Top", "Caption='MATEMATIKA'TEST DASTURI'" va "Alignment=Center" ma'nolarini beramiz.
3. "Internet" bo'limidan WebBrowser1 komponentini o'rnatamiz
4. Label2 ni formaga joylashtirib, "Caption='ISMINGIZ'" xossasini o'zgartiramiz
5. Edit1 komponentasini Label2 yoniga qo'yamiz
6. "Additional" bo'limidan BitBtn1 komponentini joylashtiramiz va "Kind=Ignore", "King=Custom", "Caption='TESTNI BOSHLASH'" ma'nolarini xossalarga o'rnatamiz
7. "Standard" bo'limidan 1 RadioGroup va 4 RadioButton o'rnatamiz, RadioButton1 da "Checked=True", "Caption='A'" xossalarini, boshqa RadioButtonlarga esa mos ravichda 'B', 'C' va 'D' ma'nolarni Caption xossalariga qo'yib chiqamiz
8. "Additional" bo'limidan Shape1 ni formaga WebBrowser1 pastiga joylashtiramiz va uning ustiga "Standard" bo'limidan Label3 ni qo'yamiz
9. Label3 ning "Caption" xossasigga "Natiyja" degan ma'no beramiz
10. "System" bo'limidan Timer1 ni o'rnatamiz

Ushbu orqali biz 1-rasmdagi formaga ega bo'lamiz. Shu bilan forma dekorativ ishlari tugaydi.

Endi savollarni tayyorlaymiz. Savollarni Word tekst muharririda web-sahifa rejimida saqlash lozim. Barcha savollarni proektning 'test' katalogi ichiga joylashtirish lozim.

Testning ishlash jarayonini 2-rasmda ko'rsak bo'ladi:

2-rasm. Test ishlash jarayoni

Test olish jarayoni to'lig'i bilan Timer1 elementining onTimer voqeasida bo'ladi. Demak, biz Timer1 komponentining onTimer voqeasi ichiga quyidagilarni yozamiz:

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
  inc(i);
  if i<=10 then
    begin
      webbrowser1.Navigate(extractfilepath(application.ExeName)+'test\'
+inttostr(i)+' .htm');
      label4.Caption:='SAVOL #' +inttostr(i);
      timer1.Enabled:=false;
    end
end
```

```
else finish;  
end;
```

Shuning bilan birga har bitta javob tanlanganda savollar olmashib turushi amalgam oshiriladi va berilgan javob to'g'ri yoki to'g'ri emasligi tekshirilib turadi:

```
procedure TForm1.RadioButton1Click(Sender: TObject);  
begin  
if key[i]=1 then inc(k);  
timer1.Enabled:=true;  
end;
```

```
procedure TForm1.RadioButton2Click(Sender: TObject);  
begin  
if key[i]=2 then inc(k);  
timer1.Enabled:=true;  
end;
```

```
procedure TForm1.RadioButton3Click(Sender: TObject);  
begin  
if key[i]=3 then inc(k);  
timer1.Enabled:=true;  
end;
```

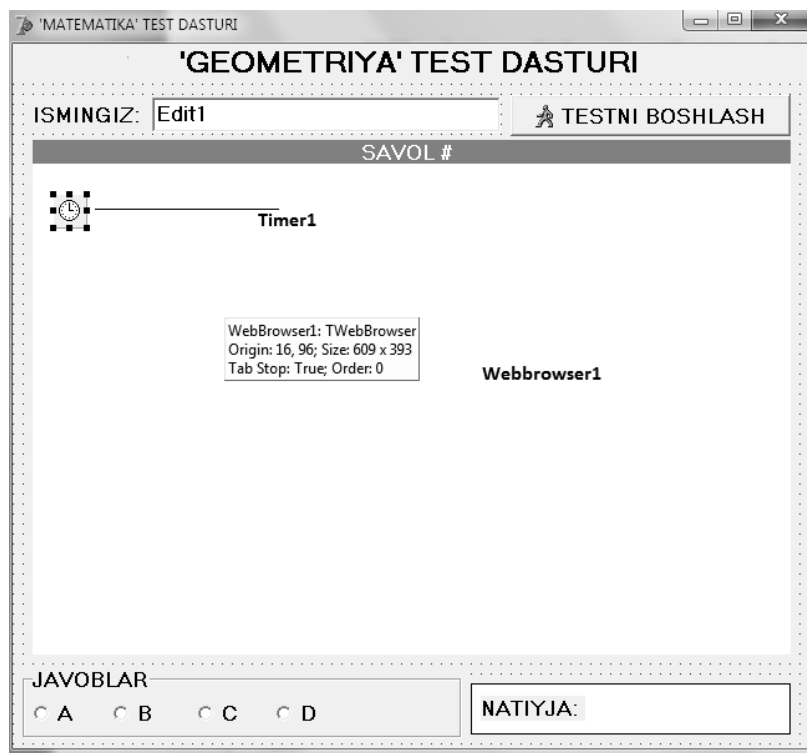
```
procedure TForm1.RadioButton4Click(Sender: TObject);  
begin  
if key[i]=4 then inc(k);  
timer1.Enabled:=true;  
end;
```

Bu yerda key – test kaliti massivi. K esa – to'g'ri javoblar xisoblagichi. Natijalar finish protsedurasida beriilgan:

```
procedure TForm1.finish;  
begin  
    timer1.Enabled:=false;  
    if k<6 then label3.Caption:='NATIYJA: 2!';  
    if k=7 then label3.Caption:='NATIYJA: 3!';  
    if k=8 then label3.Caption:='NATIYJA: 4!';  
    if k>8 then label3.Caption:='NATIYJA: 5!';  
end;
```

‘GEOMETRIYA’ test olish dasturi

Ushbu dastur maktab geometriyasining 7-sinfi bo'yicha 10 ta savoldan iborat. Dastur tashqi ko'rinishi 3-rasmda ko'rsatilgan:



3-rasm. Dastur tashqi ko'rinishi

Dasturni xosil qilishni bosqichma-bosqich ko'rib chqamiz. Delphi muhitini ishga tushuramiz va quyidagilarni bajaramiz:

11. Formaga Label componentasin o'rnatamiz.
12. Uning xossalarida "Align=Top", "Caption='GEOMETRIYA'TEST DASTURI'" va "Alignment=Center" ma'nolarini beramiz.
13. "Internet" bo'limidan WebBrowser1 komponentini o'rnatamiz
14. Label2 ni formaga joylashtirib, "Caption='ISMINGIZ'" xossasini o'zgartiramiz
15. Edit1 komponentasini Label2 yoniga qo'yamiz
16. "Additional" bo'limidan BitBtn1 komponentini joylashtiramiz va "Kind=Ignore", "King=Custom", "Caption='TESTNI BOSHLASH'" ma'nolarini xossalarga o'rnatamiz
17. "Standard" bo'limidan 1 RadioGroup va 4 RadioButton o'rnatamiz, RadioButton1 da "Checked=True", "Caption='A'" xossalarini, boshqa RadioButtonlarga esa mos ravichda 'B', 'C' va 'D' ma'nolarni Caption xossalariga qo'yib chiqamiz
18. "Additional" bo'limidan Shape1 ni formaga WebBrowser1 pastiga joylashtiramiz va uning ustiga "Standard" bo'limidan Label3 ni qo'yamiz
19. Label3 ning "Caption" xossasigga "Natiyja" degan ma'no beramiz
20. "System" bo'limidan Timer1 ni o'rnatamiz

Ushbu orqali biz 1-rasmdagi formaga ega bo'lamiz. Shu bilan forma dekorativ ishlari tugaydi.

Endi savollarni tayyorlaymiz. Savollarni Word tekst muharririda web-sahifa rejimida saqlash lozim. Barcha savollarni proektning 'test' katalogi ichiga joylashtirish lozim.

Testning ishlash jarayonini 4-rasmda ko'rsak bo'ladi:

'MATEMATIKA' TEST DASTURI

ISMINGIZ: Asal TESTNI BOSHLASH

SAVOL #1

1. Qaysi qatorda raqamlar to`g`ri ko`rsatilgan?

a) 1,3,6,10,14,16,72 b) 0,1,2,3,4,5,6,7,8,9 s) 12,2,4,8,90,11,19

JAVOBLAR

☐ A ☐ B ☐ C ☐ D

NATIYJA:

4-rasm. Test ishlash jarayoni

Test olish jarayoni to`lig`i bilan Timer1 elementining onTimer voqeasida bo`ladi. Demak, biz Timer1 komponentining onTimer voqeasi ichiga quyidagilarni yozamiz:

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
inc(i);
if i<=10 then
begin
webbrowser1.Navigate(extractfilepath(application.ExeName)+'test\'
+inttostr(i)+'.htm');
label4.Caption:='SAVOL #' +inttostr(i);
timer1.Enabled:=false;
end
```

```
else finish;  
end;
```

Shuning bilan birga har bitta javob tanlanganda savollar olmashib turushi amalgam oshiriladi va berilgan javob to'g'ri yoki to'g'ri emasligi tekshirilib turadi:

```
procedure TForm1.RadioButton1Click(Sender: TObject);  
begin  
if key[i]=1 then inc(k);  
timer1.Enabled:=true;  
end;
```

```
procedure TForm1.RadioButton2Click(Sender: TObject);  
begin  
if key[i]=2 then inc(k);  
timer1.Enabled:=true;  
end;
```

```
procedure TForm1.RadioButton3Click(Sender: TObject);  
begin  
if key[i]=3 then inc(k);  
timer1.Enabled:=true;  
end;
```

```
procedure TForm1.RadioButton4Click(Sender: TObject);  
begin  
if key[i]=4 then inc(k);  
timer1.Enabled:=true;  
end;
```

Bu yerda key – test kaliti massivi. K esa – to'g'ri javoblar xisoblagichi. Natijalar finish protsedurasida beriilgan:

```
procedure TForm1.finish;  
begin  
    timer1.Enabled:=false;  
    if k<6 then label3.Caption:='NATIYJA: 2!';  
    if k=7 then label3.Caption:='NATIYJA: 3!';  
    if k=8 then label3.Caption:='NATIYJA: 4!';  
    if k>8 then label3.Caption:='NATIYJA: 5!';  
end;
```

III-bob. Pedagogik dasturiy vositalarni ma'lumotlar bazalarin boshqarish tizimlari bilan integratsiyalash

3.1. Pedagogik dasturiy vositalarning ma'lumotlar bazasini yaratish

Biz ushbu ishimizda “Algortimlash asoslari” mavzusidagi electron resurslar ma'lumotlar bazasini yaratishni maqsad qilib qo'yganmiz. Ushbu resurs nima uchun kerakli bo'ladi?

Birinchi navbatda, ushbu dasturimiz, albatta, algoritmlashtirish asoslari kabi fanni o'rganuchi talabalar yoki dasturlashga qiziquvchi hohlagan inson uchun bizning dasturiy ishlanmamiz katta foyda keltiradi deb aytsak ham bo'ladi. Ushbu dasturni yaratish uchun biz ADO texnologiyasidan foydalanganmiz. Keeling ushbu texnologiya haqida qisqacha tushuncha berib ketayik.

ADO (ActiveX Data Objects) Delphi dasturlash muhitining 5-chi versiyadisan boshlab paydo bo'ldi. Ushbu texnologiya Microsoft tomonidan yaratilgan bo'lib, u ishlash printsiplari BDE ga uxshash bo'lib keladi.

ADO texnologiyasidan Delphi dasturlash muhitida foydalanishga misol sifatida quyidagicha masalani yechib ko'rsatamiz.

Mayli bizda qandaydir MS Access MBBT da yaratilgan quyidagicha jadval berilgan bo'lsin:

№	Ismi	Sharifi
1	Yodgorbek	Risqulov
2	Elyor	Doniyorov

Ushbu jadvalni biz Delphi proektida DBGrid komponenti orqali chiqarishimiz kerak bo'lsin.

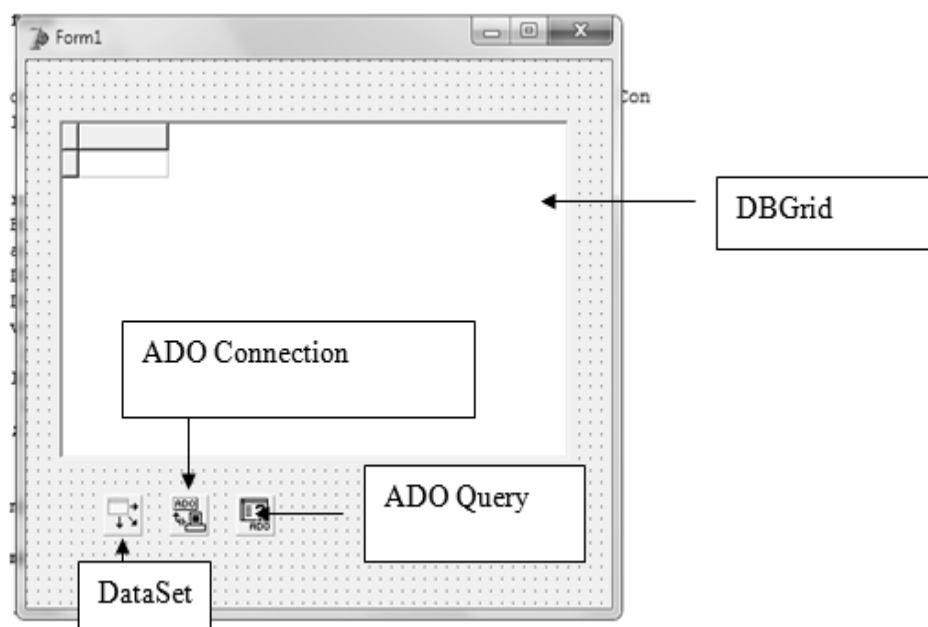
Delphidi ishga tushurib, File-New-Application orqali yangi proyekt hosil qilamiz. Endi formaga quyidagi komponentlardi o'rnatamiz (1-rasm):

TDBGrid

TDataSource

TADOConnection

TADOQuery



1-rasm

Ushbu komponentlari ko'rib chiqamiz:

TDBGrid DataControls sarlavhasida joylashgan bo'lib, u ma'lumotlar bazasidagi jadvalni ko'rsatish uchun qo'llaniladi

TDataSource Data Access sarlavhasida joylashgan bo'lib, u DBGrid bilan ma'lumotlar bazasini bog'lash uchun qo'llaniladi

TADOConnection ADO sarlavhasida joylashgan bo'lib, u bizning ma'lumotlar bazamizni qandaydir provayder bo'yicha bog'lanishini ishga oshirib turadi

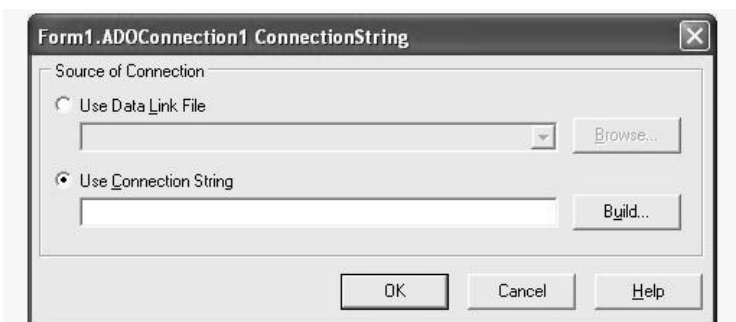
TADOQuery ham ADO sarlavhasida joylashgan bo'lib, u ma'lumotlar bazasidan kerakli ma'lumotni olishga yordam berib turadi.

Mayli bizning yuqoridagi jadvalimiz nomi Student bo'lsin. Ushbu jadval to'g'ri bizning proektimizda ko'rinishi uchun kerakli amallarni bajarish lozim:

1. TAdoQuery komponentining Connection hossasi orqali ushbu maydonda TAdoConnection ni tanlash orqali bog'lanishni o'rnatamiz
2. TDataSource komponentining DataSet hossasida esa TadoQuery ni ko'rsatamiz
3. TDBGrid ning DataSet xossasida TdataSet ni tanlaymiz

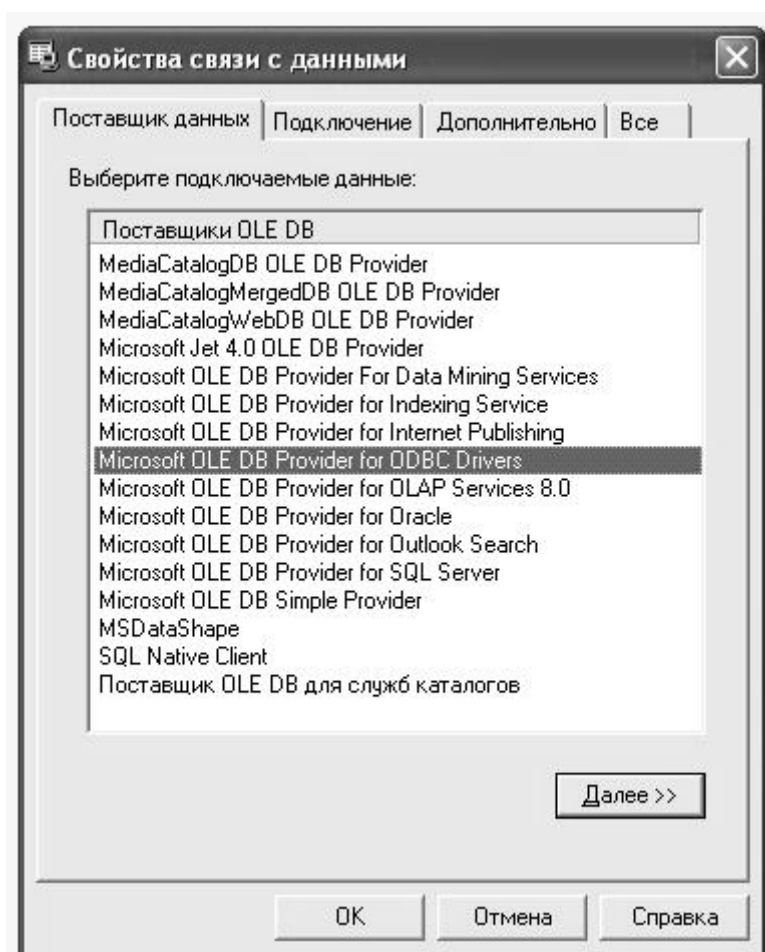
Endi esa biz AdoConnection komponentini sozlashimiz kerak. Buning uchun quyidagilarni amalga oshiramiz:

1. LoginPromt xossasiga False ma'nosini beramiz (ma'lumotlar bazasiga qo'shilgan paytimizda parol so'raydigan oyna xosil bo'lmasligi uchun)
2. ConnectionString xossasi yonindagi ... tugmasiga bo'sib, 2-rasmdagi oynani xosil qilamiz:



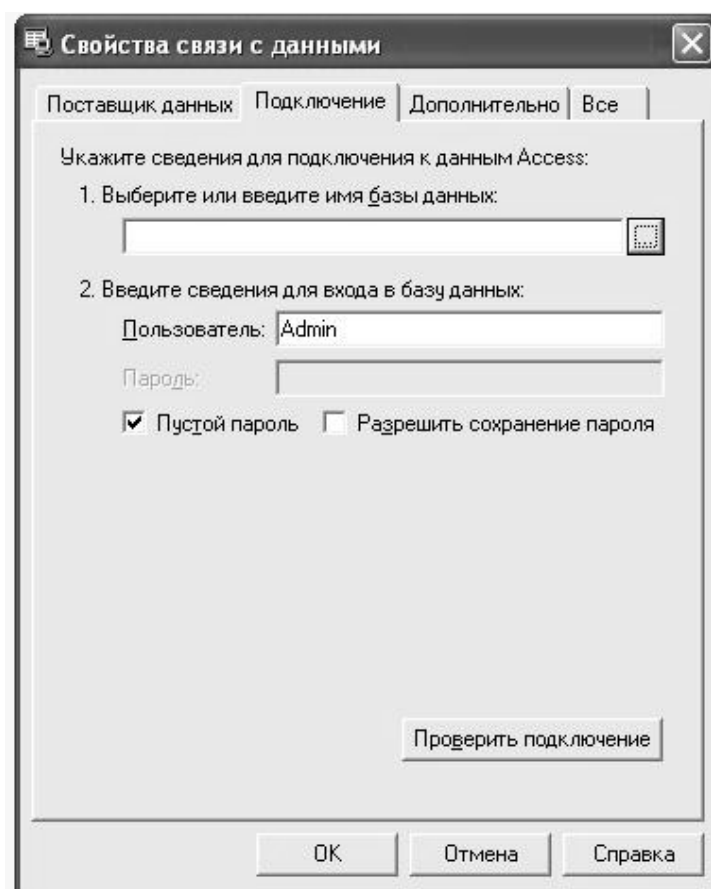
2-rasm

3. Ushbu oynada “Build” tugmasini bosamiz va 3-rasmdagi oynani xosil qilamiz:



3-rasm

4. Ushbu oynadan Microsoft Jet 4.0 OLE DB Provider ni tanlaymiz va Daleye tugmasiga bosamiz (4-rasm)



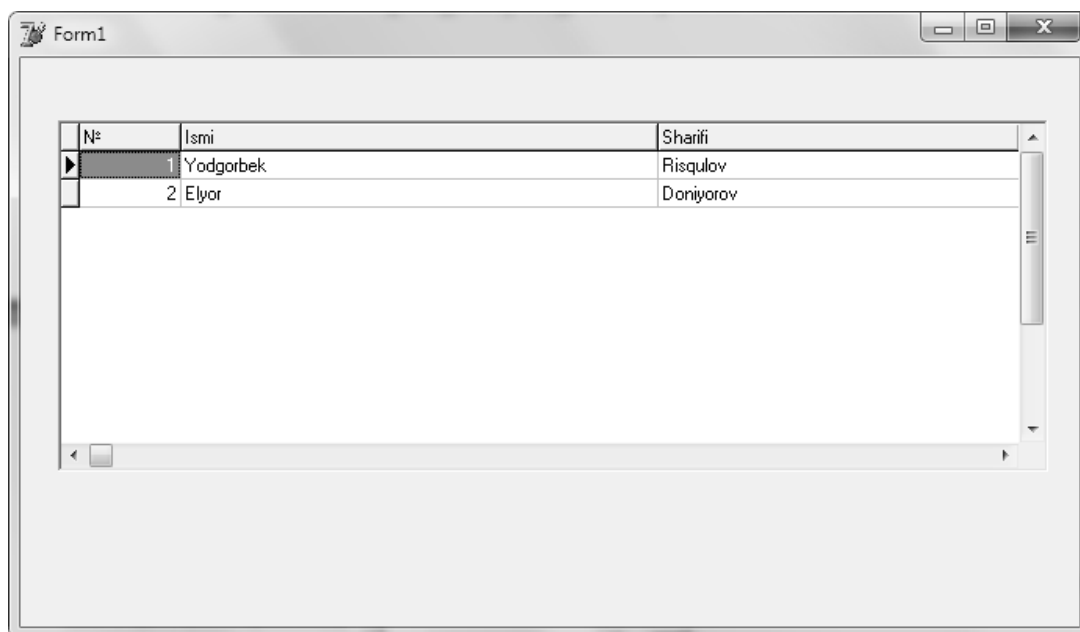
4-rasm

5. Ushbu oynada bizning ma'lumotlar bazamizga yo'lni ko'rsatamiz va foydalanuvchi nomiga Admin deb yozamiz. Agarda bizning ma'lumotlar bazamiz proekt papkasida joylashgan bo'lsa, bunday sharoyitda faqat ma'lumotlar bazasining kengaytmasi va nomi ko'rsatilsa yetarli bo'ladi/ masalan BD\mydb.mdb.
6. Endi esa «Ок» tugmasiga bosib, TADOConnection ning Mode xossasiga cmShareDenyNone ma'nosini ko'rsatamiz.

Bog'lanishni amalgam oshirdik. Endi ma'lumotlar javalimizda ko'rinishi uchun formamizning onCreate voqeasiga quyidagi kodni yozsak etarli bo'ladi:

```
procedure TForm1.FormCreate(Sender: TObject);  
begin  
  ADOQuery1.SQL.Clear;  
  ADOQuery1.SQL.Add('SELECT * FROM Student');  
  ADOQuery1.Active:=True;  
end;
```

Dasturning ishlash natijasi quyida ko'rsangiz bo'ladi (5-rasm):



The screenshot shows a standard Windows application window with the title 'Form1'. Inside the window, there is a table with three columns: '№', 'Ismi', and 'Sharifi'. The table contains two rows of data. The first row has the values '1', 'Yodgorbek', and 'Risqulov'. The second row has the values '2', 'Elyor', and 'Doniyorov'. The table is displayed with a light gray background and black text. The window has a standard Windows XP-style title bar with minimize, maximize, and close buttons.

№	Ismi	Sharifi
1	Yodgorbek	Risqulov
2	Elyor	Doniyorov

5-rasm

Dasturning to'liq kodini "Qo'shimchalar" bo'limida keltirib o'tganmiz.

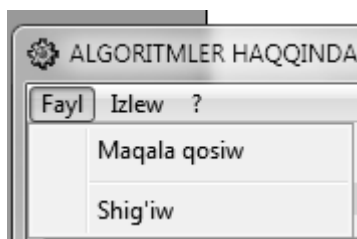
3.2. O'QUVCHILAR BILIM DARAJASINI RIVOJLANISHTIRADIGAN MA'LUMOTLAR BAZASINI YARATISH

Bizning ushbu dasturimiz quyidagi interfeysga ega bo'ladi (1-rasm):



1-rasm

Dastur quyidagi menyu satrlaridan iborat (2-rasm):



2-rasm

Fayl bo'limida maqola qo'shishga (6-rasm) va dasturdan chiqishga imkon beruvchi buyruqlar joylashgan. "Izlew" bo'limi esa ma'lumotni izlash oynasini hosil qiladi (3-rasm):

Statyalardan mag'li'wmat izlew

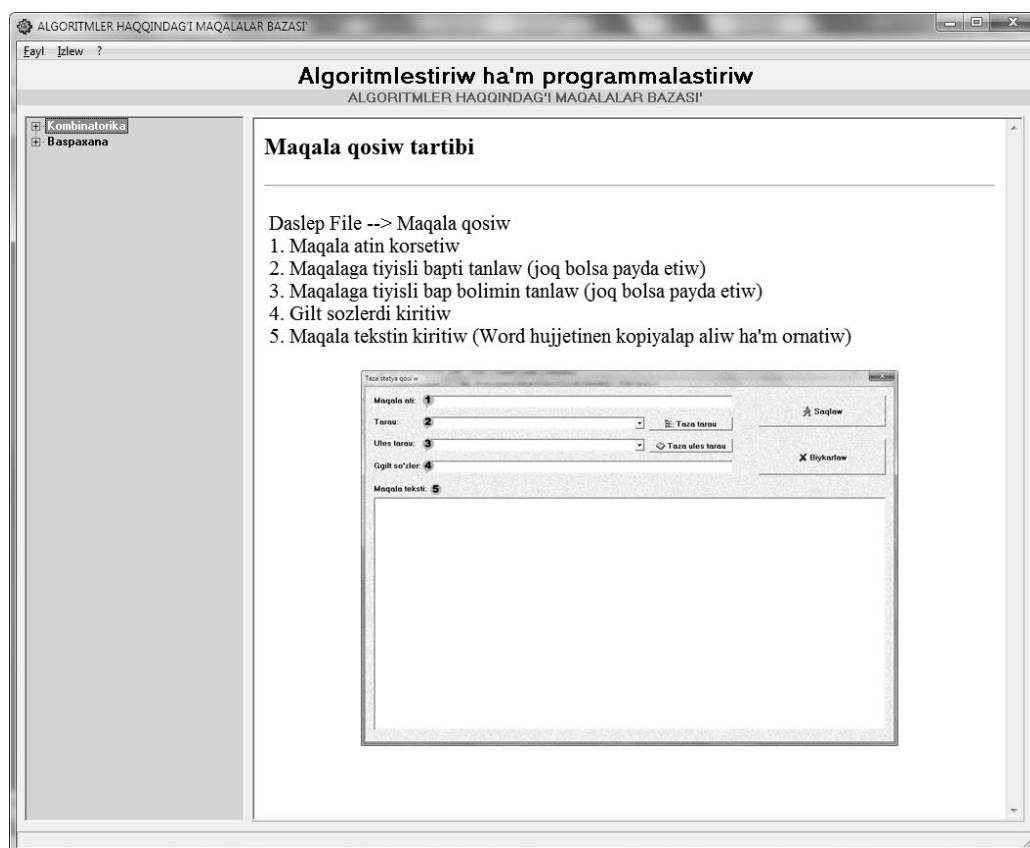
Izleniwshi so'z:

Izlew turi

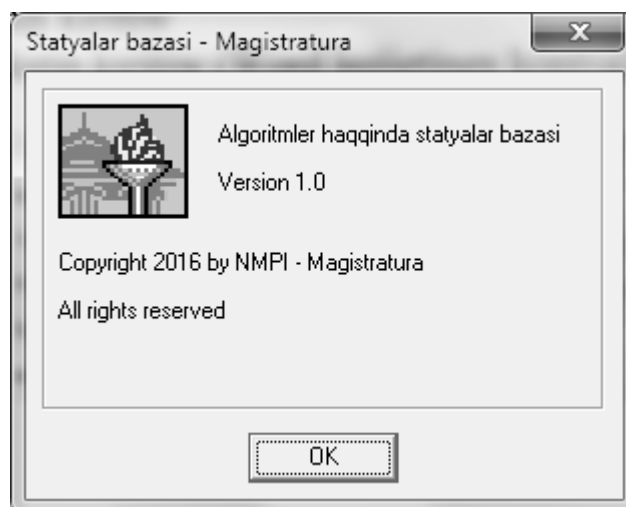
☒ Statya atamasinda ☐ Gilt so'zlerden

3-rasm

“?” bo’limida maqola qo’shish haqida yordamchi ma’lumot chiqarish (4-rasm) va dastur avtorlari haqida ma’lumot chiqarish (5-rasm) imkonini beruvchi buyruqlar joylashgan



4-rasm



5-rasm

Taza statiya qosi'w

Maqala ati:

Taraui:

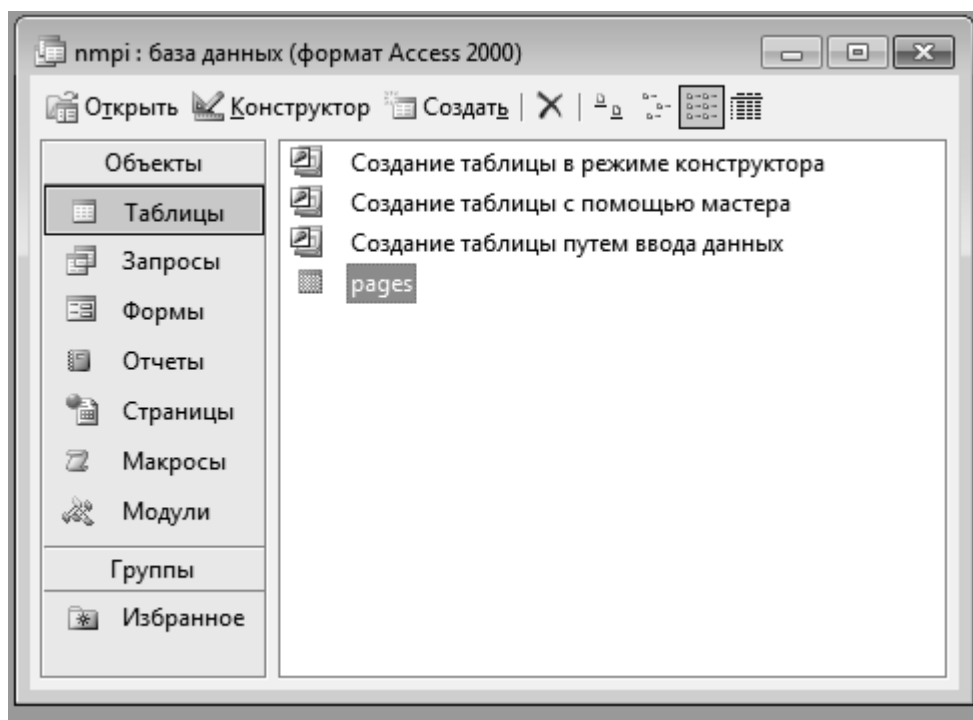
Ules taraui:

Ggilt so'zler:

Maqala teksti:

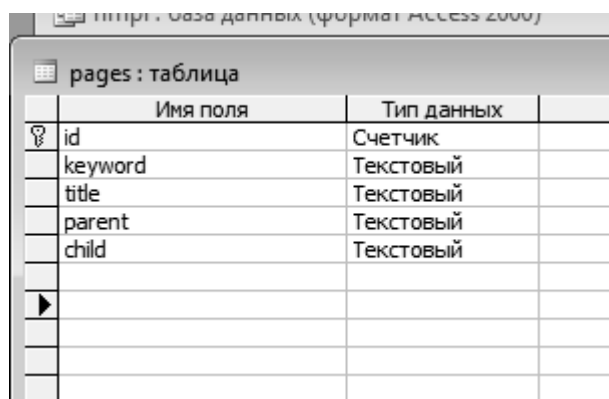
6-rasm

Ushbu dasturli tayorlashda biz ADO texnologiyasidan foydalanganmiz. Ma'lumotlar bazasi esa Microsoft Access 2003 dasturida yaratilgan (7-rasm).



7-rasm

7-rasmdan ko'rinib turibdi, ma'lumotlar bazamiz nomi – nmpi. Ushbu ma'lumotlar bazamizda faqat bitta pages nomli jadvalimiz mavjud. Jadval strukturasi 8-rasmda berilgan:



	Имя поля	Тип данных
PK	id	Счетчик
	keyword	Текстовый
	title	Текстовый
	parent	Текстовый
	child	Текстовый

8-rasm.

Keling ushbu dastur bilan ish olib borishni ko'rib chiqayik.

Eng birinchi marta dastur ishga tushirilganida chap tomondagi menyu bo'sh ko'rinishga ega bo'ladi (9-rasm):



9-rasm

Dasturga yangi maqola qo'shish kerak. Buning uchun Fayl-Maqola qo'shish buyrug'ini amalgam oshirib, 6-rasmdagi oynani xosil qilamiz. Ushbu oynaga diqqat bersak, bizdan quyidagi maydonlar to'ldirilishi so'ralgan:

- 1) Maqola nomi
- 2) Maqola tegishli bo'lgan guruh (chap tomonda chiqadi)
- 3) Maqola tegishli bo'lgan kichik guruh (agar tegishli bo'lsa)

4) Kalit so'zlar (maqolada uchrashadigan kalit so'zlar)

5) Maqola matni (Word dasturida tayorlangan maqolani ko'chirib, ichiga solamiz)

Ushbu amallarni bajarib bo'lgach, "Saqlash" tugmasini boshish lozim bo'ladi. Natijada maqola bazaga qo'shilib, o'ngga tegishli chap tomonda menyu paydo bo'ladi. Agar biz maqolani saqlashni hohlamasak, "Bekorlash" tugmasini boshish orqali maqola kiritish oynasini yopib tashlaymiz, maqola bazada saqlanmaydi.

10-12 rasmlarda biz dasturga "Graflar" nomli guruhiga kiruvchi va shu guruhning ichidagi "Eng qisqa yo'l" nomli kichik guruh a'zosi bo'lgan "Deykstra algoritmi" nomli maqolani qo'shganimiz qadamba-qadam ko'rinib turibdi.



10-rasm

Taza statya qosi'w

Maqala ati: Deykstra algoritmi

Tarau: Graflar Taza tarau

Ules tarau: Eng kichik yo'l Taza ules tarau

Ggilt so'zler: Deykstra, algoritm

Maqala teksti:

Deykstra algoritmi haqida

Deykstra algoritmi qandaydir A va B nuhtalari orasida mumkin bo'lgan yo'llar orasidan eng kichik yo'lni topish algoritmi bo'lib topiladi

Saqlaw

Biykarlaw

11-rasm

ALGORITMLER HAQQINDAG'I MAQALALAR BAZASI'

Fayl Izlew ?

Algoritmlestiriv ha'm programmalastiriv

ALGORITMLER HAQQINDAG'I MAQALALAR BAZASI'

- Graflar
 - Eng kichik yo'l
 - Deykstra algoritmi

Deykstra algoritmi haqida

Deykstra algoritmi qandaydir A va B nuhtalari orasida mumkin bo'lgan yo'llar orasidan eng kichik yo'lni topish algoritmi bo'lib topiladi

12-rasm

Endi ushbu dizimda izlashni o'rganamiz. Izlash tizimi ikki hilda: kalit so'zlar yoki maqola nomi orqali ishlaydi. Biz izlash turini o'zimiz mos maydonni belgilash orqali o'rnatamiz.

13-rasmda biz kalit so'zlar orqali yuqorida kiritilgan maqolani topdik:



Statyalardan mag'li'wmat izlew

Izleniwshi so'z:

Izlew turi

☐ Statya atamasinda ☒ Gilt so'zlerden

Deykstra algoritmi

13-rasm

Rasmdan ko'rinib turibdi, biz kalit so'zlar bo'yicha izlash rejimini o'rnatganmiz va izlanuvchi satrga faqay 'dey' deb yozib qo'yganmiz. Izlash interaktiv rejimda amalgam oshirilib turibdi. Demak, tizimimiz yaxshi ishlab turibdi.

Ushbu tizim kodlari I-qo'shimchada berilgan.

Xulosa

Biz ushbu magistrlik dissertatsiyamizda Delphi dasturlash tizimi orqali o'rgatuvchi adabiyotlar va boshqa da materiallarni ma'lumotlar bazalarini boshqarish tizimlari bilan bog'langan xolda ishlatish masalasini ko'rib chiqdik. Dastlab ishimizda Delphi muhiti orqali ma'lumotlar berib ketganmiz. So'ngra ushbu muhitda qandaydir amaliy masalalarni yechish ham qarashtirilib o'tilgan.

Delphi muhitining MBBT lar bilan ishlash mexanizmlari ham ko'rib chiqilgan. BDE, Database Desktop yordamchi tizimi va, albatta, ADO texnologiyasi haqida aytilgan. ADO tizimi bilan masalalar yechishga mizollar keltirilgan.

Ishimizning uchinchi bo'limida biz yuqorida keltirib o'tilgan a'lumotlardan foydalangan xolda Delphi muhiti va ADO texnologiyasiga asoslangan "Algoritmlashtirish asoslari" nomli amaliy maqolalar ma'lumotlar bazasidan turuvchi tizim ishlab chiqqanmiz va ushbu tizim bilan ishlash jarayonini qadamba-qadam ko'rsatib o'tganmiz.

Albatta, biz ushbu tizimni to'liq darajada yetishgan tizim deb ayta olmaymiz, sababi bu yerda biz tizim funktsionalini sal kamroq jiddiylashtirishimiz lozim bo'ladi, masalan, tizimga maqola qo'shish uchun maxsus parol yoki maxsus menyu yaratish kerak, birnechta tillar (rus, ingliz) bilan ishlashishni amalga oshirish kerak va hokazo. Biz bu ishimiz natijasida paydo bo'lgan tizim misolida Delphi muhitining MBBT lar bilan ish olib borish mexanizmlarini ko'rib chiqishni asosiy maqsad qilib qo'ganmiz. Dasturimizni esa xoxlagan payt xoxlagan odam rivojlantirishi mumkin. Dastur kodini to'lig'i bilan I-qo'shimchada keltirib o'tganmiz.

QO'SHIMCHALAR

I-Qo'shimcha

“Algoritmlashtirish asoslari” maqolalar bazasi kodi

unit Unit1; /Bosh forma

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, OleCtrls, SHDocVw, ComCtrls, StdCtrls, Menus, ExtCtrls, DB,
ADODB,comobj;

type

TForm1 = class(TForm)
 Label1: TLabel;
 Label2: TLabel;
 TreeView1: TTreeView;
 StatusBar1: TStatusBar;
 MainMenu1: TMainMenu;
 Fayl1: TMenuItem;
 Janalaw1: TMenuItem;
 N1: TMenuItem;
 Shigiw1: TMenuItem;
 Izlew1: TMenuItem;
 N2: TMenuItem;
 Statyaqosiwtartibi1: TMenuItem;
 Dasturavtorlari1: TMenuItem;
 Panel1: TPanel;
 WebBrowser1: TWebBrowser;
 DataSource1: TDataSource;

```

ADOConnection1: TADOConnection;
ADOQuery1: TADOQuery;
procedure Dasturavtorlari1Click(Sender: TObject);
procedure Shigiw1Click(Sender: TObject);
procedure Izlew1Click(Sender: TObject);
procedure Janalaw1Click(Sender: TObject);
procedure FormCreate(Sender: TObject);
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure TreeView1Click(Sender: TObject);
procedure Statyaqosiwtartibi1Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
node:TTreeNode;
end;

var
    Form1: TForm1;

implementation

uses Unit2, Unit3, Unit4;

{$R *.dfm}

procedure TForm1.Dasturavtorlari1Click(Sender: TObject);
begin
    aboutbox.show
end;

```

```
procedure TForm1.Shigiw1Click(Sender: TObject);  
begin  
  showmessage('Xosh, saw boliniz!');  
  form1.Close  
end;
```

```
procedure TForm1.Izlew1Click(Sender: TObject);  
begin  
  form3.show  
end;
```

```
procedure TForm1.Janalaw1Click(Sender: TObject);  
var  
  i:integer;  
begin  
  form4.utarau:=false;  
  form4.tarau:=false;  
  for i:=0 to treeview1.Items.Count-1 do  
    if treeview1.Items.Item[i].Level=0 then  
      form4.ComboBox1.Items.Add(treeview1.Items.Item[i].Text);  
  form4.show  
end;
```

```
procedure TForm1.FormCreate(Sender: TObject);  
  
begin  
  form1.Caption:=label2.Caption;  
  treeview1.LoadFromFile(extractfilepath(application.ExeName)+'config.sys');  
end;
```

```

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
if      messagebox(0,'Saqlanamizba?','Shigiw',1)=1      then
treeview1.SaveToFile(extractfilepath(application.ExeName)+'config.sys');
end;

```

```

procedure TForm1.TreeView1Click(Sender: TObject);
var
ss:string;
n:TTreeNode;
k:integer;
w:olevariant;
begin
n:=treeview1.Selected;
if (n.Level<>0) and (n.Parent.Level<>0) then
begin
ss:=n.Text;
adoquery1.SQL.Clear;
adoquery1.SQL.Add('SELECT * FROM [pages] WHERE [title]=:p1');
adoquery1.Parameters.ParamByName('p1').Value:=ss;
adoquery1.Active:=true;
k:=adoquery1.Fieldbyname('id').Value;
w:= CreateOleObject('Word.Application');
w.Visible := False;

w.Documents.Open(extractfilepath(application.ExeName)+'files\'+'inttostr(k)+''.rtf')
;

```

```
w.ActiveDocument.SaveAs(extractfilepath(application.ExeName)+'files\html\' +int  
tostr(k)+'.html', 8);
```

```
w.Quit(False);
```

```
w:=Unassigned;
```

```
webbrowser1.Navigate(extractfilepath(application.ExeName)+'files\html\' +inttostr  
(k)+'.html');
```

```
end;
```

```
end;
```

```
procedure TForm1.Statyaqosiwtartibi1Click(Sender: TObject);
```

```
var
```

```
f:textfile;
```

```
begin
```

```
assignfile(f,extractfilepath(application.ExeName)+'temp.htm');
```

```
rewrite(f);
```

```
writeln(f,'<h3>Maqala qosiw tartibi</h3><hr>');
```

```
writeln(f,'<p align=justify>');
```

```
writeln(f,'&nbsp;Daslep File --> Maqala qosiw<br>');
```

```
writeln(f,'&nbsp;1.&nbsp;Maqala atin korsetiw<br>');
```

```
writeln(f,'&nbsp;2.&nbsp;Maqalaga tiyisli bapti tanlaw (joq bolsa payda  
etiw)<br>');
```

```
writeln(f,'&nbsp;3.&nbsp;Maqalaga tiyisli bap bolimin tanlaw (joq bolsa payda  
etiw)<br>');
```

```
writeln(f,'&nbsp;4.&nbsp;Gilt sozlerdi kiritiw<br>');
```

```
writeln(f,'&nbsp;5.&nbsp;Maqala tekstin kiritiw (Word hujjetinen kopiyalap aliw  
ha'+#39+'m ornatiw)<br>');
```

```
writeln(f,'<br><center><img src=files/intro.gif width=550px></center></p>');
```

```
closefile(f);
```

```
webbrowser1.Navigate(extractfilepath(application.ExeName)+'temp.htm');  
end;
```

```
end.
```

```
unit Unit2; /Dastur avtorlarini chiqaruvchi forma
```

```
interface
```

```
uses Windows, SysUtils, Classes, Graphics, Forms, Controls, StdCtrls,  
Buttons, ExtCtrls;
```

```
type
```

```
  TAboutBox = class(TForm)
```

```
    Panel1: TPanel;
```

```
    ProgramIcon: TImage;
```

```
    ProductName: TLabel;
```

```
    Version: TLabel;
```

```
    Copyright: TLabel;
```

```
    Comments: TLabel;
```

```
    OKButton: TButton;
```

```
    procedure OKButtonClick(Sender: TObject);
```

```
  private
```

```
    { Private declarations }
```

```
  public
```

```
    { Public declarations }
```

```
end;
```

```
var
```

```
  AboutBox: TAboutBox;
```

implementation

{ \$R *.dfm }

procedure TAboutBox.OKButtonClick(Sender: TObject);

begin

aboutbox.Close

end;

end.

unit Unit3; /Izlash formasi

interface

uses

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

type

TForm3 = class(TForm)

Label1: TLabel;

Edit1: TEdit;

RadioGroup1: TRadioGroup;

rb1: TRadioButton;

ListBox1: TListBox;

rb2: TRadioButton;

procedure FormShow(Sender: TObject);

procedure ListBox1Click(Sender: TObject);

procedure Edit1Change(Sender: TObject);

```
private
    { Private declarations }
public
    { Public declarations }
end;
```

```
var
    Form3: TForm3;
```

```
implementation
```

```
uses Unit1;
```

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

```
procedure TForm3.FormShow(Sender: TObject);
begin
    edit1.Clear;
    edit1.SetFocus;
    listbox1.Clear
end;
```

```
procedure TForm3.ListBox1Click(Sender: TObject);
var
    k:integer;
    w:olevariant;
begin
    if listbox1.Items.Count>0 then
        begin
            with form1 do
```

```

begin
    adoquery1.SQL.Clear;
    adoquery1.SQL.Add('SELECT * FROM [pages] WHERE [title]=:p1');

    adoquery1.Parameters.ParamByName('p1').Value:=listbox1.Items.Strings[listbox1.
ItemIndex];

    adoquery1.Active:=true;
    k:=adoquery1.Fieldbyname('id').Value;
    w:= CreateOleObject('Word.Application');
    w.Visible := False;

    w.Documents.Open(extractfilepath(application.ExeName)+'files\'+inttostr(k)+'.rtf')
;

    w.ActiveDocument.SaveAs(extractfilepath(application.ExeName)+'files\html\'+int
tostr(k)+'.html', 8);
    w.Quit(False);
    w:=Unassigned;

    webbrowser1.Navigate(extractfilepath(application.ExeName)+'files\html\'+inttostr
(k)+'.html');
    end;
    form3.Close;
end;
end;

procedure TForm3.Edit1Change(Sender: TObject);
var
    ii:integer;

```

```

begin
    listbox1.Clear;
if length(edit1.Text)>1 then
    begin
        if rb1.Checked then
            begin
                with form1 do
                    begin
                        adoquery1.SQL.Clear;
                        adoquery1.SQL.Add('SELECT * FROM [pages] WHERE [title] LIKE
'+#39+'%'+edit1.Text+'%'+#39);
                        //adoquery1.Parameters.ParamByName('p1').Value:=edit1.Text;
                        adoquery1.Active:=true;
                        adoquery1.First;
                        for ii:=0 to adoquery1.RecordCount-1 do
                            begin
                                listbox1.Items.Add(adoquery1.fieldbyname('title').AsString);
                                adoquery1.Next;
                            end;
                        end;
                    end
                end
            else
                begin
                    with form1 do
                        begin
                            adoquery1.SQL.Clear;
                            adoquery1.SQL.Add('SELECT * FROM [pages] WHERE [keyword] LIKE
'+#39+'%'+edit1.Text+'%'+#39);
                            //adoquery1.Parameters.ParamByName('p1').Value:=edit1.Text;

```

```
adoquery1.Active:=true;
adoquery1.First;
for ii:=0 to adoquery1.RecordCount-1 do
begin
    listbox1.Items.Add(adoquery1.fieldbyname('title').AsString);
    adoquery1.Next;
end;
end;
end;
end;
end;
end.
```

Foydalanilgan adabiyotlar dizimi

1. Lyubavin S.A. Programmirovaniye na Delphi Win32.
2. Lyubavin S.A. Turbo Delphi dlya novichkov i ne tol'ko
3. Shupruta V.V. Delphi 2005 Uchimsya programmirovat
4. Marco Cantu Delphi 2009
5. Rey Konopka Sozdanie originalnyx komponent v srede Delphi
6. Mironchenko A.C. Imperativnoe i ob'ektno-orientirovannoe programmirovaniye na Turbo Pascal i Delphi
7. Djulian B. Fundamentalnye algoritmy i struktury dannyx v Delphi
8. Faronov V.V. Delphi programmirovaniye na yazyke vysokogo urovnya
9. Gofman V., Xomonenko A., Delphi bystry start