

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

BUXORO DAVLAT UNIVERSITETI

Fizika –matematika fakulteti

“Informatika va axborot texnologiyalari” kafedrası

“KT: Informatika va axborot texnologiyalari” ta’lim yo’nalishi

“Zamonaviy dasturlash tillari” fanidan

KURS ISHI

Mavzu: Delphi dasturlash tilida tarmoqni scannerlash va ularni boshqarish.

Bajardi: 1-1PMIK-13 guruh talabasi Rahmonqulov Ma’mur Qobul o’g’li.

Kurs ishi himoya qilingan sana

“ ____ ” _____ 2015y.

Ball _____

Komissiya a’zolari

Mundarija:

Kirish.....	3
I. Dasturlash tillari.....	5
1.1. Dasturlash tillari haqida umumiy ma'lumot.....	5
1.2. Delphi dasturlash tili va uning imkoniyatlari.....	10
1.3. Delphini o'rnatish. Delphi oynalari haqida.....	12
II. Delphi dasturlash tilida tarmoqni skanerlovchi va	
boshqaruvchi dastur yaratish.....	20
2.1. Tarmoq haqida umumiy tushunchalar.....	20
2.2. Delphi dasturlash tilining tarmoq bilan ishlovchi	
komponentalari.....	23
2.3. Tarmoqni skanerlovchi va boshqaruvchi dastur yaratish.....	26
2.4. Yaratilgan dasturdan foydalanish uchun qo'llanma.....	33
Xulosa.....	36
Foydalanilgan adabiyotlar ro'yxati.....	37

KIRISH

O'zbekiston Respublikasi mustaqillikka erishgan yillaridan e'tiboran mamlakatimiz hayotida haqiqatan ham ulkan tarixiy ahamiyatga ega bo'lgan eng muhim ijtimoiy dasturlar – kadrlar tayyorlash milliy dasturi va maktab ta'limini rivojlantirish davlat umummilliy dasturining ijrosi amalda nihoyasiga yetkazildi.

O'zbekistonda o'qitish texnologiyalari zamonaviylashtirish, jadallashtirish rivojlangan iqtisodiyotli mamlakatlarga qaraganda yanada dolzarb ahamiyatga ega. Chunki hozirgi kunda o'zbek milliy ta'lim tizimining salohiyatli tizimli rivojlanishi yanada yuqori pog'onasiga ko'tarildi.

Prezidentimiz I.A.Karimov 2001-yil Oliy Majlisining 5-sessiyasida so'zlagan nutqida axborot texnologiyalarni va kompyuterlarni jamiyat hayotiga, kishilarning turmush tarziga, o'rta maktab, o'rta maxsus va oliy ta'lim muassasalariga jadallik bilan kirish g'oyasi ilgari surildi. Xususan, biz kabi yoshlarni zamon talablariga javob beruvchi mutaxassislar qilib tayorlashda alohida e'tibor qaratilmoqda.

Respublikamizda olib borilayotgan islohotlarning tarkibida yuqori malakali mutaxassislarning roli benihoyat kattadir. Prezidentimiz ta'kidlaganlaridek, "Ertangi kun yangicha fikrlay oladigan zamonaviy bilimga ega bo'lgan yuksak malakali mutaxassislarni talab etadi".

Respublikamiz mutaxassislar tayyorlashni davr talablari darajasida tashkil etishda yetarlicha ishlar olib borilayotganligiga qaramasdan, hozirgacha bir qator muammolar o'z yechimini topgani yo'q. Xususan, jahon andozalariga mos va raqobatbardosh, bilimli, malakali, kuchli mutaxassislarni tayyorlashga imkon beruvchi pedagogik shart – sharoitlarni yaratish, mazkur jarayonni metodik jihatdan to'g'ri tashkil etish, mutaxassislik fanlaridan kasbiy ta'lim va malakalarni shakllantiruvchi ilmiy – uslubiy qo'llanmalar ishlab chiqish kabi masalalarni shular qatoriga kiritish mumkin.

Hozirgi zamon mutaxisslardan faoliyat doirasi qanday bo'lishidan qat'iy nazar, informatika bo'yicha keng ko'lamdagi bilimlarga zamonaviy hisoblash

texnikasi va information aloqa va kommunikatsiya tizimlari, orgtexnika vositalari va ulardan foydalanish borasida yetarli malakaga ega bo'lishi, hamda yangi information texnika va texnologiya asoslarini uning ertangi kuni, rivoji to'g'risidagi bilimlarni o'zida mujassamlashtirgan bo'lishi shart. Zamonaviy hisoblash texnikalari va information texnologiyalarining kun sayin rivojlanib, jamiyatning esa, tobora informatsiyalashib borishi sababli, uzluksiz ta'lim tizimining o'rta va yuqori bosqichlariga informatika, ishlab chiqarish va boshqarish jarayonining kompyuterlashtirish bo'yicha bir qator o'quv fanlari kiritilgan.

XX asr o'rtalariga kelib, tezkor mashina mexanizmlaridan foydalanila boshlandi, murakkab texnika va texnologiyalardan ko'p funksiyalilari o'ylab topildi. Ko'pgina masalalarni hal qilish jarayonida axborot hajmi behisob bir majmuaga aylandi, hamda bu axborotlarni yig'ish va uzatish vositalarini yaratish, ularni vaqtida qayta ishlab, boshqarish uchun zarur bo'lgan choralarni belgilab chiqish kerak bo'ladi.

Axborot kommunikatsion texnologiyalar(AKT) talabalarning kelajakdagi faoliyatiga tayyorlashga ko'mak beradi. Hozirgi zamonda mehnat faoliyati o'quvchilar qoniqish bilan ishlayotgan kompyuterlar, texnologiyalar, dasturlar va qurilmalar yordamida boshqariladi. AKT o'rganish va o'qitishning yangi imkoniyatlarini ochib beradi. Inson multimediya ko'rgazma vositalari yordamida, o'quv ma'lumotlarini qabul qilganda sezgi a'zolari ishtirok etadi va u oson tushunadi. Multimediya ko'rgazma vositalari kompyuter animatsiyalari yordamida ham yaratiladi. Animatsiya obyektlar harakatini, immitatsiyasini namoyon etishga keng imkoniyat ochib beradi. O'quv ma'lumotlarini shartli grafik tasvirlar (sxemalar, bloksxemalar, diagrammalar, tarayektoriyalar) yordamida tushunishda animatsiyalardan foydalanish unumli hisoblaniladi.

I. Dasturlash tillari.

1.1. Dasturlash tillari haqida umumiy ma'lumot.

Dasturlash tillari sun'iy tillar hisoblanadi, ularda sintaksis va semantik qoidalar qat'iy aniqlangan bo'ladi. Shu sababli Dasturlash tillari tabiiy tillardan farqli ravishda jummalarni ko'pmazmunli va erkin talqin etishga yo'l qo'ymaydi. Buning asosiy sababi tildagi har bir ko'rsatma mashina tilidagi aniq bir buyruqqa o'tadi.

Sintaksis-bu qoidalar to'plami bo'lib, Dasturlash tilida ruxsat etilgan belgilarning ketma-ketligi va asosiy ichki tuzilishni aniqlaydi.

Semantika-til birliklariga (so'z, so'z birikmalari, jumllariga) beriladigan qiymatlardir.

EHMlar endi yuzaga kelgan paytda dastur tuzishda, faqat mashina tillarida, ya'ni sonlar yordamida EHM bajarishi kerak bo'lgan amallarning kodlarida kiritilgan. Bu holda mashina uchun tushinarli sanoq, sistemasi sifatida 2 lik, 6 lik, 8 lik sanoq sistemalari bo'lgan. Dastur mazkur sanoq sistemasidagi sonlar vositasida kiritilgan.

Yuqori bosqichli dasturlashda, mashina tillariga qaraganda mashinaga moslashgan (yo'naltirilgan) belgili kodlardagi tillar hisoblanadi. Belgilar kodlashtirilgan tillarning asosiy tamoyillari shundaki, unda mashina kodlari ularga mos belgilar bilan belgilanadi, hamda xotirani avtomatik taqsimlash va xatolarni tashhis qilish kiritilgan. Bunday mashina moslashgan til - ASSEMBLER tili nomini oldi.

EHM faqat mashina tilini-buyruqlar, operand va sonlarning ikkilik sanok, sistemasidagi ko'rinishini «tushinadi». Shu sababli, Assembler tilida yozilgan dasturlar uchun ularni mashina tiliga o'tkazuvchi «tarjimon» kerak bo'ladi.

Assembler tilidagi dasturni mashina tiliga *o'tkazuvchi* maxsus dastur translator deyiladi, ayrim hollarda uni assembler deb ham atashadi.

Mashinaga moslashgan tillarning asosiy kamchiligi bir turdagi mashina uchun tuzilgan dastur boshqa mashinalarda bajarilmaydi, ya'ni bunday tillar mashinaning apparat tuzilishiga bog'liq qilib yaratiladi. Masalan, Praves mashinasi uchun tuzilgan dastur IBM yoki YAMAHA turidagi mashinalar uchun o'tmaydi va aksincha.

Dasturlash tillarining keyingi bosqichida protsedurali tillar joylashadi. Bu tillarning asosiy xususiyati shundaki, ularning sintaksis va semantikasi konkret EHM (protessor) buyruqlari tarkibiga bog'liq emas. Tuzilgan dasturni konkret EHM ga bog'lashni translator amalga oshiradi.

Dasturning boshlang'ich matni operativ xotiraga kiritilgandan keyin u translatsiya qilinadi. Natijada, aynan shu mashina «tushinadigan» buyruqlar ketma-ketligi hosil bo'ladi va ularni bajarish asosida EHM masalani yechadi.

Dasturni tarjimasi va uni bajarish jarayoni ikki usulda amalga oshirilishi mumkin:

Birinchi usul-kompilator deb nomlanuvchi usulda dasturni EHMda bajarilishi-tarjima jaryoni to'liq, tugagandan keyin amalga oshiriladi. Bunda tarjima dastursini operativ xotirada saqlab turishga hojat yuq, shu sababli xotirani tejashga erishiladi.

Ikkinchi usul – interpretatsiya - dasturdagi ayrim operatorlar ular tarjima qilingan zahoti bajariladi, shundan keyin navbatdagi operator tarjima qilinadi, bajariladi va hokazo. Bu rejimda xotirada interpretator dastursi ham bo'lishi kerak, natijada qo'shimcha xotira ishlatiladi va dasturning ishlashi kompilyatsiya qilinganga nisbatan sekin ishlaydi. Interpretatorning afzallik tomonlari dasturlarni sozlash paytida ko'rinadi, ya'ni yo'l qo'yilgan xatolar tuzatilishi mumkin. Kompilatorida bunday imkoniyat yo'q.

Protsedurali tillarning umumiy xususiyatlarini ko'raylik. Bu tillarda dasturdagi yozuvlar umumiy qabul qilingan matematik yozuvlarga juda yaqin va tushunish uchun yengildir. Masalan, 7 va 5 sonlarning yig'indisini hisoblash amalini dasturlash tillarida yozilishini ko'raylik:

- 1) mashina tilida

Buyruq	Operand adresi
0011 1111	1101 0001
0000 0101	1101 0010
0000 0110	1101 00111
0000 0111	1101 0100
1000 0000	1101 0101

2) mashinaga moslashgan tilda

MOV AX, 7

ADD AX, 5

MOV A, AX

3) protsedurali tillarda $A=7+5$.

Ko‘rinib turibdiki, protsedurali tillardagi yozuv tabiiy yozuvga yaqin.

Birinchilar qatorida yuzaga kelgan protsedurali tillar ichida FORTRAN (FORmula TRANslation - formulalarni boshqa ko‘rinishga aylantirish). O‘zining sodda tuzilishi, samarali translatsiyalanishi bu tilni nafaqat shu kungacha saqlanib qolishiga sabab bo‘ldi, balki uning ilmiy tadqiqot, injener-texnik masalalarini yechishda eng samarali til sifatida rivojlanib, keng tarqalishiga sabab bo‘ldi.

Fortran tili Beysik tilining asosiga qo‘yildi. Beysik so‘zi inglizcha «VASIC-Veginners All-purpose Symbolic Instruction Sode» ma‘noni anglatadi. Bu jumla quyidagicha tarjima kilinadi: boshlovchilar uchun ko‘pmaqsadli belgili buyruqlar tili.

Beysik tili mutaxassis bo‘lmagan odamlarni dasturlash texnologiyalariga jalb qilish imkoniyatini berdi. Beysik tilida dastur operatorlarining kamligi, tuzilishining soddaligi uni dasturlashga o‘rgatish tili sifatida keng tarqalishiga sabab bo‘ldi. Bu til asosan maktablarda o‘rgatiladi.

Algol-60 tili Fortranga nisbatan takomillashgan til, unda tuzilgan dasturlar moslashuvchanlik va yuqori ishonchlilik xususiyatlari bilan xarakterlanadi.

Prolog va Lisp tillari sun'iy tafakkur sohasidagi masalalarni yechishga moslashgan. Bu tillar belgili ma'lumotlarni qayta-ishlash orqali matematika va mantiqiy masalalarni yechishga eng qulay tillardir.

1971 yilda paydo bo'lgan Paskal tili dasturlash texnologiyalariga strukturali dasturlash g'oyasini olib kirdi. Natijada bitta masala bir nechta bo'laklarga bo'lingan holda alohida yechilishi va oxirida bitta dasturga jamlanish imkoniyati yuzaga keldi.

Si tili zamonaviy kompyuterlar xususiyatlarini o'zida aks ettirgan va samarali dasturlar yaratishda Assemblerga murojaat qilmaydigan til sifatida namoyon bo'ldi. Bu til UNIX operatsion sistemasida yozilgan.

Ayni paytda dasturlash texnologiyasining istiqbolli yo'nalishlaridan biri obyektga yo'naltirilgan dasturlash tillaridir (объектно-ориентированное программирование).

Obyektlar ko'p ishlatiladigan dastur modullaridir. O'z navbatida obyektlar ikkita qismdan tashkil topadi: metodlar va o'zgaruvchilar. Metodlar protsedura va funksiyalar to'plami bo'lib, obyekt ishlash algoritmini aniqlaydi. O'zgaruvchilar turli xil qiymatlarni qabul qiladi.

Obyektga moslashgan dasturlashning asosiy mohiyati shundaki, sodda obyektlardan vorislik orqali murakkab obyektlar shajarasini yaratish mumkin. Bunda biror masalani yechish deganda masala algoritmiga mos obyektlar shajarasini qurish tushiniladi.

Obyektga yo'naltirilgan tillarning yaqqol namunasi - bu Delphi vizual dasturlash muhitidir. Bu muhit Turbo Paskal tili asosida yaratilgan bo'lib, unda dastur yaratish muhitidagi mavjud komponentalarni loyihalash asosida bo'ladi. Shuning uchun ham bosh dastur Rgoject deyiladi. Delphining yana bir muhim xususiyati vizual dasturlashdir, ya'ni dastur yaratilayotgan paytda tuzuvchi

hosil bo'ladigan dastur shaklini ko'rib turadi va grafik interfeys yordamida komponentalar shaklini o'zi ma'qul topgan joyiga qo'yishi mumkin.

Visual Basic tili ham obyektga moslashgan til hisobalanadi va Windowsning ofis dasturlarida makroslar yozish uchun til sifatida ishlatiladi.

Internet uchun dasturlar yaratish uchun NTML (Nureg Text Markur Language) 1989 yilda yaratilgan bo'lib, WWW sahifalarini yaratish uchun asosiy til bo'lib qolmoqda. Bu til yordamida sahifa poligrafik usulda yaratishilishi mumkin, unda matn, audio, video, animatsiya va boshqa ma'lumotlar joylashishi mumkin.

Java tili kompyuter tarmoqlarida amal qiladigan dasturlarni yaratishga mo'ljallangan, ya'ni Java tilida yozilgan dasturlar global tarmoqlarda dinamik reklamalarni (animatsiya, teletayp lentolari) global tarmoqda joylashtirish uchun ishlatiladi, Java tilining afzalligi shundaki, u turli operatsion sistemalarda ham birdek amal qilaveradi. Java dastursi bir paytda Windows, UNIX va Macintosh operatsion sistemalarida ishlashi mumkin.

Quyidagi jadvalda dasturlash tillari haqida ma'lumotlar keltirilgan.

Til	Yaratilgan yili	Mualliflar	Tashkilot, firma
Ada	1979-80	Jean Ichbian	Cii-Honeywell (Fransiya)
Algol	1960		International Commitee
ARL	1961-1962	Kenneth Iverson, Adin Falkoff	IBM
DELPHI	1995		Borland
BASIC	1964-1965	JohnKemeny, Thomas Kurtz	Dartmouth Colleje
C	1972-1973	Dennis Ritchie	Bell Laboratories
C++	1980	Bjarne Strostrup	Bell Laboratories
COBOL	1959-1961	Grace Murray Hopper	

Fort	1971	Charles H.Moore	
FORTTRAN	1950-1958	John Backus	IBM
HTML	1989	Tim Berners-Li	CERN, Jeneva
LISP	1956-1960	John MCCarthy	
LOGO	1968-70	Seymour Papert	MassachusetS Institute of Techn.
Pascal	1967-1971	Niklaus Wirth	Federal Institute of Technology (Shveytsariya)
PL1	1964-1966		
PROLOG	1978	Alan Kalmeroe	
SIMULA	1967	Ole-Yoxan Dal, Kristen Nigaard	Norvegiya XM
Java	1995	Djeyms Gosling	Sun Microsystems

1.2. Delphi dasturlash tili va uning imkoniyatlari

Delphi - dasturiy vositasi: Bu windows uchun mo'ljallangan dasturlash muhitida 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.

Borland Delphi ning paydo bo'lishi dasturlashni rivojlantirish tarixida yorqin ko'rinish bo'ldi. Delphi ning vujudga kelishiga quyidagilar sabab bo'ldi;

- Windows uchun dasturlash va komponentalar texnologiyasi.
- Masalalarni yechish uchun ob'ektga yo'naltirilgan usul.
- Komponentalar texnologiyasiga asoslangan ilovalarni tez yaratishning vizual muhitlari.

- Interpretatsiyadan emas, kompilyatsiyadan foydalanish. Bu shundan iboratki, interpretator bilan ishlashga qaraganda kompilyator bilan ishlash tezligi o'n martalab ustunlikka ega bo'ladi.

- Universal usullar yordamida ma'lumotlar bazasi bilan ishlash imkoniyatlarining mavjudligi.

Borland Delphi yuqorida bayon etilgan fikrlarni joriy etish maqsadida yaratilgan. Ammo, uning eng asosiy elementi Paskal tili bo'lib hisoblanadi. Hozirgi kunda Delphi tilining bir necha xil variantlari mavjud (Delphi 2, Delphi 3... , Delphi 7 va h.k.).

Delphi dasturlash tili uchun Paskal dasturlash tili asos qilib olingan. Bu ikki tilning qanday o'xshashlik va farqli tomonlari bor?

Bilamizki, 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 Delphi da 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 Delphi da, agar unga o'zimiz nom bermasak, Project1.1dpr deb nomlaydi. Bunda .dpr fayl kengaytmasi. Avtomatik tarzda tashkil qilingan modulga biz nom bermasak, u holda Delphi uni Unit1.1pas deb nomlaydi. O'zimiz esa *.pas kengaytmasini saqlagan holda hohlagan nomimizni berishimiz mumkin.

Delphi – dasturlashtirishda, dasturni tez ishlatishda foydalaniladigan dasturlash tili. Delphi tili Object Pascal tiliga asoslangan dasturlash tili hisoblanadi. Hozirgi kunda Delphi ning yangi loyihasi Borland Delphi 7 dasturchilarga ma'qul tushdi. Borland Delphi bilan katta bazali dasturlarni yaratish mumkin. Bu paketga turli xil ma'lumotlar, XML – hujjatlari, axborotlashtirish sistemalari va boshqalar kiritilgan. Delphi 7 ning asosiy xususiyati NET texnologiyasini qo'llab quvvatlaydi.

Borland Delphi 7 bilan Windows 98 dan Windows XP gacha ishlash mumkin. Xozirgi zamon talabiga asosan bu dasturlash tili uchun Pentium va Celeron protsessorlari bo'lishi kerak va ularning chastotasi 400 MGts, operativ xotirasi 128 Mbayt, o'rnatish uchun diskda 475 Mbayt bo'sh joy bo'lishi kerak

1.3. Delphini o'rnatish. Delphi oynalari haqida.

Delphi 7 kompyuterga CD ROM orqali o'rnatiladi. Bunda hamma kerakli fayllar va dasturga nom qo'yish bor. Dasturni nomlovchi qurilma disk diskovodga quyilgandan darrov ishga tushadi.

Dasturni nomlovchi qurilmalar ishga tushgach darrov ekranga Delphi 7 oynasi chiqadi. Bular CD ROM, DVD ROM, orqali kompiyuterga o'rnatiladi.



1.3.1-chizma. Delphi 7 dasturini o'rnatishni boshlanishi.

Delphi 7 ni tezroq o'rnatish uchun **Delphi 7** tugmasini bosamiz. **Delphi 7** ni o'rnatish oddiy. (**Serial Number**) nomeri va (**Authorization Key**) kalitini yuborgandan keyin ekranda ochishga ruxsat beradi, keyin esa **Setup Type** oynasi paydo bo'ladi. Bu oynadan kerakli variantni tanlash mumkin.

Typical (oddiy), **Compact** (kompaktli) va **Custom** (tanlab olish mumkin).

Ommabop varianti **CD ROM** orqali qattiq diskka **Delphi** ning hamma komponentlarini nusxalaydi. Bu variant qattiq diskda ko'p joyni talab qiladi.

Tanlanma variant dasturchiga faqat ayrim zarur **Delphi** komponentlarini tanlash imkoniyatiga ega. Bundan malakali dasturchilar foydalanadi. Bu variant agar diskda juda kam joy bo'lganda foydalaniladi.



1.3.2-chizma. Dialog oynasi kerakli variantni o'rnatadi.

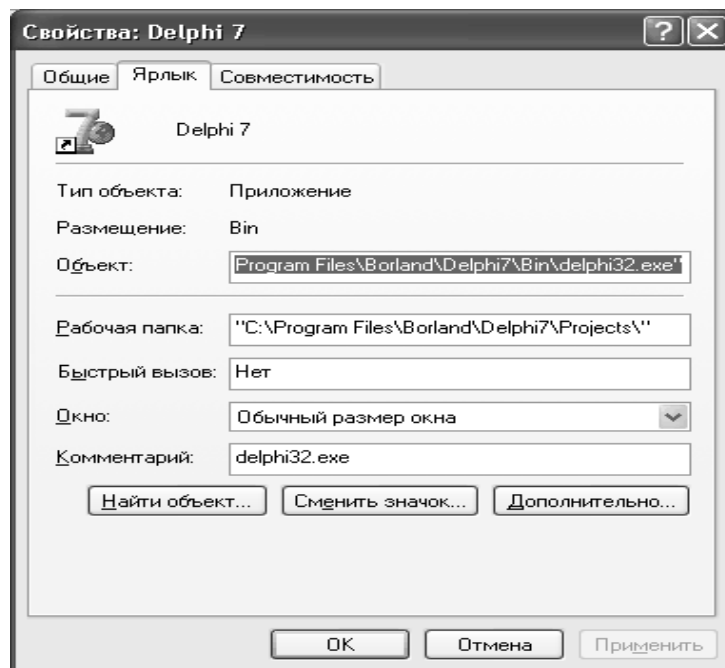
Kerakli variantni tanlab olgach **Next** tugmasini bosamiz va davom ettiramiz.

Ish jarayoni tugagandan keyin **Finish** tugmasini bossak oyna yopiladi.



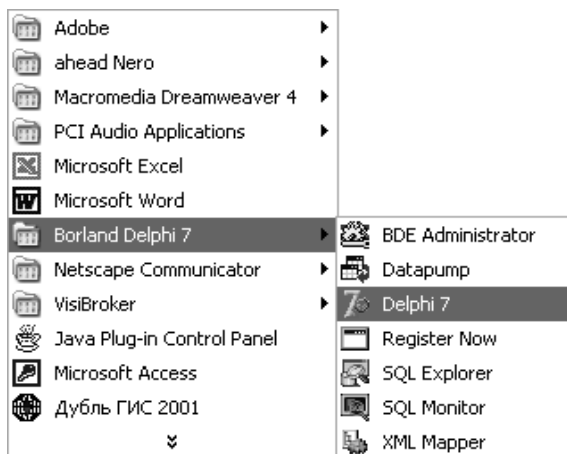
1.3.3-chizma. O'rnatish jarayoni tugadi.

Endi Delphi ni ochib, ishga tushirish mumkin. Ammo buni bajarishdan oldin proekt yoki ish kataloglarini ochish kerak. Bu uchun sichqoncha bilan Pusk – Программы- Borland Delphi 7 ga sichqonchani o'ng tugmasini bosib Svoystva ga kirib, Delphi 7 dagi Rabochaya papka maydonida papka yaratamiz. Delphi ni proektlari uchun.



1.3.4-chizma. Proekt papkasini aniqlash.

Ishning boshlanishi. **Delphi** har doimgidek ochiladi, ya'ni quyidagicha.

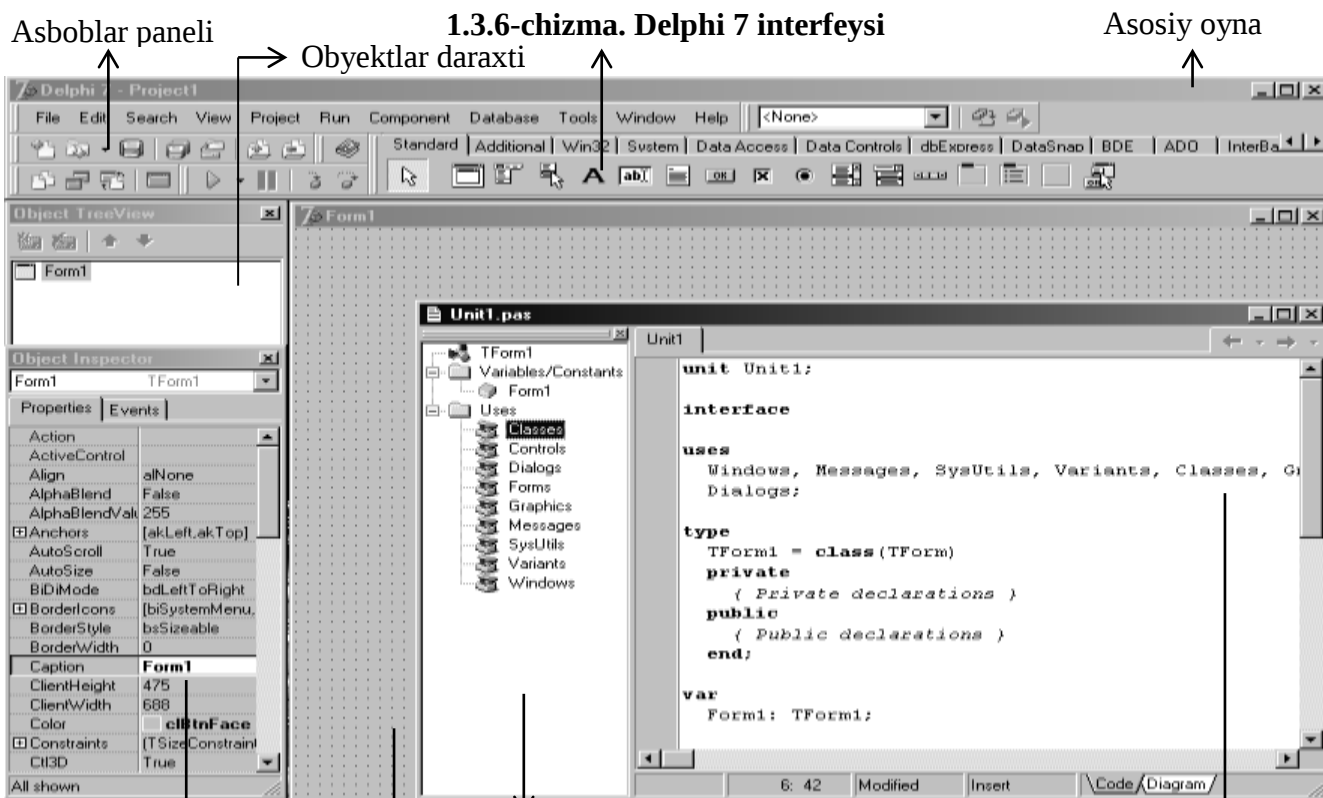


1.3.5-chizma. Delphi ni ishga tushurish.

Bu paytda bitta oyna o'rniga 5 ta oyna chiqadi.

Bu yerda Delphi ning asosiy tashkil etuvchilari keltirilgan:

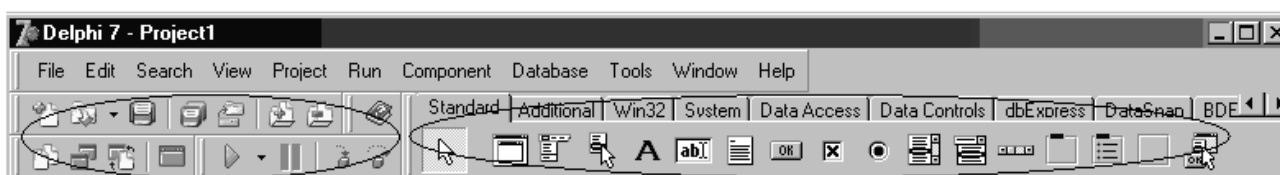
1. Delphi ning asosiy oynasi.
2. Formalar konstruktori (Form Designer)
3. Ob'ektlar inspektori (Object Inspector)
4. Ob'ektlar daraxti (Object TreeView)
5. Dastur matni muharriri (Code Editor)
6. Dastur matniga yo'l ko'rsatuvchi oynasi (Code Explorer)



Asosiy oynada buyuqlar ro'yxati chiqadi. Boshlang'ich forma oynasi asosiy oynada tayyorgarlik jarayonini ko'rsatadi. Dasturni ta'minlash ikki turga bo'linadi:

1.1 Sistemali 2. Ilovali.

Sistemali programmani ta'minlash bu operatsion sistemaning hamma ko'rinishlari, qolgan hamma ma'lumotlar ilovali. Asosiy oyna haqida



Asboblari paneli

Komponentlar paltirasi

1.3.7-chizma. Komponentlar.



Asboblari paneli	
	Yangi loyiha yaratish yoki ochilgan loyihaga yangi forma qo'yish
	Mavjud bo'lgan loyiha, forma yoki dastur matni faylini ochish. Oldingi ochilgan fayllar ro'yxatidan faylni ochish.

	Joriy faylni xotiraga saqlash. (Ctrl+S)
	Loyihaning barcha fayllari bilan birga xotiraga saqlash. (Shift+Ctrl+S)
	Mavjud bo'lgan loyiha faylini ochish. (Ctrl+F11)
	Mavjud bo'lgan forma va dastur matni faylini loyihaga qo'shish. (Shift+F11)
	Loyihadagi forma va dastur matni faylini loyihadan olib tashlash.
Custom (Help contents) asboblar paneli	
	Delphi da yordam oynasini chaqirish
View asboblar paneli	
	Delphi da modullaridan birini tanlash orqali dastur matni oynasini ochish. (Ctrl+F12)
	Delphi da formallardan birini tanlash orqali forma konstruktori oynasini ochish. (Shift+F12)
	Delphi da forma konstruktori oynasidan dastur matni oynasiga o'tish va teskarisi. (F12)
	Loyihada yangi forma qo'yish.
Debug asboblar paneli	
	Loyihani ishga tushirish. (F9)
	Ishga tushirilgan loyihani (pauza qilishi) to'xtatish.
	Kursor turgan joygacha (satrgacha) bajarish. (F7)
	Qadamba qadam bajarish. (F8)

Komponentlar palitrasi.

Komponentlar palitrasi asboblar panelining ikki qavatli balandligi ko'rinishida. Yuqori qismida bir nechta komponentlar palitrasi nomi berilgan o'tish tugmalaridan iborat, unda VCL va ActiveX barcha komponentlari mavjud. Tugmalarni joylashish tartibini sichqonchani o'ng tugmasini yoki **Component\Configure Palette** menyu buyrug'i orqali o'zgartirishingiz mumkin.

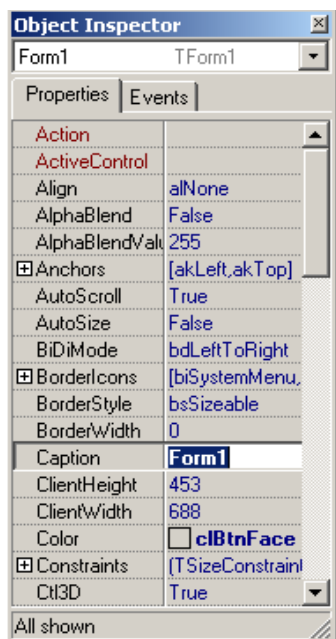
Formalar konstruktori (Form Designer) oynasi.

Forma konstruktorida yangi forma, bo'sh oyna ko'rinishida yuklanadi va siz xohlagan Windows muhitidagi oynasi ko'rinishida tashkil etish mumkin. Foydalanuvchi nuqtai nazarida forma konstruktori, loyihani interfeysi ko'rinishini sozlash uchun muhim rol o'ynaydi. Komponentlar palitrasidan tanlangan komponentingizni formada keltirib qo'yish orqali formangiz ko'rinishini

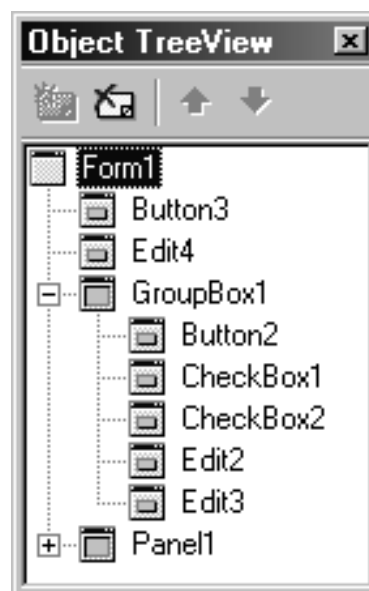
yaratishingiz mumkin. Qo'yilgan komponentlar o'lchamini, joylashish o'rnini sichqoncha orqali o'zgartirishingiz mumkin. Bundan tashqari komponentalarni tashqi ko'rinishi va ishlashini Object Inspector va CodeEditor oynasidan ham boshqarish mumkin.

Ob'ektlar inspektori (Object Inspector) oynasi.

Formadagi komponentlarning xususiyat-larini, ob'ektlar inspektori (Object Inspector) orqali o'zgartirish mumkin yoki forma yoki uning komponentalarida ta'sir etadigan holatni aniqlash mumkin. Xususiyat (properties) – Ob'ektning ekranga ko'rinishi haqida ma'lumotlarga aytiladi (balandligi, rangi, joylashish koordinatasi va h.k.) Holat (events) –Loyihangizda sodir bo'layotgan jarayon (hodisa) dagi bajarilishi kerak bo'lgan qismdastur. Misol sifatida komponenta ustida sichqonchani kelishi, yurishi, ketishi holati (jarayoni). Ob'ektlar inspektorda (Object Inspector)tanlangan ob'ektingizni holati yoki xususiyati bilan ishlashingiz uchun sichqonchani (Properties yoki Events) o'tish tugmasidan tanlashingiz kifoya.(1.3.8-chizma)



1.3.8-chizma



1.3.9-chizma

Ob'ektlar daraxti (Object TreeView) oynasi.

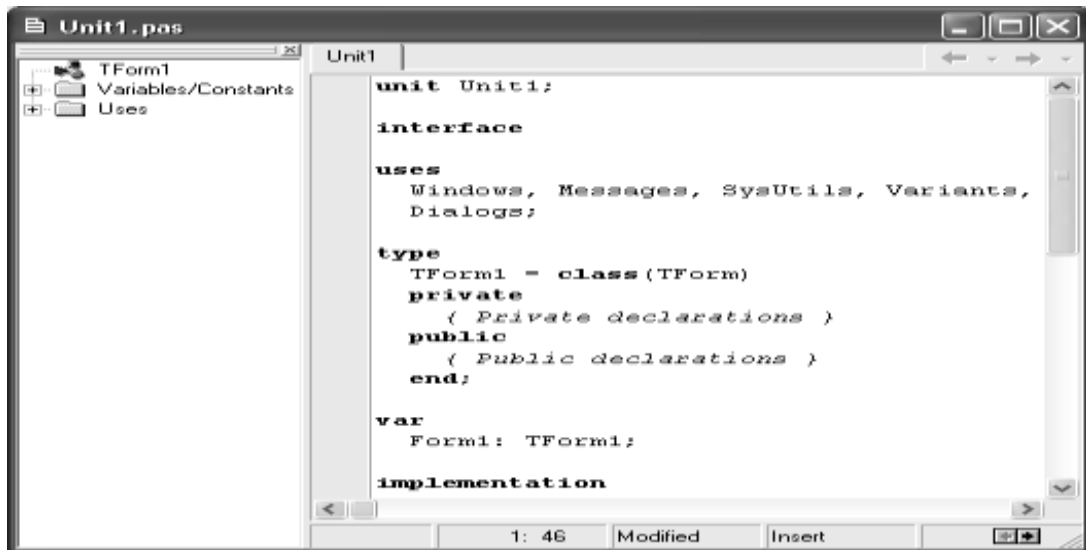
Forma konstruktorida joylashgan ob'ektlar, Ob'ektlar daraxti (Object TreeView) oynasida ob'ektlar ichma ich joylashishi daraxsimon ko'rinishida chiqadi (masalan Forma, Panel va h.k.) va siz xohlagan vaqtingizda shu oynadan qaysi ob'ekt ikkinchi bir ob'ekt ichida joylashishini boshqarishingiz mumkin. Bundan tashqari bu oynadan xohlagan ob'ektinigizni tanlashingiz mumkin ob'ektlar inspektori (Object Inspector) oynasida uning xususiyati yoki holatlari bilan ishlamoqchi bo'lsangiz.(1.3.9-chizma)

Dastur matni muharriri (Code Editor) oynasi.

Dastur matni muharriri (Code Editor) oynasida yaratayotgan dasturingizni ishlash jarayonida bajarilishi kerak bo'lgan dastur matnlari yoziladi. Bu oynada har bir formaning modullari ochilgan bo'lishi mumkin yoki (proekt) loyiha fayli. Loyihangizda har bir yaratilgan formangiz uchun modul (dastur matni) fayli yaratiladi va bu fayllar dastur matni muharriri (Code Editor) oynasida ochiladi. Code Editor oynasi bir nechta yangi oyna ochish uchun asosiy menyudan **View\New Edit Window** buyruq orqali bajariladi.

Unitl.pas –paskal tilidagi kodi oynasi. Unitl.pas ni ishlatish uchun Form1 ni chetga surib dasturni kiritamiz. (1.3.10-chizma)

Dastur matniga yo'l ko'rsatuvchi oynasi (Code Explorer).



1.3.10-chizma

Code Editor oynasida, modul (dastur matni) faylidagi dasturingizning tuzilishini Code Explorer oynasida daraxsimon ko'rinishida chiqadi (masalan o'zgaruvchilar, konstantalar, ob'ektlar holatlarini qism dasturlari, chaqirilgan modullar va h.k.). Bu oynada dasturning o'zgaruvchining e'lon qilgan joyiga, ob'ektlar holatlarini qism dasturlari joyiga va h.k. larga tez o'tish uchun qulay.

II. Delphi dasturlash tilida tarmoqni scannerlovchi va boshqaruvchi dastur yaratish.

2.1. Tarmoq haqida umumiy tushunchalar.

Jamiyatning xozirgi bosqichida axborot texnologiyalarining rivojlanishini kompyuter tarmoklarisiz tasavvur etib bulmaydi.

Kompyuter (xisoblash) tarmoKi — bu, aloka kanallari orkali yagona tizimga bog'langan kompyuter va terminallar majmuasidir.

Tarmokda axborotni ishlab chikaruvchi va undan foydalanuvchi ob'ektlar tarmok ob'ektlari deyiladi. Tarmok ob'ektlari aloxida kompyuter, kompyuterlar kompleksi, ishlab chikarish robotlari va boshkalar bulishi mumkin. Axborotlarni territorial joylashuviga kura kompyuter tarmoklarini uchta asosiy sinfga bulish mumkin: global tarmoklar, regional (mintakaviy) tarmoklar, lokal (maxalliy) tarmoklar.

Global kompyuter tarmoklari turli mamlakatlarda, turli kit'alarda joylashgan abonentlarni birlashtiradi.

Abonentlar orasida aloka bunday tarmoklarda telefon aloka liniyalarda, radioaloka va sputnik aloka tizimlari asosida amalga oshiriladi.

Mintakaviy kompyuter tarmoklari bir-biridan ancha uzakda joylashgan biror mintakaga tegishli abonentlarni birlashtiradi. Masalan, biror shaxar ichidagi yoki iktisodiy regionda yoki aloxida bir mamlakatda joylashgan abonentlarni birlashtiruvchi tarmok.

Lokal (maxalliy) tarmok kichik bir xududda joylashgan abonentlarni birlashtiradi. Bunday tarmok odatda anik bir joyga boKlangan buladi. Masalan, biror korxona yoki tashkilotga. Maxalliy tarmokning uzunligini 2—3 km bilan cheklash mumkin.

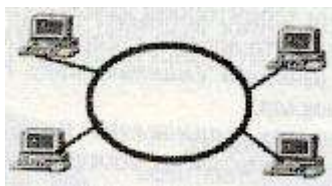
Global, mintakaviy va maxalliy tarmoklar birlashmasi kup tarmokli ierarxiyani tashkil etish imkonini beradi. Masalan, Internet kompyuter tarmoKi keng tarkalgan, ommaviylashgan global kompyuter tarmoKidir. Uning tarkibiga

erkin ravishda birlashgan tarmoklar kiradi. Uning nomining uzi «tarmoklar orasida» ma'nosini bildiradi. Internet aloxida tarmoklarni birlashtirgan. Shuning uchun u katta imkoniyatlarga ega. Uzining shaxsiy kompyuteri orkali Internetning ixtiyoriy abonenti axborotni boshka shaxarga uzatishi, Vashington kongressi kutubxonasidagi adabiyotlar katalogini kurib chikishi, Nyu-Yorkdagi metropoliten muzeyining eng sunggi kurgazmasining rasmlari bilan tanishib chikishi, tarmokka ulangan abonentlar bilan konferentsiyada yoki uyinda ishtirok etishi mumkin. Internetning asosiy yacheykasini maxalliy kompyuter tarmoklari tashkil etadi.

Kompyuterlarni maxalliy tarmokka ulashning uch kurinishi mavjud:

- xalkasimon;
- shinali;
- yulduzsimon.

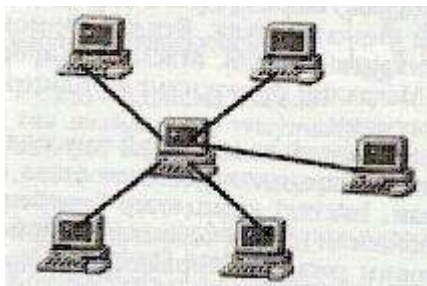
Xalkasimon bog'lanishda kompyuterlar yopik chizik buyicha boKlangan buladi. Tarmokning kirish kismi chikish kismi bilan ulangan buladi. Axborot xalka buyicha kompyuterdan kompyuterga utadi.



Tarmokning shinali boKlanishida ma'lumotlar uzatuvchi kompyuterdan shina buyicha xar ikki tomonga uzatiladi.



Yulduzsimon boglanishda markaziy kompyuter mavjud bulib, unga kolgan barcha kompyuterlar boglangan buladi.



TCP/IP protokollar stekining asosiy transport vositasini tarmoqlararo o'zaro aloqa protokoli – Internet Protocol (IP) tashkil qiladi. IP protokolining asosiy funktsiyasiga quydagilar kiradi:

tarmoqlar o'rtasida uni funktsiyalashtirilgan formada har xil turdagi adres axborotini almashish.

tarmoqlar o'rtasidagi paket uzunligining turli xil maksimal qiymati bilan uzatilganida uni yig'ish va bo'laklash.

IP-paketi sarlavxa va malumotlar maydonidan iborat. Paket sarlavxasida quyidagi maydonlar bo'ladi:

- IP protokol versiyasini ko'rsatuvchi versiya nomeri (VERS) maydoni. hozirgi vaqtda 4 versiyadan barcha fayllarda foydalanishmoqda va Ipng (Ipnex generation) deb ataluvchi 6 versiyaga o'tishga tayyorgarlik ko'rilmogda.

- IP paketining sarlavxa uzunligi (HLEN), 4 bit joy oladi va sarlavxa uzunligining qiymatini ko'rsatib, 32 bitli so'zlarda o'lchangan. Odatda sarlavxa 20 bayt (5 ta 32 bitli so'z) uzunlikda bo'ladi, faqat hizmatchi axborotlarning ortganda bu uzunlik zaxira maydoniga (IPOPTIONS) qo'shimcha baytlardan foydalanish hisobiga ortgan bo'lishi mumkin.

- Servis turi maydoni (SERVICE TYPE), 1 bayt joy oladi va paket navbatini o'rnatish va marshrutni tanlash shartlarining ko'rinishlarini bildiradi. Bu maydonning birinchi 3 biti paket navbati (PRECEDENCE) maydon osti maydonini bildiradi. Navbat o'rnatish 0 (normal paket)dan 7 (axborotni boshqaruvchi paket)gacha bo'lishi mumkin. Marshrutizatorlar va kompyuterlar paketini navbati bo'yicha qabul qiladi va muximroq bo'lgan paketlarni birinchi navbatda qayta ishlaydi. Servis turi maydoni ham 3 bitdan iborat bo'ladi va marshrutni tanlash shartlarini aniqlaydi. O'rnatilgan D (delay) bitda marshrut paketni eltishni kechikishini minimallashtiradi haqidagi gap yuritiladi. T bit – chiqish yoki kirishni maksimal holatiga o'tkazish, R bit esa – almashinish (paketlarda) ishonchliligini maksimal holatga o'tkazish.

- Umumiy uzunlik maydoni (TOTAL LENGTH), 2 bayt joy oladi va sarlavxa va maydoni hisobiga paketning umumiy uzunligini ko'rsatadi.

- Paket identifikatori (IDENTIFICATION) maydoni, 2 bayt joy oladi va paketni aniqlaydi.

- Bayroqchalar (FLAGS) maydoni, 3 bayt joy oladi va paketni qismlarga ajratish imkoniyatlarini ko'rsatadi, undan tashqari paketni oraliq yoki oxirgi ekanligini aniqlab beradi.

- Aralash fragment (FRAGMENT OFFSET) maydoni, 13 bit joy oladi. Umumiy ma'lumotlar maydonining kirish paketidan ushbu paketning aralash ma'lumotlar maydoninibaytlarda ko'rsatish uchun foydalaniladi. Tarmoqda paketlar almashinuvida uning qisimlarini yig'ish/bo'laklash bilan shug'ullanadi.

- Yashash vaqti (TIME TO LIVE) maydoni, 1 bayt joyni oladi va paketni tarmoq bo'ylab ko'chirish chegaraviy muxlatni bildiradi. Paketning yashash davri sekundlarda o'lchanadi va IP protokolining uzatuvchi vositasiga manba sifatida beriladi.

- Yuqori satxli protokol identifikatori (PROTOCOL), 1 baytni egallaydi va paketni yuqori satxdagi qaysi protokoliga tegishli ekanligini ko'rsatadi (misol uchun TCP, UDP yoki RIP protokollari bo'lishi mumkin).

- Nazorat yig'indisi (HEADE CHECKSUM), 2 baytni egallaydi. U barcha sarlavxalar bo'yicha xisoblanadi.

- Adres manbai maydoni (SOURCE IP ADDRES) va belgilangan adres (DESTINATION IP ADDRES) maydonlari, bir xil 32 bit uzunlikda va bir xil strukturada.

- Zaxira maydoni (IP OPTIONS). Bu maydon muxim hisoblanmaydi va tarmoqni sozlashda ishlatiladi. Bu maydon bir nechta maydon osti maydonlardan tashkil topgan. Bu maydon osti maydonlarda marshrutizatorlardan o'tayotgan paketlarni ro'yxatga olish, ushbu tizim xavfsizligini kiritish va yana vaqtincha belgilash mumkin. Paket ma'lumotlar maydoni maksimal uzunligi maydon razryadi bilan chegralangan. Bu kattalik 65535 baytni tashkil qiladi.

2.2. Delphi dasturlash tilining tarmoq bilan ishlovchi komponentalari.

Delphi da ixtiyoriy komponentalarida ko'rinishi jihatidan uchga bo'linadi.

1. Komponent konteyner – bunda bu komponent o'zi ichiga yana boshqa komponentni olishi mumkin. (Masalan: Form, Panel, GroupBox va h.k.).

2. Komponent vizual - bunda bu komponent o'zi ko'rinmaydi uni chaqirish orqali ko'rish mumkin. (Masalan: Faylni ochish, saqlash yoki rang tanlash muloqot oynasi komponentlari).

3. Komponenta - bu komponent oddiy komponenta.

Delphi da ixtiyoriy komponentalarda (ob'ektlarda) uch xil buyruqlar bo'ladi.

1. Xususiyat (Properties) – Bunda komponentalarni ko'rinishi xossalari kiradi. (Masalan: Color, Width, Height, Caption va h.k.)

2. Hodisa (Events) – Bunda komponentalarni ustida sodir bo'ladigan jarayonlari kiradi. (Masalan: OnClick, OnDblClick, OnKeyPress va h.k.)

3. Metod – Bunda komponent ustida bajariladigan amallar kiradi. (Masalan Memo komponentiga LoadFromFile metodi orqali biror faylni ochilishi va h.k.)

TIPEdit – bu komponenta bizga IP adressni xatoliksiz kiritishimiz uchun yordam beradi. Bu komponenta komponentlar palitrasida mavjud emas, quyidagi kod yordamida bu komponentani hosil qilamiz:

```
constructor TIPEdit.Create(AOwner: TWinControl; Rect: TRect);
```

```
begin
```

```
  InitCommonControl(ICC_INTERNET_CLASSES);
```

```
  FHandle:= CreateWindow(WC_IPADDRESS, nil, WS_CHILD or  
WS_VISIBLE,
```

```
    Rect.Left, Rect.Top, Rect.Right, Rect.Bottom, AOwner.Handle, 0, hInstance,  
nil);
```

```
  FFont := CreateFont(-11, 0, 0, 0, 400, 0, 0, 0, DEFAULT_CHARSET,  
    OUT_DEFAULT_PRECIS, CLIP_DEFAULT_PRECIS,  
DEFAULT_QUALITY,
```

```
    DEFAULT_PITCH or FF_DONTCARE, 'MS Sans Serif');
```

```
  SendMessage(FHandle, WM_SETFONT, FFont, 0);
```

```
  Text := '0.0.0.0';
```

end;

destructor TIPEdit.Destroy;

begin

 DeleteObject(FFont);

 inherited;

end;

function TIPEdit.GetText: String;

begin

 SendMessage(FHandle, IPM_GETADDRESS, 0, Longint(PDWORD(@FIP)));

 Result := IntToStr(FIRST_IPADDRESS(FIP))+

 '.' + IntToStr(SECOND_IPADDRESS(FIP)) +

 '.' + IntToStr(THIRD_IPADDRESS(FIP)) +

 '.' + IntToStr(FOURTH_IPADDRESS(FIP));

end;

procedure TIPEdit.SetText(const Value: String);

function MakeIPAddressEx(b1, b2, b3, b4: Char):LPARAM;

begin

 Result := MAKEIPADDRESS(DWORD(b1), DWORD(b2), DWORD(b3),
DWORD(b4));

end;

var

 Tmp: TInAddr;

begin

 Tmp.S_addr := inet_addr(PChar(Value));

 if Tmp.S_addr = INADDR_NONE then Exit;

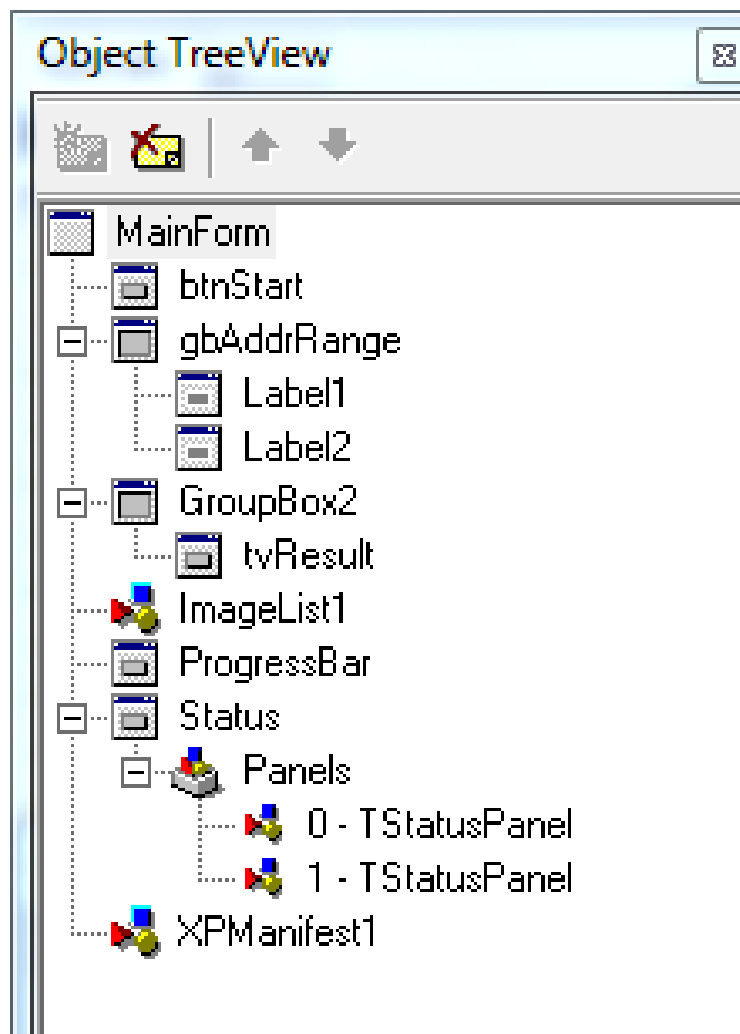
```

with Tmp.S_un_b do
    FIP := MakeIPAddressEx(s_b1, s_b2, s_b3, s_b4);
    SendMessage(FHandle, IPM_SETADDRESS, 0, FIP);
end;

```

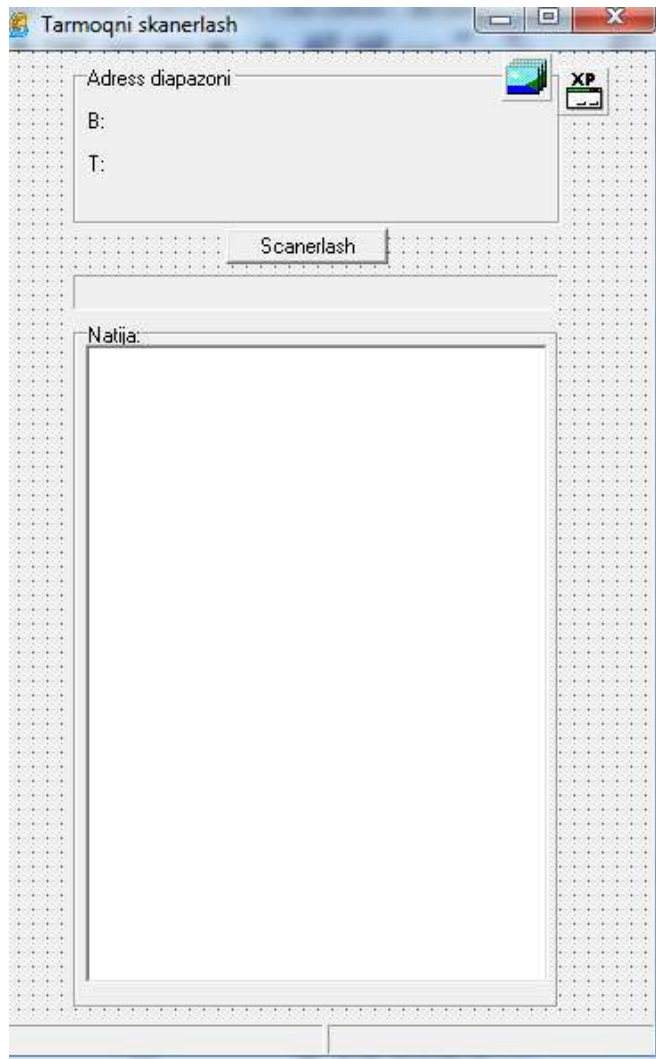
2.3. Tarmoqni scannerlovchi va boshqaruvchi dastur yaratish.

Bu dasturni yaratish uchun formamizga quyidagiga komponentalarni qo'yamiz (2.3.1-chizma):



2.3.1 – chizma.

va oynamiz quyidagi holatga keladi (2.3.2 – chizma):



2.3.2 – chizma.

Dastur kodi:

```
unit Unit1;
```

```
interface
```

```
uses
```

```
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
  Dialogs, StdCtrls, ComCtrls, CommCtrl, Winsock, XPMan, ImgList, Menus;
```

```
resourcestring
```

```
  RES_UNKNOWN = 'Aniqlanmagan';  
  RES_THREADCOUNT = 'Bajarilgan potok: %d';  
  RES_COMPCOUNT = 'Topildi: %d';  
  RES_ERR_RANGE = 'Noto`g`ri IP adress';
```

```
const
```

```
  WSA_TYPE = $101;
```

```
type
```

```
  LMSTR = LPWSTR;  
  NET_API_STATUS = DWORD;
```

```

PShareInfo1 = ^_SHARE_INFO_1;
_SHARE_INFO_1 = record
  shi1_netname: LMSTR;
  shi1_type: DWORD;
  shi1_remark: LMSTR;
end;
TShareInfo1 = _SHARE_INFO_1;

TIPEdit = class
private
  FHandle: THandle;
  FIP: Integer;
  FFont: Integer;
  function GetText: String;
  procedure SetText(const Value: String);
public
  constructor Create(AOwner: TWinControl; Rect: TRect);
  destructor Destroy; override;
  property Text: String read GetText write SetText;
end;

TScanThread = class(TThread)
private
  FIP: Integer;
  FRes: TStringList;
  function GetCompName(const Addr: Integer): String;
  procedure Scan;
  procedure UpdateTree;
  procedure IncCount;
  procedure DecCount;
protected
  procedure Execute; override;
public
  property IP: Integer read FIP write FIP;
end;

TMainForm = class(TForm)
  btnStart: TButton;
  gbAddrRange: TGroupBox;
  Label1: TLabel;
  Label2: TLabel;
  ProgressBar: TProgressBar;
  GroupBox2: TGroupBox;
  tvResult: TTreeView;
  Status: TStatusBar;
  XPManifest1: TXPManifest;
  ImageList1: TImageList;
  procedure btnStartClick(Sender: TObject);
  procedure FormCreate(Sender: TObject);
private
  IPFrom, IPTo: TIPEdit;

```

```

    FThreadCount, FCompFound: Integer;
    procedure SetThreadCount(const Value: Integer);
    procedure SetCompFound(const Value: Integer);
public
    property ThreadCount: Integer read FThreadCount write SetThreadCount;
    property CompFound: Integer read FCompFound write SetCompFound;
end;

function NetShareEnum(servername: LMSTR; level: DWORD; var bufptr: Pointer;
    prefmaxlen: DWORD; entriesread, totalentries,
    resume_handle: LPDWORD): NET_API_STATUS; stdcall; external 'Netapi32.dll';
function NetApiBufferFree(Buffer: Pointer): NET_API_STATUS; stdcall; external
'Netapi32.dll';

var
    MainForm: TMainForm;

implementation

{$R *.dfm}

{ TIPEdit }

constructor TIPEdit.Create(AOwner: TWinControl; Rect: TRect);
begin
    InitCommonControl(ICC_INTERNET_CLASSES);
    FHandle:= CreateWindow(WC_IPADDRESS, nil, WS_CHILD or WS_VISIBLE,
        Rect.Left, Rect.Top, Rect.Right, Rect.Bottom, AOwner.Handle, 0, hInstance, nil);
    FFont := CreateFont(-11, 0, 0, 0, 400, 0, 0, 0, DEFAULT_CHARSET,
        OUT_DEFAULT_PRECIS, CLIP_DEFAULT_PRECIS, DEFAULT_QUALITY,
        DEFAULT_PITCH or FF_DONTCARE, 'MS Sans Serif');
    SendMessage(FHandle, WM_SETFONT, FFont, 0);
    Text := '0.0.0.0';
end;

destructor TIPEdit.Destroy;
begin
    DeleteObject(FFont);
    inherited;
end;

function TIPEdit.GetText: String;
begin
    SendMessage(FHandle, IPM_GETADDRESS, 0, Longint(PDWORD(@FIP)));
    Result := IntToStr(FIRST_IPADDRESS(FIP))+
        '.' + IntToStr(SECOND_IPADDRESS(FIP)) +
        '.' + IntToStr(THIRD_IPADDRESS(FIP)) +
        '.' + IntToStr(FOURTH_IPADDRESS(FIP));
end;

procedure TIPEdit.SetText(const Value: String);

```

```

function MakeIPAddressEx(b1, b2, b3, b4: Char):LPARAM;
begin
    Result := MAKEIPADDRESS(DWORD(b1),    DWORD(b2),    DWORD(b3),
DWORD(b4));
end;

var
    Tmp: TInAddr;
begin
    Tmp.S_addr := inet_addr(PChar(Value));
    if Tmp.S_addr = INADDR_NONE then Exit;
    with Tmp.S_un_b do
        FIP := MakeIPAddressEx(s_b1, s_b2, s_b3, s_b4);
        SendMessage(FHandle, IPM_SETADDRESS, 0, FIP);
    end;

    { TScanThread }

procedure TScanThread.DecCount;
begin
    MainForm.ThreadCount := MainForm.ThreadCount - 1;
end;

procedure TScanThread.Execute;
begin
    inherited;
    Synchronize(IncCount);
    Scan;
    Synchronize(DecCount);
end;

function TScanThread.GetCompName(const Addr: Integer): String;
var
    WSA: TWSAData;
    Host: PHostEnt;
    Err: Integer;
begin
    Result := RES_UNKNOWN;
    Err := WSASStartup(WSA_TYPE, WSA);
    if Err <> 0 then
    begin
        Exit;
    end;
    try
        if Addr = INADDR_NONE then Exit;
        Host := gethostbyaddr(@Addr, SizeOf(Addr), PF_INET);
        if Assigned(Host) then
            Result := Host.h_name
        else
            //ShowMessage(SysErrorMessage(GetLastError));
    finally
        WSACleanup;
    end;
end;

```

```

    end;
end;

procedure TScanThread.IncCount;
begin
    MainForm.ThreadCount := MainForm.ThreadCount + 1;
end;

procedure TScanThread.Scan;
type
    TShareInfo1Array = array of TShareInfo1;
var
    entriesread, totalentries: DWORD;
    Info: Pointer;
    I: Integer;
    CompName: PWideChar;
begin
    CompName := StringToOleStr(GetCompName(FIP));
    if CompName = RES_UNKNOWN then Exit;
    FRes := TStringList.Create;
    try
        FRes.Add(CompName);
        if NetShareEnum(CompName, 1, Info, DWORD(-1), @entriesread,
            @totalentries, nil) = 0 then
            try
                if entriesread > 0 then
                    begin
                        for I := 0 to entriesread - 1 do
                            FRes.Add(TShareInfo1Array(@(Info^))[I].shi1_netname);
                        Synchronize(UpdateTree);
                    end;
                finally
                    NetApiBufferFree(Info);
                end;
            finally
                FRes.Free;
            end;
        end;
    end;

procedure TScanThread.UpdateTree;
var
    I: Integer;
    Root: TTreeNode;
begin
    MainForm.tvResult.Items.BeginUpdate;
    try
        Root := MainForm.tvResult.Items.Add(nil, FRes.Strings[0]);
        for I := 1 to FRes.Count - 1 do
            MainForm.tvResult.Items.AddChild(Root, FRes.Strings[I]);
            MainForm.CompFound := MainForm.CompFound + 1;
        finally
            MainForm.tvResult.Items.EndUpdate;
        end;
    end;
end;

```

```

    end;
end;

{ TMainForm }

procedure TMainForm.btnStartClick(Sender: TObject);
var
    I, AFrom, ATo: Integer;
    Prefix: String;

function ValidRange: Boolean;
var
    F, T: TInAddr;
begin
    F.S_addr := inet_addr(PChar(IPFrom.Text));
    T.S_addr := inet_addr(PChar(IPTo.Text));
    Result := (F.S_un_b.s_b1 = T.S_un_b.s_b1) and
               (F.S_un_b.s_b2 = T.S_un_b.s_b2) and
               (F.S_un_b.s_b3 = T.S_un_b.s_b3);
    if Result then
    begin
        AFrom := Integer(F.S_un_b.s_b4);
        ATo := Integer(T.S_un_b.s_b4);
        Prefix := IntToStr(Integer(F.S_un_b.s_b1)) + '.' +
                  IntToStr(Integer(F.S_un_b.s_b2)) + '.' +
                  IntToStr(Integer(F.S_un_b.s_b3)) + '.';
        ProgressBar.Max := ATo - AFrom;
        ProgressBar.Position := 0;
    end
    else
        MessageDlg(RES_ERR_RANGE, mtError, [mbOK], 0);
end;

begin
    CompFound := 0;
    ThreadCount := 0;
    tvResult.Items.Clear;
    if ValidRange then
    begin
        btnStart.Enabled := False;
        for I := AFrom to ATo do
            with TScanThread.Create(False) do
            begin
                IP := inet_addr(PChar(Prefix + IntToStr(I)));
                FreeOnTerminate := True;
                Resume;
            end;
        end;
    end;
end;

procedure TMainForm.FormCreate(Sender: TObject);
begin

```

```

IPFrom := TIPEdit.Create(gbAddrRange, Rect(Label1.Left+Label1.Width+8, 24, 160,
24));
IPFrom.Text := '192.168.137.0';
IPTo := TIPEdit.Create(gbAddrRange, Rect(Label2.Left+Label2.Width+8, 54, 160, 24));
IPTo.Text := '192.168.137.255';
end;

procedure TMainForm.SetCompFound(const Value: Integer);
begin
  FCompFound := Value;
  Status.Panels.Items[1].Text := Format(RES_COMPCOUNT, [Value]);
  Application.ProcessMessages;
end;

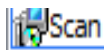
procedure TMainForm.SetThreadCount(const Value: Integer);
begin
  if Value < FThreadCount then
    ProgressBar.Position := ProgressBar.Max - Value;
  FThreadCount := Value;
  Status.Panels.Items[0].Text := Format(RES_THREADCOUNT, [Value]);
  if Value = 0 then
    begin
      ProgressBar.Position := 0;
      btnStart.Enabled := True;
    end;
  Application.ProcessMessages;
end;

end.

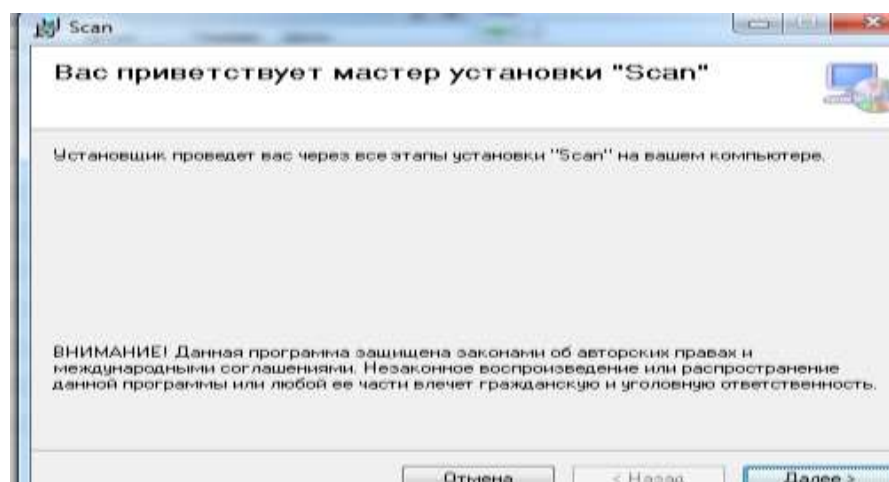
```

2.4. Yaratilgan dasturdan foydalanish uchun qo'llanma.

Dasturdan foydalanish uchun avval dasturni o'rnatib olish kerak. Buning uchun quyidagi ketma-ketlikni bajarish kerak:

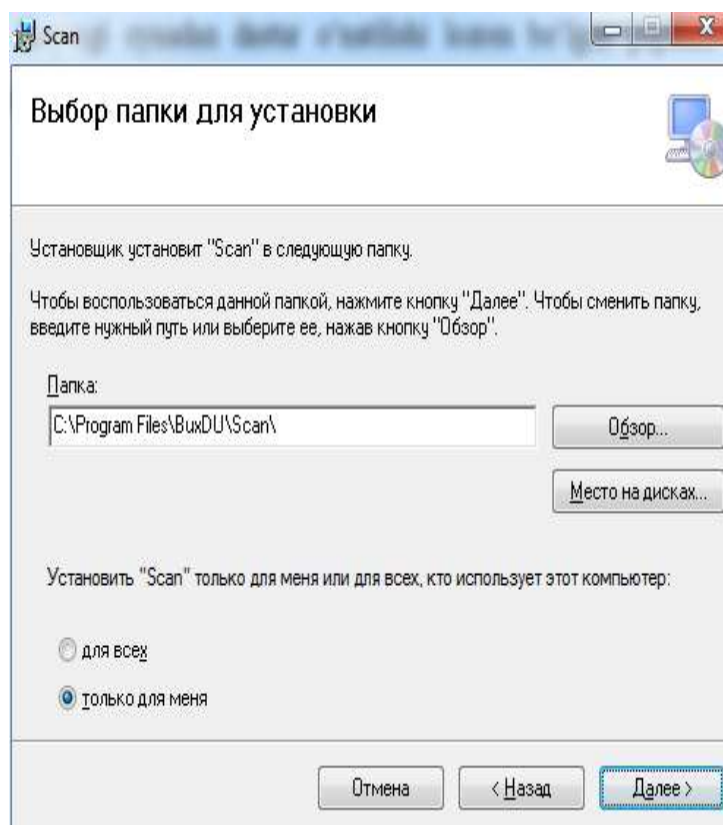


ustanovichni faylini tanlaymiz va sichqonchanning chap tugmasini ikki marta bosamiz. Quyidagi oyna ochiladi, undan “Далее” tugmasini bosamiz (2.4.1 - chizma):



2.4.1 – chizma.

Keyingi oynadan dastur o'natilishi lozom bo'lgan papkani tanlaymiz va “Далее” tugmasini bosamiz (2.4.2-chizma), keyingi oynadan ham “Далее” tugmasini bosamiz:



2.4.2 – chizma.

Dastur o'rnatilgandan so'ng “Рабочий стол” uning yoqlig'i paydo bo'ladi va shu yerdan biz uni ishga tushirishimiz mumkin.

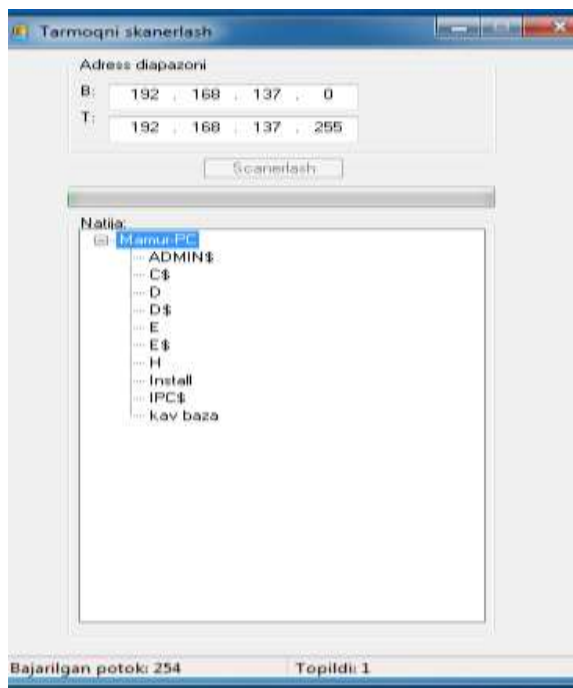


Dastur ishga tushganda quyidagi oyna hosil bo'ladi (2.4.3-chizma):



2.4.3 – chizma.

Adres diapazoniga kerakli adresni kiritamiz va scanerlash tugmasini bosamiz, “Natiga” bo’limiga bizga berilgan diapazon bo’yicha mayjud kompyuterlar ro’yxati va shu kompyuterlarda ruxsat berilgan papkalar nomi chiqadi (2.4.4-chizma)



2.4.4 – chizma.

Xulosa

Bugungi kunda Informatika va axborot kommunikatsiya texnologiyalari sohasi jadal rivojlanmoqda, shu bilan birgalikda kompyuter tarmoqlari ham. Biz yoshlar o'zimiz va o'zimizdan keyingi avlodlarga Informatika va axborot kommunikatsiya texnologiyalari sirlarini chuqurroq o'rganishlari uchun yordam berishimiz kerak.

Men bu kurs ishini yozish va bu dasturni yaratish davomida o'z bilim va ko'nikmalarimni oshirib oldim. Bu dasturni yaratishda Delphi 7 dasturlash tilining men uchun yangi komponentalari bilan tanishdim va kompyuter tarmoqlari bo'yicha ham yangi bilimlarga ega bo'ldim.

Foydalanilgan adabiyotlar ro'yxati:

1. Karimov I.A. Yuksak malakali mutaxassislar –taraqqiyot omili. T., O'zbekiston, 1995 y, 30 – 31 b.
2. Sh.F. Madraximov va boshqalar, Informatika va programmalash, Toshkent-2005.
3. Н. А. Отаханов. DELPHI тилини ўқитишдаги айрим муаммолар ва уларни бартараф қилиш услубиёти «ФАРФОНА» нашриёти 2008 йил
4. Jalolov O.I, Xayatov X.U, Jalolov F.I D E L P H I MUHITIDA DASTURLASH (o'quv-uslubiy qo'llanma).
5. Raxmatov M. N. Informasion texnika vositalaridan qo'llanma. Toshkent.«Oliy maktab». 1989. 379 b