

O'ZBEKISTON RESPUBLIKASI ALOQA, AXBOROTLASH TIRISH VA
TELEKOMMUNIKASIYA TEXNOLOGIYALARI DAVLAT QO'MITASI

TOSHKENT AXBOROT TEXNOLOGIYALAR UNIVERSITETI
SAMARQAND FILIALI
AXBOROT VA PEDAGOGIK TEXNOLOGIYALAR FAKULTETI

UMUMKASBIY FANLAR KAFEDRASI

5521900 -“Informatika va axborot texnologiyalari” yo'nalishi bo'yicha bakalavr
akademik darajasini olish uchun

BITIRUV MALAKAVIY ISHI

**Mavzu: Obyektlar to'plamini mantiqiy klassifikatorlar asosida sinflashtirish
algoritmi va dasturiy ta'minoti**

Ish kafedraning 2013 yil ____

_____dagi ____ -sonli

majlisda muhokama qilindi va

Bajardi: 403-guruh talabasi

himoyaga tavsiya etildi.

_____Mamatqulov Oyebek

Kafedra mudiri _____

Ilmiy rahbar:

Dots. Rahimov N.O.

_____ t.f.n. Aliyev Sh.

“ ____ ” _____ 2013y.

Samarqand – 2013

MUNDARIJA

	Bet
Kirish	3
1-bob Mantiqiy klassifikatorlar asosida qiyofalarni anglovchi tizimlar	7
1.1 Asosiy tushunchalar va ta'riflar	7
1.2 Qiyofalarni anglovchi tizimlarning turlari	10
1.3 Obyektlar to'plamini yadro obyektlar yordamida sinflashtirish	11
2-bob Obyektlar to'plamini mantiqiy klassifikatorlar asosida sinflashtirish algoritmi	17
2.1 Masalaning qo'yilishi va ishning maqsadi	17
2.2 Obyektlarni taqqoslash qoidalari	19
2.3 Obyektlar to'plamini mantiqiy klassifikatorlar asosida sinflashtirish algoritmi	22
3-bob Dasturiy ta'minot va undan foydalanish tartibi	30
3.1 Dasturiy ta'minotning tasnifi	30
3.2 Dasturiy ta'minotdan foydalanish tartibi	33
3.3 Komp'yuter bilan ishlashda texnika xavfsizligi qoidalari va talablari	36
Xulosa va takliflar	40
Foydalanilgan adabiyotlar	41
Ilova 1 Oynalarni hosil qilish kodi	42
Ilova 2 Dasturning kodi	46

KIRISH

Mavzuning dolzarbligi. O'zbekiston Respublikasi Prezidentining 2012 yil 21 martdagi «Zamonaviy axborot-kommunikasiya texnologiyalarini yanada joriy etish va rivojlantirish chora-tadbirlari to'g'risida» PQ-1730 sonli qarori Respublikamizda axborot-kommunikasiya texnologiyalari taraqqiyotida yana bir muhim bosqich bo'lib qoladi. Prezidentimiz I.Karimov XXI asrni ma'naviy-ma'rifat asri, ilm-fan madaniyat va axborotlar asri deb ta'riflar ekan, bu bilan hozirgi davrning muhim belgisi va xususiyatini ham belgilab berganini payqab olish qiyin emas.

Mantiqiy klassifikatorlar[2-4] asosida qiyofa(obyekt)larni anglovchi tizimlar hozirgi vaqtda xalq xo'jaligining turli tarmoqlarida uchraydigan murakkab formallashgan masalalarni yechishda samarali qo'llanmoqda. Bu tizimlar ma'lum bir predmet sohasiga taalluqli aprior va aposterior ma'lumotlar asosida etalon tanlov ko'rinishda berilgan ma'lumotlarni qayta ishlash jarayonida obyektlardagi belgilar fazosini qisqartirish va ular asosida yangi obyektlarni anglash yoki obyektlar to'plami ko'rinishda berilgan ma'lumotlarni qayta ishlash natijasida ularni sinflashtirish masalalarini yechishda qo'llaniladi.

Mantiqiy klassifikatorlar asosida qiyofalarni anglovchi tizimlar hozirgi davrga kelib juda muhim amaliy samara bermoqda. Jumladan, texnika va tibbiyotda tashxis qo'yish, gelologiyada foydali qazilmalarning holati va turini anglash, gdrogeologiyada ichimlik suvlarining holatini anglash, qishloq xo'jaligi maxsulotlarini tartiblash kabi masalalarni va hodisalarni yechishga olib keladi[2-4].

Mantiqiy klassifikatorlar asosida qiyofalarni anglovchi tizimlar etalon tanlov ko'rinishda berilgan ma'lumotlarni qayta ishlash jarayonida obyektlardagi belgilar fazosini qisqartirish va qisqartirilgan belgilar fazosiga mos qaror qabul qiluvchi qoidalarni qurish hamda ular asosida yangi obyektlarni anglash masalalarini yechishda

keng qo'llaniladi. Bu yo'nalishda ko'plab usullar[2-8] yaratilgan va algoritmlar[2, 3, 4, 7, 8] ishlab chiqilgan va amaliy masalalarni yechishda qo'llanib kelinmoqda. Lekin amaliyotda shunday masalalar uchraydiki, bunda ma'lumotlar etalon tanlov ko'rinishida berilmaydi, ya'ni boshlang'ich ma'lumotlar sifatida faqat obyektlar to'plami va ulardagi belgilar beriladi. Bunday holda masala obyektlar to'plamini sinflashtirishga olib kelinadi. Mantiqiy klassifikatorlar asosida qiyofalarni anglovchi tizimlar bunday masalalarni yechishda qo'llanilmagan. Bu esa bitiruv malakaviy ishining dolzarbligini belgilaydi.

Bitiruv malakaviy ishida ma'lum bir predmet sohasiga taalluqli obyektlar haqidagi aprior va aposterior ma'lumotlar etalon tanlov ko'rinishda emas, balki ma'lumotlar obyektlar to'plami ko'rinishda berilganda obyektlar to'plamini mantiqiy klassifikatorlar asosida sinflash masalasi qaralgan.

Tadqiqot maqsadi - Mantiqiy klassifikatorlar asosida obyektlar to'plamini sinflash va yangi obyektlarni anglovchi tizimni yaratish algoritmi va dasturiy ta'minotini yaratishdan iborat. Bu maqsadga erishish uchun quyidagi masalalar yechiladi:

- mantiqiy klassifikatorlar asosida qiyofalarni anglovchi tizimni yaratish usullari va algoritmlarini o'rganish va taxlil qilish;
- obyektlar to'plamini mantiqiy klassifikatorlar asosida sinflash va yangi obyektlarni anglash algoritmini ishlab chiqish;
- ishlab chiqilgan algoritm asosida amaliy masalalarni yechish uchun dasturiy vositani yaratish.

Tadqiqot obyekti. Biror predmet sohaga taalluqli obyektlar to'plami beriladi. Masalan, tibbiyot sohasida bemorlar to'plami, qishloq xo'jaligi sohasida mahsulotlar(mevalar) to'plami, texnika sohasida dvigatellar to'plami va h.k. Ushbu obyektlar to'plamini mantiqiy klassifikatorlar asosida sinflarga ajratish va yangi obyektlarni anglash masalasi qaraladi.

Tadqiqot usullari. Ishda mantiqiy klassifikatorlar asosida obyektlar to'plamini sinflashtirish tizimini yaratish masalasining matematik statistika, to'plamlar nazariyasi hamda diskret matematika usullaridan foydalanildi

Ishning yangiligi. Mantiqiy klassifikatorlar asosida qiyofalarni anglovchi tizimlar ma'lumotlar etalon tanlov ko'rinishida berilganda keng qo'llaniladi. Lekin amaliyotda shunday masalalar uchraydiki, bunda ma'lumotlar etalon tanlov ko'rinishida berilmaydi, ya'ni boshlang'ich ma'lumotlar sifatida faqat obyektlar to'plami va ulardagi belgilar beriladi. Bunday holda masala obyektlar to'plamini sinflashtirishga olib kelinadi. Mantiqiy klassifikatorlar asosida qiyofalarni anglovchi tizimlar bunday masalalarni yechishda qo'llanilmagan. Bu esa bitiruv malakaviy ishining yangiligini belgilaydi. Bu ishda mantiqiy klassifikatorlar asosida obyektlar to'plamini sinflarga ajratuvchi algoritmi va dasturiy ta'minoti yaratilgan.

Ishning amaliy ahamiyati. Ishlab chiqilgan algoritmi va dasturiy ta'minoti tibbiyot sohasida bemorlar to'plamini, qishloq xo'jaligi sohasida mahsulotlar(mevalar) to'plamini, texnika sohasida dvigatellar to'plamini sinflarga ajratishda qo'llash mumkin.

Bitiruv malakaviy ishining tuzilishi va hajmi. Bitiruv malakaviy ishi matni komp'yuterda terilgan 60 bet hajmidan iborat bo'lib, uning strukturasini kirish, 3ta bob, xulosa, foydalanilgan adabiyotlar ro'yxati va 2 ta ilova tashkil qiladi.

1-bob mantiqiy klassifikatorlar asosida qiyofalarni anglovchi tizimlarga bag'ishlangan bo'lib, unda asosiy tushunchalar va ta'riflar, qiyofalarni anglovchi tizimlarning turlari va obyektlar to'plamini yadro obyektlar yordamida sinflashtirish usullari keltirilgan.

2-bobda obyektlar to'plamini mantiqiy klassifikatorlar asosida sinflashtirish algoritmi keltirilgan bo'lib, masalaning qo'yilishi va ishning maqsadi, obyektlarni taqqoslash qoidalari va obyektlar to'plamini mantiqiy klassifikatorlar asosida sinflashtirish algoritmi bayon etilgan.

3-bob dasturiy ta'minot va undan foydalanish tartibiga bag'ishlangan bo'lib, unda dasturiy ta'minotning tasnifi, dasturiy ta'minotdan foydalanish tartibi va komp'yuter bilan ishlashda texnika xavfsizligi qoidalari va talablari keltirilgan.

Xulosa qismida bitiruv malakaviy ishi ishida olingan asosiy natijalar keltirilgan.

Adabiyotlar qismida bitiruv malakaviy ishini bajarishda foydalanilgan asosiy adabiyotlar ro'yxati keltirilgan.

Ilovalarda dasturning ishlash jarayoni, oynalarni hosil qilish kodi va dasturning kodi keltirilgan.

1-bob. Mantiqiy klassifikatorlar asosida qiyofalarni anglovchi tizimlar

1.1. Asosiy tushunchalar va ta'riflar

Mantiqiy klassifikator[2-3] – bu shunday qoidaki, unda obyektlar bir-biri bilan solishtiriladi. Solishtirish natijasida obyektlar o'xshash bo'lsa, ularga 1 qo'yiladi, agarda o'xshash bo'lmasa 0 qo'yiladi. Solishtirish qoidalari har bir belgi uchun tanlanadi. Mantiqiy klassifikator tushunchasi qiyofalarni anglashda va sinflashtirishda muhim, chunki bunday masalalarda obyektlarni solishtirish jarayoni asosiy o'rinni egallaydi.

Mantiqiy klassifikatorlar asosida qiyofalarni anglovchi tizimni yaratishning asosiy vazifasi – obyektlar to'plami berilganda ularni mantiqiy klassifikatorlar asosida sinflarga ajratish va yangi obyektlarni hosil qilingan mantiqiy klassifikatorlar yordamida anglashdan iborat.

Mantiqiy klassifikatorlar asosida obyektlarni anglovchi tizimlarda eng muhim tushunchalaridan biri - bu sinf tushunchasidir. Sinf obyektlar to'plami bo'lib, bu obyektlar to'plami bitta yoki bir nechta belgilar asosida tanlab olinadi [5].

Obraz(Qiyofa) – obyekt, hodisa yoki jarayon bo'lib, ular ustida anglash yoki sinflash amallari amalga oshiriladi.

Belgi – obyektida o'lchash mumkin bo'lgan oddiy yoki murakkab xarakteristikasi.

Belgilar alfaviti – obyektlar guruxini tavsiflovchi barcha mumkin bo'lgan belgilar ro'yxati.

Obraz(Qiyofa)lar to'plami – shunday to'plamki, unda obyektlar nomi va ularning belgilar ro'yxati beriladi.

Etalon tanlov – shunday tanlovki, unda obyektlar nomi va soni, obyektlarning belgilari nomi va soni hamda obyektlarning qaysi sinfdagi joylashganligi oldindan berilgan bo'ladi.

Nazorat tanlov – shunday tanlovki, unda yangi obyektlar joylashgan bo'ladi.

O'qitish jarayoni – bu shunday jarayonki, bunda obyektlarni belgilari asosida o'rganish jarayonida qaror qabul qiluvchi funktsiya quriladi.

O'qituvchi bilan o'qitish jarayoni - bunda etalon tanlovni tuzishda mutaxassis(ekspert) bevosita qatnashadi. Bunda sinflar haqida aprior ma'lumotlar ma'lum bo'ladi va obyektlar xamda belgilar alfaviti berilgan bo'ladi.

O'qituvchisiz o'qitish jarayoni - bunda etalon tanlov mavjud bo'lmaydi, faqat obyektlar to'plami xamda belgilar alfaviti berilgan bo'ladi. Bu holda mutaxassis(ekspert) bevosita qatnashmaydi. Shuning uchun sinflar haqida aprior ma'lumotlar ma'lum bo'lmaydi.

Avtomatik o'qitish jarayoni - bunda etalon tanlov mavjud bo'lmaydi, faqat obyektlar to'plami xamda belgilar alfaviti berilgan bo'ladi. Bu holda mutaxassis(ekspert) bevosita qatnashmaydi. Shuning uchun sinflar haqida aprior ma'lumotlar ma'lum bo'lmaydi. Sinflar va ularning soni avtomatik ravishda hosil qilinadi.

Qaror qabul qiluvchi qoida - bu o'qituvchi bilan, o'qituvchisiz yoki avtomatik o'qitish jarayonida hosil qilinadigan belgilar tizimiga mos keluvchi qoida bo'lib, undan nazorat tanlovdagi yangi obyektlarni anglashda yoki sinflashda foydalaniladi.

O'qitish jarayoni o'qituvchi bilan o'qitish jarayoni, o'qituvchisiz o'qitish jarayoni va avtomatik o'qitish jarayoniga bo'linadi. O'qituvchi bilan o'qitish jarayonida sinflar soni va obyektlarning qaysi sinfga qarashlilik, obyektlar va ularning belgilari soni beriladi. O'qituvchi bilan o'qitish jarayonida bizga oldindan sinflarga ajratilgan obyektlar to'plami beriladi. Bunda mashina yordamida xuddi mutaxassis(ekspert)ga

o'xshab obyektlarni sinflarga ajratuvchi qaror qabul qiluvchi funksiyani topish talab etiladi[2, 4].

O'qituvchisiz o'qitish jarayonida[2, 4, 6, 8] sinflar alfaviti va obyektlarning qaysi sinflarga qarashliligi berilmaydi. Obyektlar va ularning belgilari soni beriladi. Obyektlarni o'qitish jarayonida sun'iy sinflar hosil qilinadi. Bunda ikkita xol bo'lishi mumkin. Birinchisi - bu hosil qilinuvchi sinflar soni oldindan ma'lum bo'lishi mumkin. Ikkinchi xolda hosil qilinuvchi sinflar soni oldindan noma'lum, ya'ni sinflar soni obyektlarni o'qitish jarayonida hosil qilinadi. Bunday masala avtomatik(taksonomiya) sinflashtirish masalasi deyiladi.

Etalon tanlovda sinflar alfaviti oldindan berilgan bo'lib, har bir obyektning qaysi sinfga qarashliligi ham oldindan ma'lum bo'ladi. O'qituvchi tanlovda har bir obyekt belgilar alfaviti yordamida beriladi. Obyektlarning belgilari sifatida sonlar, chiziqlar, simvollar va h.k. lar ishlatiladi. Etalon tanlovning hajmi deganda ushbu tanlovda sinflar, obyektlar va obyektlardagi belgilar soni tushuniladi. Hisoblashlar uchun ketadigan EHMning vaqti etalon tanlovning hajmiga bevosita bog'liq bo'ladi.

Anglash jarayoni obyektlar to'plamni sinflashda hosil qilinadigan belgilar tizimiga mos bo'lgan xal qiluvchi qoidani qurishdan iborat. Qaror qabul qiluvchi qoida yordamida nazorat tanlovda obyektlar hosil qilingan sinflarga sinflashtiriladi. Qaror qabul qiluvchi qoida obyektlar to'plamining kompaktiligi qarab chiziqli, bo'lakli – chiziqli va mantiqiy va boshqa ko'rinishlarda bo'lishi mumkin [8].

Mantiqiy klassifikatorlar asosida qiyofalarni sinflash tizimlarini yaratish belgilar alfavitini tuzish, belgilar alfavitini qisqartirish, qisqartirilgan belgilar alfavitiga mos qaror qabul qiluvchi qoidani qurish, qaror qabul qiluvchi qoida asosida sinflashtirish usullaridan foydalanish kabi asosiy hamda belgilar alfavitiga yangi belgilarni kiritish, belgilarni qisqartirish masalasini sharxlash, sinflashtirish masalasini baxolash uchun boshqa usullarni qo'llash va sinflashtirish jarayonini optimallashtirish kabi qo'shimcha bosqichlardan iborat.

1.2. Qiyofalarni anglovchi tizimlarning turlari

Qiyofalarni anglovchi tizimlar texnika tashxisida avtomobillarni texnik ko'rikdan o'tkazish, tibbiy tashxisda bemorlarga tashxis qo'yish, qishloq xo'jaligida ekiladigan maydonlarning holatini aniqlash, kosmosdan turib ekiladigan maydonlarning o'lchamini va hosilini aniqlash hamda mevalarni shakli, rangi va o'lchamlari bo'yicha taxlash, harbiy sohada tizimlarning holatini avtomatik ravishda nazorat qilish masalalarida keng qo'llaniladi[4, 7, 8].

Mantiqiy klassifikatorlar asosida qiyofalarni anglovchi tizimni yaratishda eng asosiy masalalardan biri – bu obyektlarning belgilar alfavitini tuzishdan iborat. Belgilar alfaviti determinalli, ehtimolli, mantiqiy va strukturali belgilardan iborat bo'lishi mumkin. Determinalli belgilar aniq va doimiy sonli, ehtimolli belgilar tavakkalli, mantiqiy belgilar rost yoki yolg'on qiymatlar va strukturali belgilar simvol ko'rinishida berilishi mumkin.

Mantiqiy klassifikatorlarni hosil qilish jarayonida boshlang'ich belgilar alfaviti qisqartirilishga olib kelinadi. Bunda obyektlarni taqqoslash jarayonida o'xshash belgilar qoldiriladi va o'xshash bo'lmagan belgilar olib tashlanadi. Hosil qilingan qisqartirilgan belgilar alfavitiga mos qaror qabul qiluvchi qoida quriladi va ular asosida yangi obyektlar sinflashtiriladi.

Sinflashtirish sifatini oshirish maqsadida belgilar alfavitiga yangi belgilarni kiritish mumkin. Belgilar alfavitini qisqartirish masalasini sharxlashda qisqartirish masalasi qanday mezonlar asosida amalga oshirilganligi, qanday algoritmlardan foydalanganligi va ularning sifatiy ko'rsatgichlari keltiriladi.

Mantiqiy klassifikatorlar asosida qiyofalarni anglovchi tizimlar obyektlarda berilgan belgilar alfavitida qatnashayotgan belgilar turiga qarab oddiy va murakkab tizimlarga bo'linadi. Oddiy anglovchi tizimlarda obyektlarda berilgan belgilar bir xil tipli bo'lsa (determinalli yoki ehtimolli yoki mantiqiy yoki strukturali), murakkab

anglovchi tizimlarda belgilar alfavitida belgilar turli tipli bo'ladi(determinalli va mantiqiy yoki determinalli va ehtimolli yoki determinalli va strukturali yoki determinalli, strukturali va ehtimolli). Oddiy anglovchi tizimlarda qaror qabul qilishda obyektlarning alohida belgilaridan foydalaniladi. Murakkab anglovchi tizimlarda obyektlarning alohida belgilari bilan bir qatorda belgilar majmuasidan foydalaniladi. Bu tizimlar bir pog'anali va ko'p pog'onali bo'linadi. Bir pog'onali mantiqiy klassifikatorlar asosida qiyofalarni anglovchi tizimlarni yaratish masalasida belgilar haqidagi ma'lumotlar o'rganilayotgan obyektlardan to'g'ridan-to'g'ri o'lchash natijasida olinadi. Ko'p pog'onali mantiqiy klassifikatorlar asosida qiyofalarni anglovchi tizimlarni yaratish masalasida belgilar haqidagi ma'lumotlar o'rganilayotgan obyektlardan nisbiy o'lchash natijasida olinadi.

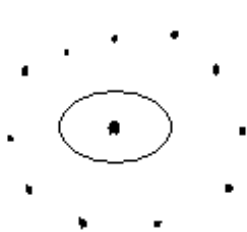
1.3. Obyektlar to'plamini yadro obyektlar yordamida sinflashtirish

Obyektlar to'plamini yadrolar yordamida sinflashtirishda $S_1, S_2, \dots, S_m$ obyektlar to'plamini shunday $K_1, K_2, \dots, K_l$ sinflarga bo'lish kerakki, bu bo'lish imkoniyat darajasida samarali bo'lsin. Masalani bunday formallashtirish $S_1, S_2, \dots, S_m$ obyektlar to'plamini $K_1, K_2, \dots, K_l$ sinflarga bo'lishda ajratuvchi funksiyani hosil qilishga olib keladi. Demak obyektlar to'plami T_{nm} tanlov ko'rinishda berilganda, T_{nm} tanlovdan T_{nml} tanlovni hosil qilish uchun shunday ajratuvchi funksiyani topish kerakki, bu bu funksiya obyektlarni sinflarga yetarli darajada sifatli va ishonchlilik bilan ajratsin.

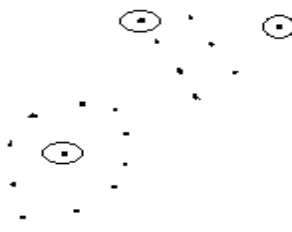
T_{nm} tanlovdan T_{nml} tanlovni hosil qilishda ajratuvchi funksiyaning yetarli darajada sifatli va ishonchli bo'lishligi T_{nm} tanlovdan tanlanadigan yadrolarga bevosita bog'liq.

T_{nm} tanlovdan yadrolarni tanlash turli usullarda amalga oshiriladi[4, 7, 8]:

- 1) yadro obyekt sifatida T_{nm} tanlovdagi obyektlar to'plamidan o'rtacha obyekt olinadi;
- 2) yadro obyekt sifatida T_{nm} tanlovdagi obyektlar to'plamidan ixtiyoriy k ta obyektlar olinadi;
- 3) yadro obyekt sifatida T_{nm} tanlovdagi obyektlar to'plamidan normal taqsimot asosida tanlanadi;
- 4) yadro obyekt sifatida T_{nm} tanlovdagi obyektlar to'plamida bir-biridan eng uzoqda joylashgan obyektlar olinadi;
- 5) yadro obyekt sifatida T_{nm} tanlovdagi obyektlar to'plamida to'g'ri chiziqqa eng yaqin obyekt olinadi;
- 6) yadro obyekt sifatida T_{nm} tanlovdagi obyektlar to'plamida tengsizliklar asosida aniqlanadi, ya'ni N_1 yadro $x < 5$; N_2 yadro $x > 5$; $y > 4$; N_3 yadro esa $x > 5$; $y < 4$ tengsizliklar bilan aniqlanadi.



1)



2)



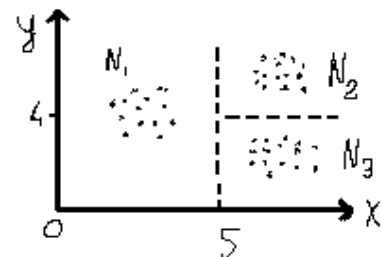
3)



4)



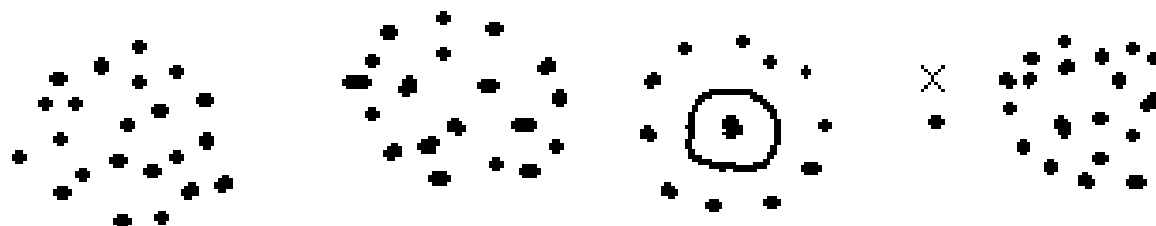
5)



6)

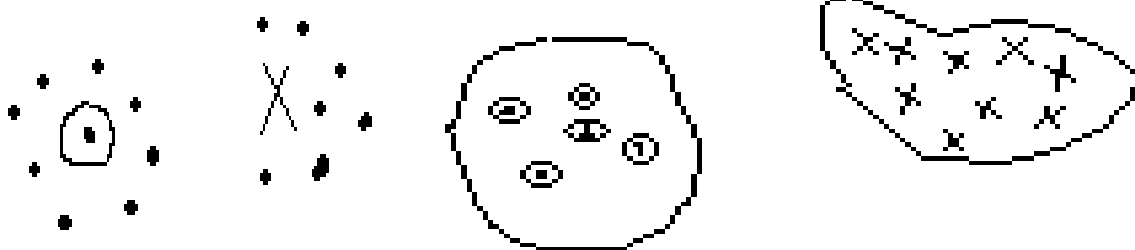
Umumiy holda T_{nm} tanlovdagi obyektlar to'plamini sinflarga yadrolar yordamida ajratish quyidagicha amalga oshiriladi: qandaydir usullar (evristik, tavakkal) bilan k ta boshlang'ich yadrolar topiladi. Undan keyin tanlangan yadrolar asosida sinflashtiriladi. Hosil bo'lgan sinflarda yana yangi yadrolar topiladi va ushbu yadrolar asosida obyektlar yana sinflashtiriladi va oldingi sinflar bilan solishtiriladi. Agar solishtirish natijasida sinflar va yadrolar o'zgarmasa, u holda sinflashtirish jarayoni tugallangan hisoblanadi, aks holda yana yangi yadrolar topiladi va ularga nisbatan sinflashtirish jarayoni davom etadi[4, 7, 8].

Bu jarayonni quyidagi rasm orqali tushuntiramiz. Rasmda “.” belgi yordamida T_{nm} tanlovdagi obyektlarni belgilaymiz. Sinflar soni K oldindan berilganda (bu rasmda $K=2$) har bir sinfdan bittadan yadroviy obyekt tanlanadi va bu yadrolarni mos ravishda “.” va “X” bilan belgilaymiz. Bu yadrolarni ixtiyoriy olishimiz mumkin[4, 7, 8].



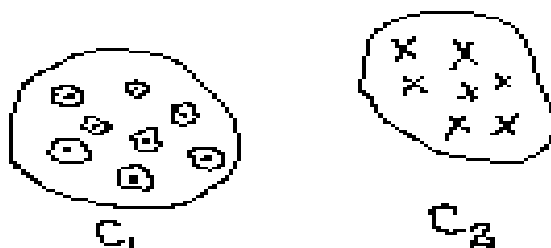
a)

b)



v)

g)

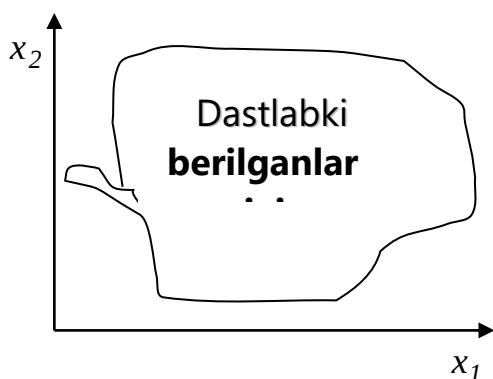


d)

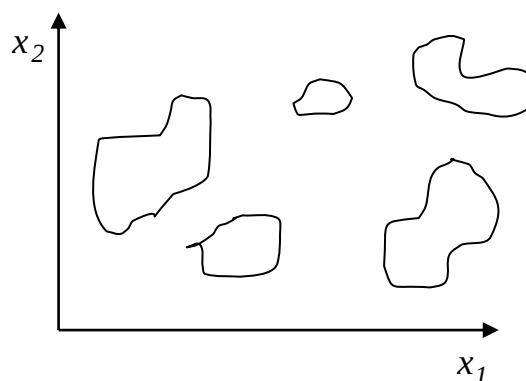
bu yerda $\bullet$ -boshlang'ich obyekt, $\odot$ - C_1 sinf, X- C_2 sinf.

Undan keyin har bir yadroga eng yaqin bo'lgan obyektни olinadi va ushbu obyektlarga nisbatan C_1 va C_2 sinflarni hosil qilinadi. Keyin yangi yadrolar topiladi. Bu yadrolar sifatida C_1 va C_2 sinflardagi markaziy obyektlarga yaqin bo'lgan obyektlar olinadi va bu obyektlar orqali vaqtinchalik C_1 va C_2 sinflar hosil qilinadi va h. bu jarayon C_1 va C_2 sinflar o'zgarmaguncha davom ettiriladi va hosil bo'lgan sinflar o'zgarmas sinflar sifatida qabul qilinadi [4, 7, 8].

Har bir sinf uchun yadrolarni topishning yana bir usullaridan biri, bu yadro sifatida har bir sinfdan markaziy obyektни olishdir. Ushbu markaziy obyektlar yordamida T_{nm} tanlovdagi obyektlar to'plami sinflashtiriladi.



1. 1 - rasm.



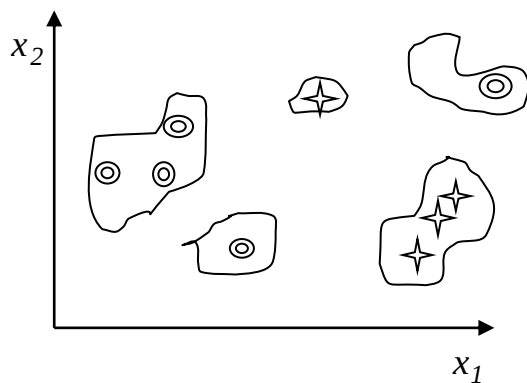
1. 2 - rasm.

Keyin ushbu sinflardagi markaziy obyektlarga eng yaqin bo'lgan boshqa ikkita obyekt orqali yana boshqa sinflar hosil qilinadi. Bu jarayon sinflar o'zgaras holda kelguncha davom ettiriladi.

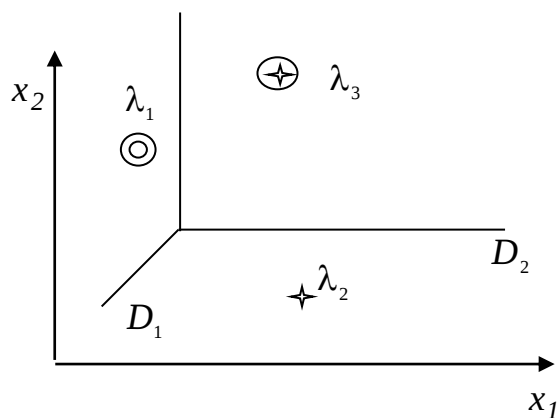
Yadrolarni tanlashning ikkinchi bir usuli, bu ehtimol qonunlariga asoslangan bo'lib, bunda yadro sifatida ixtiyoriy obyekt olinadi. Shuning uchun bu holda bir nechta sinflar olinishi mumkin. Lekin amaliyot ko'rsatadiki, bu holda ba'zi bir obyektlar majmuasi ko'p hollarda bitta sinfga tushadi, ba'zi bir obyektlar goh bir sinfga, goh boshqa sinfga tushadi. Bir sinfga tushadigan obyektlarni kuchli obyektlar va sinflarni o'zgartiradigan obyektlarni esa kuchsiz obyektlar deb ataymiz.

Endi klassifikatsiyalash sxemasini x_1 va x_2 belgilar bilan berilgan obyektlar misolida keltiramiz. Bu dastlabki berilganlar to'plami 1.1 - rasmda keltirilgan. Boshlang'ich qiymatlar sifatida 1.2 - rasmda keltirilgan obyektlar to'plamini qaraymiz. Algoritmning birinchi etapida obyektlar orasidagi eng kichik masofani optimallashtirish kriteriyasi asosida 1.3 - rasmda keltirilgan sohalarni (sinflarni) olamiz.

1.2 - rasmda keltirilgan har bir sinfda tayanch obyektlarni tanlaymiz. x_1 o'x x_2 tekislikni D_1, D_2, D_3 sohalarga ajratuvchi chegaralarni o'tkazamiz. Bu chegaralar 1.4 - rasmda keltirilgan.



1.3 – rasm



1.4 - rasm

2-bob. Obyektlar to'plamini mantiqiy klassifikatorlar asosida sinflashtirish algoritmi

2.1. Masalaning qo'yilishi va ishning maqsadi

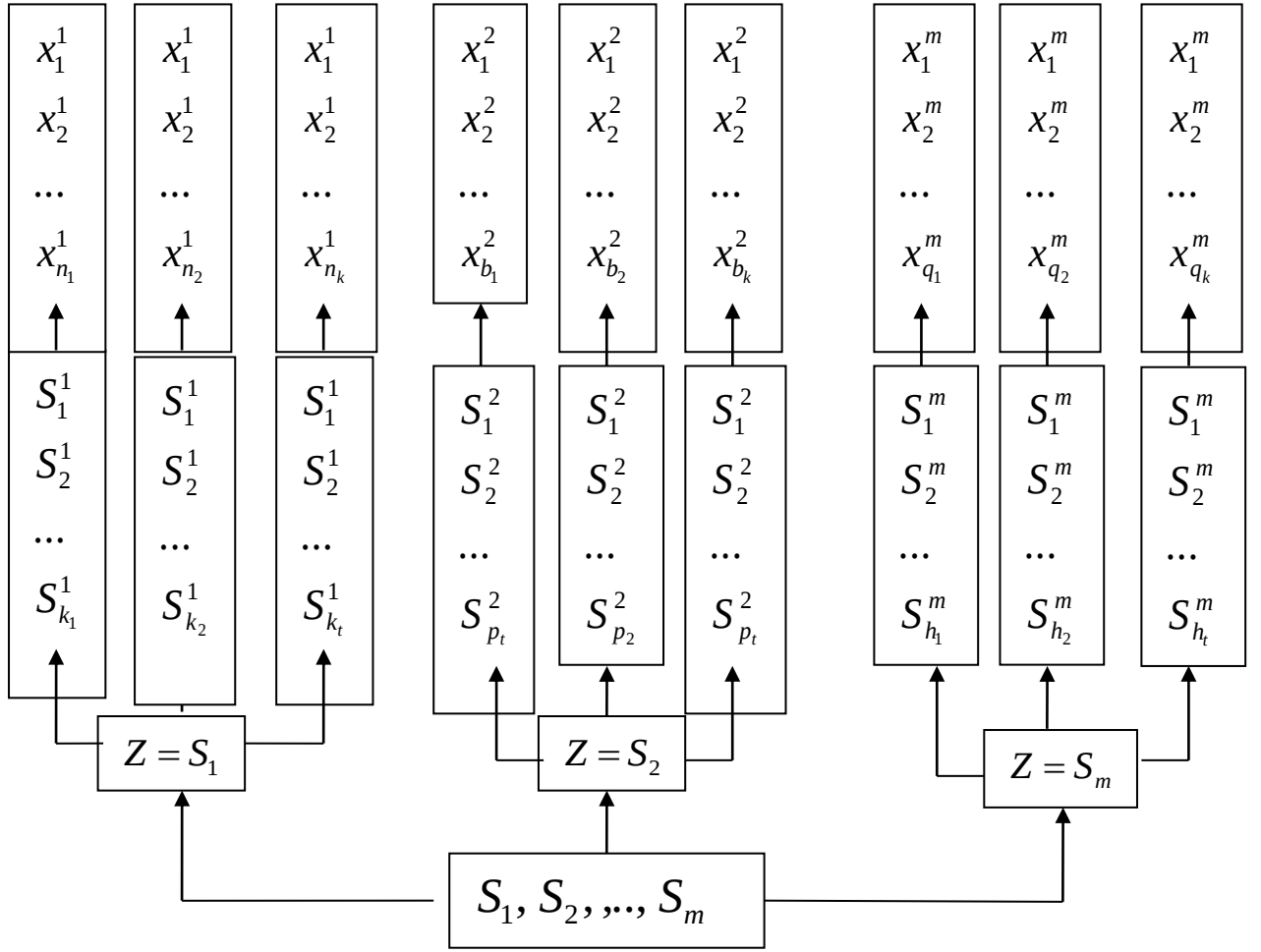
Masalaning qo'yilishini bayon qilish qulay bo'lishi uchun quyidagi belgilashlarni kiritamiz[2, 4]. Aniqlanayotgan j - raqamli obyektни S_j orqali, bu obyektning i - belgisini a_{ij} bilan belgilaymiz. Agar a_{ij} belgi, $i=1,2,\dots,n$, berilgan L_i , $i=1,2,\dots,n$ to'plamda o'z qiymatini qabul qilsa, u holda $a^{(j)} = (a_{1j}, a_{2j}, \dots, a_{nj})$ nabor va S_j , $j=1,2,\dots,m$, obyektlar to'plami to'liq bo'ladi. Har biri n ta belgilar bilan xarakterlangan m ta to'liq obyektlar to'plamini T_{nm} jadval ko'rinishda kiritamiz, bu yerda m qator - obyektlar soni, n ustun - belgilar sonini bildiradi. T_{nm} jadvalni to'liq jadval deb ataymiz.

Obyektlar to'plami T_{nm} jadval ko'rinishida berilgan bo'lsin. Bu yerda n - belgilar (ustunlar) soni, m - berilgan $S_i = \alpha_{i1}, \dots, \alpha_{in}$ obyektlar soni bo'lib, $\alpha_{ij} \in L_j$, qayerdakim L_j - j raqamli belgilar alfaviti, ya'ni

$$L_j = \{ \{0,1,\dots,K^j\}, \{0,1\} \}, \quad 2 < K^j < \infty, \quad j = \overline{1,n},$$

Talab etiladi, obyektlar to'plami ko'rinishda berilgan T_{nm} tanlovdan mantiqiy klassifikatorlar to'plamini hosil qilish va ular yordamida T_{nm} tanlovdagi obyektlar to'plamini hosil qilingan mantiqiy klassifikatorlar yordamida sinflash, ya'ni T_{nm} tanlovdan T_{nml} ni hosil qilishdan iborat. Bu yerda l - hosil qilinadigan sinflar soni.

$S_1, S_2, \dots, S_m$ obyektlar to'plami T_{nm} tanlov ko'rinishda berilgan bo'lsa, u holda T_{nm} tanlovdagi $S_1, S_2, \dots, S_m$ obyektlar to'plamini Z tayanch obyektlar yordamida hosil qilingan mantiqiy klassifikatorlar yordamida sinflashtirish masalasi sxematik ko'rinishda 2.1- rasmda keltirilgan.



2.1- rasm.

Mantiqiy klassifikatorlar asosida qiyofalarni anglovchi tizimni yaratish usullari algoritmlari [2-4] ishlarda keltirilgan bo'lib, bu ishlarda T_{nml} etalon tanlov oldindan beriladi va ushbu etalon tanlovdagi har bir K_i sinf ushbu sinfdagi joylashgan $\{S_1^i, S_2^i, \dots, S_{t_i}^i\}, (t_i \leq m_i)$ tayanch obyektlar asosida hosil qilinadigan mantiqiy klassifikatorlar bilan qoplanadi. Har bir K_i sinfdan ixtiyoriy ravishda $\{S_1^i, S_2^i, \dots, S_{t_i}^i\}, (t_i \leq m_i)$ tayanch obyektlar tanlanadi va ushbu $\{S_1^i, S_2^i, \dots, S_{t_i}^i\}, (t_i \leq m_i)$ tayanch obyektlar boshqa sinflardagi barcha

$S_i^j = \alpha_{i1}^j, \alpha_{i2}^j, \dots, \alpha_{in}^j$ obyektlar bilan solishtiriladi va solishtirish natijasida $\{S_1^i, S_2^i, \dots, S_{t_i}^i\}, (t_i \leq m_i)$ tayanch obyektlarni boshqa sinflardagi obyektlardan farqlovchi mantiqiy klassifikatorlar topiladi. Tayanch obyektlar sifatida har bir sinfda joylashgan barcha $S_i^j = \alpha_{i1}^j, \alpha_{i2}^j, \dots, \alpha_{in}^j; (j = 1, l; i = 1, m_l)$ obyektlar qatnashadi. Bu algoritmlarga asosan har bir K_i sinfda joylashgan $\{S_1^i, S_2^i, \dots, S_{t_i}^i\}, (t_i \leq m_i)$ tayanch obyektlarni boshqa sinflardagi obyektlardan farqlovchi mantiqiy klassifikatorlar to'plami hosil qilinadi. Undan keyin mantiqiy klassifikatorlar to'plami optimallashtiriladi. Tayanch obyektlar asosida hosil qilingan mantiqiy klassifikatorlarning optimal to'plamiga mos qaror qabul qiluvchi qoidalar quriladi va ushbu qoidalar asosida nazorat tanlovdagi $S_i^* = \alpha_{i1}^*, \alpha_{i2}^*, \dots, \alpha_{in}^*, i = 1, k$ obyektlar har bir K_i sinf uchun hosil qilingan mantiqiy klassifikatorlarning optimal to'plami bilan solishtiriladi va qaysi sinfga ko'proq o'xshasa, o'sha sinfga qarashli deb topiladi.

[2-4] ishlarda olingan natijalardan xulosa qiladigan bo'lsak, bu ishlarda T_{nml} etalon tanlov berilgan holda mantiqiy klassifikatorlar asosida qiyofalarni anglovchi tizimlar yaratilgan.

Ushbu bitiruv malakaviy ishida T_{nm} etalon tanlov berilmaganda, ya'ni faqat obyektlar to'plami T_{nm} tanlov ko'rinishda berilganda, ushbu tanlovdan mantiqiy klassifikatorlar to'plamini hosil qilish va ular yordamida T_{nm} tanlovdagi obyektlar to'plamini hosil qilingan mantiqiy klassifikatorlar yordamida sinflash masalasi qaralgan, ya'ni T_{nm} tanlovdan T_{nml} ni hosil qilishdan iborat.

Ushbu maqsadga erishish uchun bitiruv malakaviy ishida quyidagi masalalar qaraladi va yechiladi:

- mantiqiy klassifikatorlar asosida qiyofalarni anglovchi tizimlarning asosiy tushunchalarini o'rganish;
- obyektlar to'plamini sinflash usullari va algoritmlarini taxlil qilish;
- obyektlar to'plamini mantiqiy klassifikatorlar asosida sinflashtirish algoritmini ishlab chiqish;
- ishlab chiqilgan algoritmning tadbiqu uchun dasturiy ta'minoti yaratish;
- yaratilgan dasturiy ta'minotni amaliy masalalarni yechishda nazoratdan o'tkazish.

2.2. Obyektlarni taqqoslash qoidalari

Obyektlar to'plami T_{nm} jadval ko'rinishida berilganda ularni mantiqiy klassifikatorlar asosida sinflashda obyektlarni bir-biri bilan o'xshashligini aniqlashda taqqoslash qoidalaridan foydalaniladi. Taqqoslash qoidasi deganda ma'lum bir shart asosida har bir obyektlar jufti $S_i = \alpha_{1i}, \dots, \alpha_{ni}$ va $S_j = \alpha_{1j}, \dots, \alpha_{nj}$ ga 0 yoki 1 qo'yuvchi $d(S_i, S_j) = 1$ qoida tushuniladi. Taqqoslash qoidalari sifatida porogli, chiziqli va mantiqiy qoidalarini keltirish mumkin. Porogli taqqoslash qoidalari poroglarni (chegaraviy qiymat) berishga asoslangan bo'lib, bunda porogning qiymati har bir belgi uchun oldindan aniq beriladi yoki maxsus algoritmlar asosida izlab topiladi. Ushbu qoidaga misol sifatida [2-4]

$$d(S_i, S_j) = \begin{cases} 1, & \text{agar } |\alpha_{ki} - \alpha_{kj}| \leq \varepsilon_k, \\ 0, & \text{aks holda.} \end{cases}$$

bu yerda ε_k - L_k , $k = \overline{1, n}$ belgilar alfavitida berilgan porog.

Chiziqli taqqoslash qoidasi

$$d(S_i, S_j) = \begin{cases} 1, & \text{agar } \alpha_{ki} = \alpha_{kj}, k = \overline{1, n} \\ 0, & \text{aks holda.} \end{cases}$$

va mantiqiy taqqoslash qoidasini

$$d(S_i, S_j) = \bigwedge_{\alpha=1}^{\eta} (\alpha_{\alpha i} = \alpha_{\alpha j}) \vee (\alpha_{\alpha i} > \alpha_{\alpha j}) \vee (\alpha_{\alpha i} < \alpha_{\alpha j})$$

ko'rinishlarda berilishi mumkin.

Aytaylik, T_{nm} tanlov uchun k ($k \leq n$) ta taqqoslash qoidasi $d_1, d_2, \dots, d_k$ berilgan bo'lsin. Ushbu taqqoslash qoidalari har xil tipli belgilar uchun turli fazolarda berilishi mumkin. Ushbu taqqoslash qoidalar tizimiga k uzunlikka ega bo'lgan boshqaruvchi mantiqiy vektori $\tilde{b} = b_1, b_2, \dots, b_k, b_j \in \{0, 1\}$ ni mos qo'yamiz[2]. Agar $b_j = 0$ bo'lsa, u holda j - raqamli taqqoslash qoidasi bo'yicha $S_i \hat{a} \hat{a} S_j$ obyektlar taqqoslanmaydi va agarda $b_j = 1$ bo'lsa, u holda j - raqamli taqqoslash qoidasi bo'yicha $S_i \hat{a} \hat{a} S_j$ taqqoslanadi. $d_\beta(S_i, S_j)$ taqqoslash qoidasida birinchi S_i obyektini tayanch obyekt sifatida olsak, u holda biz $d_\beta(Z, S_j)$ taqqoslash qoidasini hosil qilamiz, bu yerda $Z (Z = \forall S_i)$ obyekt tayanch obyekt sifatida qabul qilinadi. Ko'rinib turibdiki $d_\beta(Z, S_j)$ taqqoslash qoidasi tayanch obyektga va o'ziga o'xshash obyektlarga 1 qiymat qo'yadi va o'xshash bo'lmagan obyektlarga 0 qiymat qo'yadi. Natijada Z tayanch obyektga o'xshash obyektlar to'plami M_z^1 va o'xshash bo'lmagan obyektlar to'plami M_z^0 hosil bo'ladi.

Aytaylik T_{nm} tanlovdagi $S_i = \alpha_{i1}, \alpha_{i2}, \dots, \alpha_{in}, i = \overline{1, m}$ obyektlar uchun boshqaruvchi mantiqiy vektori $\tilde{b} = b_1, b_2, \dots, b_k$, taqqoslash qoidalar tizimi $d_1, d_2, \dots, d_k$ berilgan bo'lib, tayanch obyekt sifatida $Z = z_1, z_2, \dots, z_n, z_j \in L_j, j = \overline{1, n}$ tanlangan bo'lsin.

T_{nml} etalon tanlovdagi tayinlangan $Z = z_1, z_2, \dots, z_n$ tayanch obyektни chiziqli taqqoslash qoidasi yordamida ixtiyoriy $S_i = \alpha_{i1}, \alpha_{i2}, \dots, \alpha_{in}$ bilan solishtirganda $b_1, b_2, \dots, b_n$ boshqaruvchi vektorning qiymati

$$b_j = \begin{cases} 1, & \text{agar } z_j = \alpha_{ij} \\ 0, & \text{aks holda, } j = 1, n. \end{cases}$$

aniqlanadi.

Agarda T_{nml} etalon tanlovdagi tayinlangan $Z = z_1, z_2, \dots, z_n$ tayanch obyektни porogli taqqoslash qoidasi yordamida ixtiyoriy $S_i = \alpha_{i1}, \alpha_{i2}, \dots, \alpha_{in}$ bilan solishtirsak, u holda $b_1, b_2, \dots, b_n$ boshqaruvchi vektorning qiymati

$$b_j = \begin{cases} 1, & \text{agar } |z_j - \alpha_{ij}| \leq \varepsilon_j \\ 0, & \text{boshqa hollarda, } j = 1, n. \end{cases}$$

topiladi.

$b_1, b_2, \dots, b_n$ boshqaruvchi mantiqiy vektor taqqoslash qoidalari asosida T_{nm} tanlovdagi $S_i, i = \overline{1, m}$ obyektlar orasidan $Z = z_1, z_2, \dots, z_n$ tayanch obyektga o'xshashlarini tanlab oladi.

$b_1, b_2, \dots, b_n$ boshqaruvchi vektorning $b_k, k = \forall j$ qiymati $b_k = 1$ bo'ladi, agarda S_i obyekt Z tayanch obyektga o'xshasa, $b_k = 0$ bo'ladi, agarda S_i obyekt Z tayanch obyektga o'xshamasa. $b_1, b_2, \dots, b_n$ boshqaruvchi vektorning $b_k, k = \forall j$ qiymati T_{nm} tanlovdagi navbatdagi S_{i+1} yangi obyektни Z tayanch obyekt bilan taqqoslaganda o'zgarib boradi, ya'ni $b_1, b_2, \dots, b_n$ boshqaruvchi vektorda $b_k = 1, k = \forall j$ bo'lgan elementlar qoldiriladi, $b_k = 0, k = \forall j$ bo'lgan elementlar hisobga olinmaydi. Bu

prosedura $b_1, b_2, \dots, b_n$ boshqaruvchi vektorda barcha elementlar 0 ga teng bo'lguncha davom ettiriladi, ya'ni $b_1 = 0, b_2 = 0, \dots, b_n = 0$ bajarilguncha.

2.3. Obyektlar to'plamini mantiqiy klassifikatorlar asosida sinflashtirish algoritmi

Aytaylik, T_{nm} tanlovdagi $S_i = \alpha_{i1}, \alpha_{i2}, \dots, \alpha_{in}, i = \overline{1, m}$ obyektlar uchun boshqaruvchi mantiqiy vektori $\tilde{b} = b_1, b_2, \dots, b_k$ taqqoslash qoidalar tizimi $d_1, d_2, \dots, d_k$ berilgan bo'lib, tayanch obyekt sifatida $Z = z_1, z_2, \dots, z_n, z_j \in u_j, j = \overline{1, n}$ tanlangan bo'lsin.

T_{nm} tanlovdagi har bir $S_i = \alpha_{i1}, \alpha_{i2}, \dots, \alpha_{in}, i = \overline{1, m}$ obyektga taqqoslash qoidalari yordamida 1 yoki 0 qiymat qo'yuvchi

$$D(Z, S_j) = \bigwedge_{\substack{\beta=1 \\ b_1=1, b_2=1, \dots, b_k=1}}^k [b_k \rightarrow d_\beta(Z, S_j)]$$

qoidaga mantiqiy klassifikator deyiladi[2].

Har bir mantiqiy klassifikator quyidagicha xarakterlanadi[2].:

- tayanch obyekt $Z = z_1, z_2, \dots, z_n, z_\beta \in u_\beta$;
- boshqaruvchi mantiqiy vektori $\tilde{b} = b_1, b_2, \dots, b_k$;
- tayanch taqqoslash qoidalari $d_1, d_2, \dots, d_k$;
- shaxsiy to'plam osti N;
- rang $r, r = \sum_{\beta=1}^k b_\beta$.

Aytish mumkinki, mantiqiy klassifikator belgilangan taqqoslash qoidalari asosida T_{nm} tanlovdagi $S_i, i = \overline{1, m}$ obyektlar orasida Z tayanch obyektga o'xshashlarini tanlab oluvchi qoidadir.

Agar T_{nm} tanlov ustida $d_1, d_2, \dots, d_k$ taqqoslash qoidalari berilgan bo'lsa, u holda ushbu tanlovdan olingan bitta tayanch obyekt orqali hosil qilish mumkin bo'lgan mantiqiy klassifikatorlar soni 2^k ga teng bo'ladi.

Shuni ta'kidlash kerakki, agar T_{nm} tanlovdan olingan S_i obyekt Z tayanch obyektga o'xshasa, u holda mantiqiy klassifikator S_i obyektga 1 qiymat qo'yadi, agarda S_i obyekt Z tayanch obyektiga o'xshamasa, unda mantiqiy klassifikator S_i obyektga 0 qiymat qo'yadi.

Mantiqiy klassifikator formulasi shunday qurilganki, boshqaruvchi mantiqiy vektori $\tilde{b} = b_1, b_2, \dots, b_k$ tarkibida biror $b_i = 0$ qiymat uchrasa, u holda b_i qiymatga mos keluvchi d_i taqqoslash qoidasi hisobga olinmaydi. Shuning uchun mantiqiy klassifikator formulasida implikasiya amali qo'llaniladi.

Aytaylik, Z tayanch obyekt T_{nm} tanlovdan olingan bo'lsin.

Mantiqiy klassifikator $D(Z, S_j) = \bigwedge_{\substack{\beta=1 \\ b_1=1, b_2=1, \dots, b_k=1}}^k [b_k \rightarrow d_\beta(Z, S_j)] = 1$ bo'lsa,

u holda S_i obyekt Z tayanch obyektga o'xshash bo'ladi, agarda

$D(Z, S_j) = \bigwedge_{\substack{\beta=1 \\ b_1=1, b_2=1, \dots, b_k=1}}^k [b_k \rightarrow d_\beta(Z, S_j)] = 0$ bo'lsa, u holda S_i obyekt Z

tayanch obyektga o'xshash bo'lmaydi.

T_{nm} tanlovdagi obyektlar to'plamidan har bir tayanch obyektga nisbatan hosil qilinadigan mantiqiy sifatida qatnashadigan bo'lsa, u holda ushbu tanlovdan tayanch obyektlarga nisbatan hosil bo'ladigan mantiqiy klassifikatorlar soni $|N| = m \times 2^k$ aniqlanadi. Mantiqiy klassifikatorlar to'plami $|N|$ da har bir Z tayanch obyektga

nisbatan eng kamida bitta o'xshash obyekt mavjud bo'lishi uchun

$$\bigvee_{\alpha=1}^k d_{\alpha}(Z, S_j) = 1, \forall S_j \in T_{nm} \text{ bajarilishi kerak.}$$

T_{nm} tanlovdagi obyektlar to'plamidan har bir obyekt tayanch obyekt sifatida qatnashadigan bo'lsa, u holda ushbu tanlovdan har bir tayanch obyektga nisbatan hosil bo'ladigan natijaviy mantiqiy klassifikatorlarning rangi

$$Z = \sum_{i=1}^k b_{\alpha i}$$

aniqlanadi.

Endi T_{nm} tanlovdagi obyektlar to'plamidan Z tayanch obyektlarga nisbatan o'xshash obyektlarni topuvchi mantiqiy klassifikatorlarni hosil qilish algoritmini keltiramiz. Ushbu algoritim T_{nm} tanlovdan olingan Z tayanch obyektlar bilan T_{nm} tanlovdagi obyektlarni kema-ket taqqoslash natijasida hosil qilinadigan mantiqiy klassifikatorlarni topish uchun qo'llaniladi. Har bir tayanch obyektga nisbatan hosil qilingan mantiqiy klassifikatorlar o'zining to'plam ostilariga, ya'ni tayanch obyektga o'xshash obyektlar to'plam ostiga ega bo'ladi.

T_{nm} tanlovdan mantiqiy klassifikatorlarni izlab topish algoritmi quyidagi qadamlardan iborat:

1 - qadam. Obyektlar to'plami va obyektlarning belgilari T_{nm} tanlov qo'rinishda hosil qilinadi.

2 - qadam. T_{nm} tanlovdan ixtiyoriy S_i obyekt tayanch obyekt sifatida tayinlanadi, ya'ni $Z = \forall S_i$.

3 - qadam. T_{nm} tanlovdagi obyektlarning belgilar shkalasiga mos $d_1, d_2, \dots, d_k$ taqqoslash qoidalari tanlanadi:

$$d(Z, S_j) = \begin{cases} 1, & \text{agar } |z_k - \alpha_{kj}| \leq \varepsilon_k, \\ 0, & \text{aks holda.} \end{cases}$$

$$d(Z, S_j) = \begin{cases} 1, & \text{agar } z_k = \alpha_{kj}, k = 1, n \\ 0, & \text{aks holda.} \end{cases}$$

$$d(Z, S_j) = \bigwedge_{\alpha=1}^{\eta} (z_{\alpha} = \alpha_{\alpha j}) \vee (z_{\alpha} > \alpha_{\alpha j}) \vee (z_{\alpha} < \alpha_{\alpha j})$$

4 - qadam. T_{nm} tanlovda joylashgan obyektlarning t - tartibli joylashishi tayinlanadi, ya'ni

$$S_1^t, S_2^t, \dots, S_m^t \quad (2.1)$$

Agar qo'shimcha shartlar bo'lmasa, obyektlarning joylashish tartibi tavakkal holda aniqlanadi.

5 - qadam. T_{nm} tanlovda n o'lchovga ega bo'lgan boshlang'ich boshqaruvchi mantiqiy vektori $\tilde{b}^0 = \underset{n}{1_1 1_2 \dots 1_n}$ tanlanadi.

6 - qadam. Izlashning 1 - qadami. (2.1) formulada keltirilgan obyektlardan birinchi uchragan S_1^t obyekt olinadi va $\tilde{b}_1$ boshqaruvchi mantiqiy vektorining qiymati hisoblanadi:

$$b_{1\alpha} = d_{\alpha}(Z, S_1^t), \alpha = \overline{1, k'}.$$

Natijada $\tilde{b} = b_1, b_2, \dots, b_{k'}$ boshqaruvchi mantiqiy vektoriga ega bo'lgan $D^1(Z, S_1^t)$ mantiqiy klassifikatorni hosil qilamiz.

7-qadam. Hosil qilingan $\tilde{b}_1$ boshqaruvchi mantiqiy vektorning qiymati oldingi qadamda hosil qilingan $\tilde{b}^0$ boshqaruvchi mantiqiy vektorning qiymati bilan taqqoslanadi. Agar $\tilde{b}^0 = \tilde{b}^1$ bo'lsa, u holda keyingi qadamga o'tiladi. Agar

$\tilde{b}^1 = 0,0,\dots,0,0$, bo'lsa, u holda keyingi qadamga $\tilde{b}^1 = \tilde{b}^0$ boshqaruvchi mantiqiy vektor bilan o'tiladi.

8 - qadam. Izlashning j - qadami. T_{nm} tanlovda (2.1) ko'rinishda o'rnatilgan obyektlar tartibidan j - o'rinda turgan S_j^t obyekt olinadi va boshqaruvchi mantiqiy vektori $\tilde{b}^{j-1}$ dan $\tilde{b}^j$ vektor hisoblanadi:

$$b_{j\alpha} = \begin{cases} d_{j\alpha}(Z, S_j^t), & \text{agar } b_{\alpha}^{j-1} = 1 \\ 0, & \text{agar } b_{\alpha}^{j-1} = 0 \end{cases}$$

9 - qadam. $j = j+1$. Agar $j \leq m$ bo'lsa, u holda 10-qadamga, aks holda 11-qadamga o'tadi.

10- qadam. Agar $\tilde{b}^{j+1} = \tilde{b}^j$ bo'lsa, u holda Z tayanch obyektни navbatdagi obyekt bilan solishtirish uchun $\tilde{b}^j$ vektordan foydalaniladi. Agarda $\tilde{b}^{j+1} \neq \tilde{b}^j$ bo'lsa, u holda Z tayanch obyektни navbatdagi obyekt bilan solishtirish uchun $\tilde{b}^{j+1}$ vektordan foydalaniladi.

11-qadam. Z tayanch obyekt orqali mantiqiy klassifikatorlar to'plamini hosil qilish uchun 4 - qadamda $S_1^t, S_2^t, \dots, S_m^t$ obyektlarning joylashish tartibi o'zgartiriladi va algoritmi 4 – qadamga o'tadi. Obyektlarning joylashish tartiblari sonini S deb olsak, u holda ushbu prosedura $t \leq C$ shart bajarilguncha davom ettiriladi. Agarda $t > C$ bo'lsa, u holda 12- qadamga o'tiladi.

12-qadam. $i = i+1$. Agar $i \leq m$ bo'lsa, u holda T_{nm} tanlovda Z tayanch obyekt o'zgartiriladi, ya'ni $Z = \forall S_{i+1}$ tayinlanadi va 2-qadamga o'tiladi. Aks holda 13-qadamga o'tiladi.

13-qadam. T_{nm} tanlovda m ta $Z_1, Z_2, \dots, Z_m$ tayanch obyektlarga mantiqiy klassifikatorlar hosil bo'ladi va har bir mantiqiy klassifikator o'zining

Z tayanch obyektiga va unga o'xshash obyektlar to'plamiga(soniga) ega bo'ladi. Ushbu obyektlar to'plami alohida sinflarni tashkil qiladi.

Ushbu algoritm yordamida T_{nm} tanlovdan mantiqiy klassifikatorlarni EHM da izlash operatsiyalar soni T_{nm} tanlovdagi belgilar soniga nisbatan obyektlar soniga ko'proq bog'liq.

Misol. Masalaning qo'yilishi.

Bizga $\hat{O}_{5,5}$ (2.1-jadval) ko'rinishda obyektlar to'plami va $\hat{O}_{5,2}$ (2.2-jadval) ko'rinishdagi nazorat tanlovlar berilgan.

2.1-jadval

Obyektlar	Belgilar				
	x_1	x_2	x_3	x_4	x_5
S_1	37	1	1	0	1
S_2	36	0	1	0	0
S_3	38	0	0	1	0
S_4	37	1	0	0	1
S_5	38	1	0	1	0

3.2-jadval

Obyektlar	Belgilar				
	x_1	x_2	x_3	x_4	x_5
X_1	37	0	1	1	1
X_2	38	1	1	1	0

Talab etiladi: $\hat{O}_{5,5}$ (2.1-jadval) ko'rinishda obyektlar to'plamini sinflarga ajratuvchi mantiqiy klassifikatorlarni toping va ular asosida $\hat{O}_{5,2}$ sinov tanlovdagi obyektlarning hosil bo'lgan sinflarning qaysi biriga qarashlilikini aniqlang.

Masalaning yechilishi. $\hat{O}_{5,5}$ tanlovdan olingan tayanch obyektlarni boshqa obyektlar bilan taqqoslashda chiziqli yoki porogli moslik qoidalaridan foydalanamiz.

Chiziqli moslik qoida sifatida

$$d(S_i, S_j) = \begin{cases} 1, & \text{agar } x_{ki} = x_{kj}, k = 1, n \\ 0, & \text{aks holda.} \end{cases}$$

va porogli moslik qoida sifatida

$$d(S_i, S_j) = \begin{cases} 1, & \text{agar } |x_{ki} - x_{kj}| \leq \varepsilon_k, k = 1, n \\ 0, & \text{aks holda.} \end{cases}$$

foydalaniladi. Bu yerda $\varepsilon_k - u_k$, $k = \overline{1, n}$ belgilar alfavitida berilgan porog.

1. S_1, S_2, S_3, S_4, S_5 obyektlarni $\hat{O}_{5,5}$ tanlovdan olamiz va tartiblaymiz, ya'ni S_1, S_2, S_3, S_4, S_5
2. $Z = S_1$ ni tayanch obyekt sifatida tanlaymiz.
3. $b^0 = 11111$ boshqaruvchi vektorni tanlaymiz va ushbu vektor bo'yicha tekshiramiz.

	x_1	x_2	x_3	x_4	x_5
ε_k	2	0	0	0	0
b^0	1	1	1	1	1
$Z=S_1$	37	1	1	0	1
S_2	36	0	1	0	0
b^1	1	0	1	1	0

4. $b^1 = 10110$ boshqaruvchi vektor bo'yicha tekshiramiz

	x_1	x_3	x_4
ε_k	2	0	0
b^1	1	1	1
$Z=S_1$	37	1	0
S_3	38	0	1
b^2	1	0	0

6. $b^2 = 10000$ boshqaruvchi vektor bo'yicha tekshiramiz

	x_1
$\mathcal{E}_k$	2
b^2	1
$Z=S_1$	37
S_4	37
b^3	1

7. $b^3 = 10000$ boshqaruvchi vektor bo'yicha tekshiramiz

	x_1
$\mathcal{E}_k$	2
b^3	1
$Z=S_1$	37
S_5	38
b^4	1

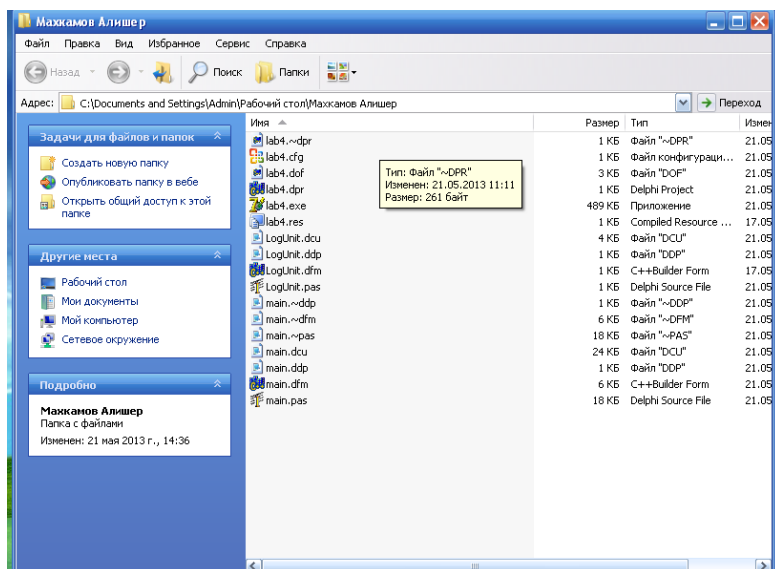
Demak $\dot{O}_{5,5}$ tanlovda $Z = S_1$ tayanch obyektga x_1 belgisi bo'yicha S_1, S_2, S_3, S_4, S_5 obyektlar o'xshash. Demak $Z = S_1$ tayanch obyekt orqali bitta sinf hosil bo'ldi. Huddi shunday boshqa sinflar hosil qilinadi.

3-BOB. DASTURIY TA'MINOT VA UNDAN FOYDALANISH TARTIBI

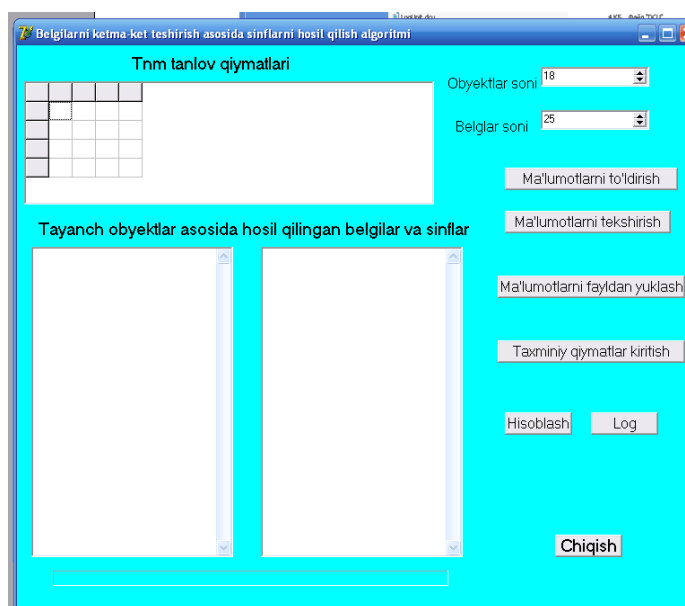
3.1. Dasturiy ta'minotning tasnifi

 **Программа Маматкулов Ойбек** - tugmachasini bosish bilan dasturga kiriladi.

Ish oynasida quyidagi oyna hosil bo'ladi.

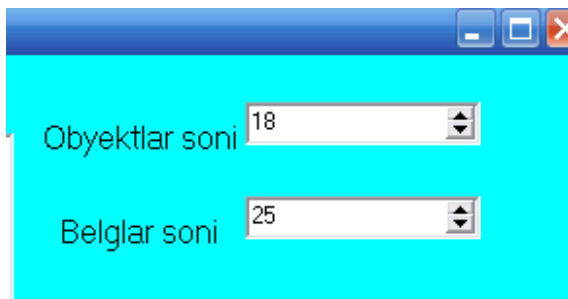


 **lab4.exe** - tugmachasini bosish bilan dastur ishga tushiriladi. Ish oynasida quyidagi oyna hosil bo'ladi.

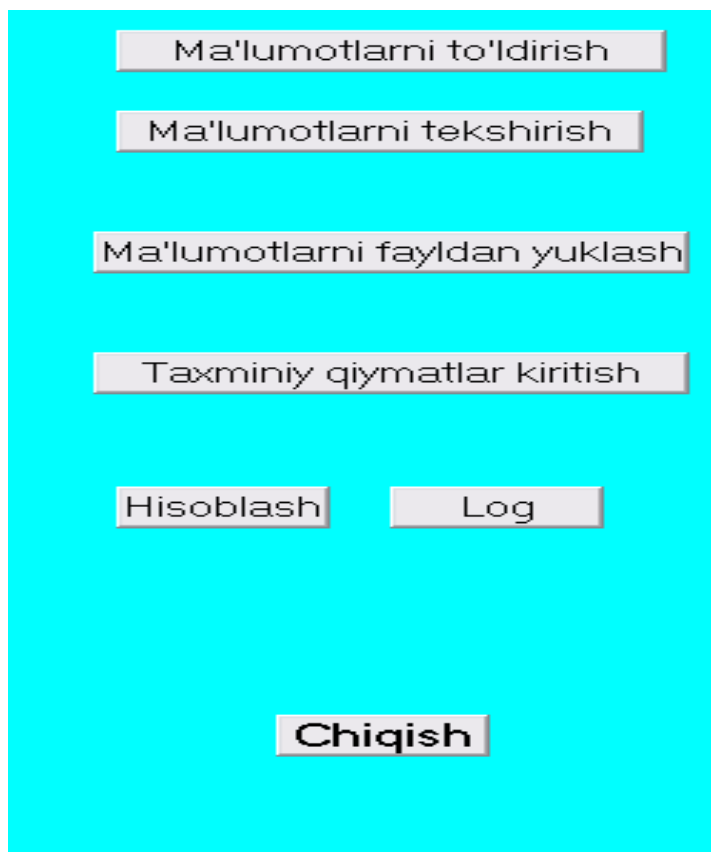


Ushbu oyna quidagi bo'limlardan iborat:

1. Ma'lumotlarni kiritish bo'limi



2. Boshqaruvch tugmachalar bo'limi:



- **Ma'lumotlarni to'ldirish** - T_{nm} tanlovdagi obyektlar va ularning belgilari qiymatini kiritish uchun ishlatiladi;
- **Ma'lumotlarni tekshirish** - T_{nm} tanlovdagi obyektlar va ularning belgilari qiymatlarini to'g'ri kiritilganligini tekshiradi;

- **Ma'lumotlarni fayldan yuklash** - T_{nm} tanlovdagi obyektlar va ularning belgilari qiymatlarini fayldan kiritish uchun ishlatiladi;
- **Taxminiy qiymatlar kiritish** - T_{nm} tanlovdagi obyektlar va ularning belgilari qiymatlarini taxminiy kiritish uchun ishlatiladi;
- **Hisoblash** - dasturni ishga tushiradi;
- **Log** - dasturning ishlash jarayoni va natijalarni ko'rsatadi;
- **Chiqish** - dasturdan chiqish uchun ishlatiladi;

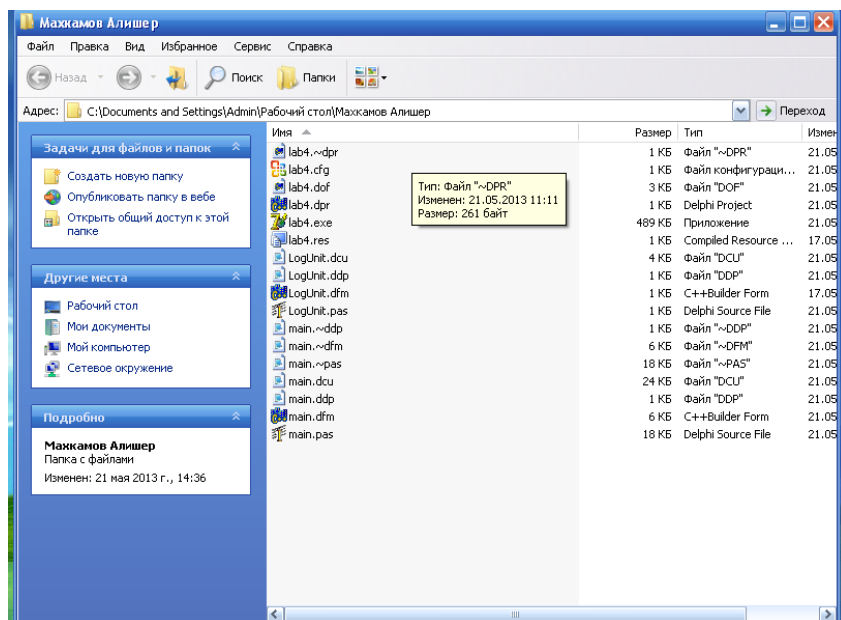
3. T_{nm} tanlovdagi obyektlar va ularning belgilari qiymatlarini kiritish bo'limi

4. T_{nm} tanlovdan tayanch obyektlar asosida hosil qilingan belgilar tizimini va ular yordamida hosil qilingan sinflarni ko'rsatish bo'limi

3.2. Dasturiy vositadan foydalanish tartibi

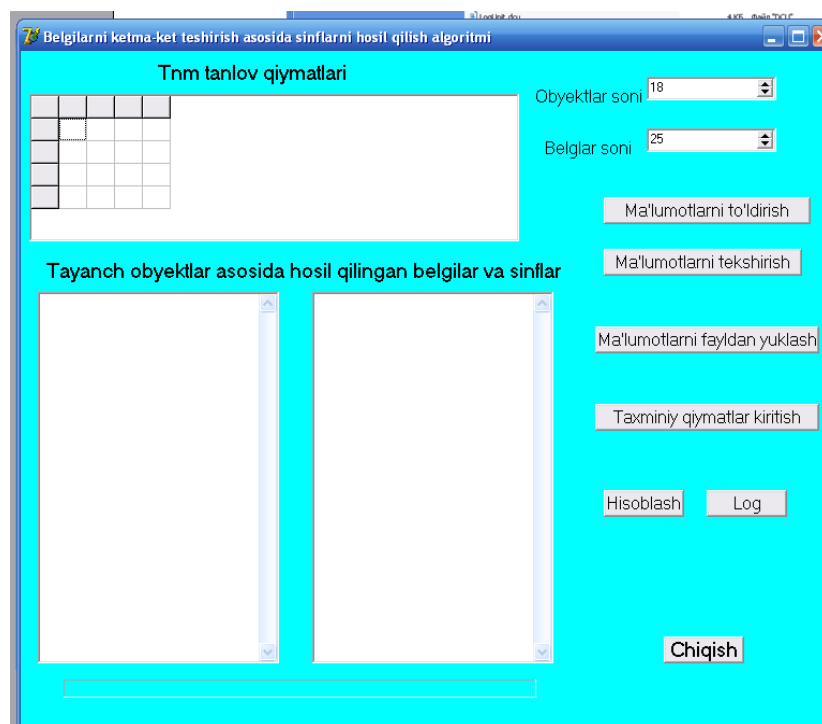
 Программа Маматкулов Ойбек - tugmachasini bosish bilan dasturga kiriladi.

Ish oynasida quyidagi oyna hosil bo'ladi.



 lab4.exe - tugmachasini bosish bilan dastur ishga tushiriladi. Ish

oynasida quyidagi oyna hosil bo'ladi.



Ushbu oynadagi ma'lumotlarni kiritish bo'limiga

Obyektlar soni

Belglar soni

qiymatlar kiritiladi va - tugmachasi bosiladi.

Ekranda

Belgilarni ketma-ket teshirish asosida sinflarni hosil qilish algoritmi

Tnm tanlov qiymatlari

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
S1	21	49	23	7	36	59	40	4	40	53
S2	53	47	8	44	32	49	1	18	59	53
S3	54	0	58	0	37	20	44	14	41	22
S4	58	26	16	30	5	0	4	23	10	59
S5	22	10	42	57	40	55	23	11	8	8

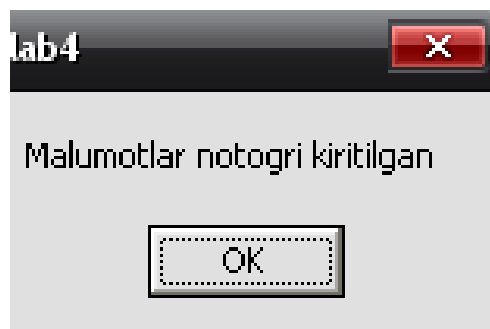
Obyektlar soni: 12
Belglar soni: 10

Ma'lumotlarni to'ldirish
Ma'lumotlarni tekshirish
Ma'lumotlarni fayldan yuklash
Taxminiy qiymatlar kiritish
Hisoblash Log
Chiqish

Tayanch obyektlar asosida hosil qilingan belgilar va sinflar

hosil bo'ladi.

Ushbu oyanada **Ma'lumotlarni tekshirish** - tugmachasi yordamida T_{nm} tanlovdagi kiritilgan qiymatlarining to'gri kiritilganligi tekshiriladi. Agar ma'lumotlar noto'gri kiritilgan bo'lsa, u holda oyanada



ma'lumot chiqadi.

- **Ma'lumotlarni fayldan yuklash** - tugmachasi yordamida T_{nm} tanlovdagi obyektlar va ularning belgilari qiymatlarini fayldan kiritiladi;
- **Taxminiy qiymatlar kiritish** - tugmachasi yordamida T_{nm} tanlovdagi obyektlar va ularning belgilari qiymatlarini taxminiy kiritiladi;

- **Hisoblash** - tugmachasini ishga tushirish bilan dastur ishlaydi va «Tayanch obyektlar asosida hosil qilingan belgilar va sinflar» bo'limida belgilar tizimi va sinflar hosil bo'ladi

Belgilarni ketma-ket tashirish asosida sinflarni hosil qilish algoritmi

Tnm tanlov qiymatlari

	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
S1	21	49	23	7	36	59	40	4	40	53
S2	53	47	8	44	32	49	1	18	59	53
S3	54	0	58	0	37	20	44	14	41	22
S4	58	26	16	30	5	0	4	23	10	59
S5	22	10	42	57	40	55	23	11	8	8

Obyektlar soni

Belglar soni

Ma'lumotlarni to'ldirish

Ma'lumotlarni tekshirish

Ma'lumotlarni faydan yuklash

Taxminiy qiymatlar kiritish

Tayanch obyektlar asosida hosil qilingan belgilar va sinflar

D(S1) = x6 ∨ x2
D(S2) = x8 ∨ x4 ∨ x5
D(S3) = x8 ∨ x7 ∨ x1 ∨ x6
D(S4) = x9^x10
D(S5) = x9^x10
D(S6) = x7 ∨ x10 ∨ x3 ∨ x2
D(S7) = x9^x10
D(S8) = x1 ∨ x10
D(S9) = x9^x10
D(S10) = x1
D(S11) = x4 ∨ x3
D(S12) = x5 ∨ x1

Tayanch Si obyekt asosida sinflarni hosil qilamiz
K(1) = {S1 S3 }
K(2) = {S1 S6 }

Tayanch Si obyekt asosida sinflarni hosil qilamiz
K(1) = {S2 S3 }
K(2) = {S2 S11 }
K(3) = {S2 S12 }

Tayanch Si obyekt asosida sinflarni hosil qilamiz
K(1) = {S2 S3 }
K(2) = {S3 S6 }
K(3) = {S3 S8 }
K(4) = {S1 S3 }

Tayanch Si obyekt asosida sinflarni hosil qilamiz
K(1) = {S4 }

Hisoblash

Log

Chiqish

- **Log** - tugmachasini ishga tushirish bilan dasturning ishlash jarayoni va natijalarni ko'rish mumkin

```

7* Log

b = 0000000000
Z* = S1 = 11 4 37 38 23 17 11 20 6 54
      S7 = 56 22 25 47 45 45 11 31 34 68
      -----
b = 0000000000
Z* = S1 = 11 4 37 38 23 17 11 20 6 54
      S8 = 1 40 79 79 63 57 51 70 23 60
      -----
b = 0000000000
Z* = S1 = 11 4 37 38 23 17 11 20 6 54
      S9 = 30 14 67 29 48 29 4 77 78 2
      -----
b = 0000000000
Z* = S1 = 11 4 37 38 23 17 11 20 6 54
      S10 = 57 46 10 7 42 26 41 35 57 76
      -----
b = 0000000000
Z* = S1 = 11 4 37 38 23 17 11 20 6 54
      S11 = 42 25 64 34 39 61 9 18 29 19
      -----
b = 0000000000
Z* = S1 = 11 4 37 38 23 17 11 20 6 54
      S12 = 57 72 1 68 9 27 11 63 8 47
      -----
b = 0000000000
D(S1) = x6 v x2
Hosil qilingan sinflar
K(1) = {S1 S3}
K(2) = {S1 S6}
D(S1) = x6 v x2 (intervallar usuli bilan qisqartirilgandan so'ng)
Tayanch obyekt: S2
Solishtiriladigan obyektlar: S3 S4 S5 S6 S7 S8 S9 S10 S11 S12 S1
b = 1111111111
Z* = S2 = 2 18 50 34 9 58 28 15 17 3
      S3 = 1 39 0 12 4 17 49 15 39 23
      -----
b = 0000000100 +
Z* = S2 = 2 18 50 34 9 58 28 15 17 3
      S4 = 40 33 3 10 12 34 14 6 43 25
      -----
b = 0000000000
Z* = S2 = 2 18 50 34 9 58 28 15 17 3

```

3.3. Komp'yuter bilan ishlashda texnika xavfsizligi qoidalar va talablari

Komp'yuter bilan ishlash davomida bir qator faoliyat bilan bog'liq fiziologik jarayonlar sodir bo'ladi va inson organizmiga turli darajada ta'sir ko'rsatadi.

Bu kabi surunkali ta'sirlar va xavfli holatlar muxandis-dasturlarni ish faoliyatiga ta'sir ko'rsatib kasb kasalliklarini keltirib chiqarishi tasdiqlangan.

Yong'in xavfsizligi qoidalariga binoan EHM joylashtirilgan xonalar, ko'paytirish qiyofa ko'chirish uskunalar atrofida tutunli xabar berish moslamalari o'rnatiladi. Yonqin boshlangan holatda plastmassalarni tutashi natijasida zaharli is gazi ajralib

chiqadi va ishchilarni bug'ishi mumkin. Shu sababli, belgilangan talablarga mos keladigan o'chirish usullari va nazariyasi- dan foydalanish maqsadidga muvofiqdir. Olib borilayotgan tadqiqotchilar va tahlillar asosida aynan shu holatga mos keluvchi usullar tanlanadi. Ish o'rnini samarali tashkil yetishda asosiy yе'tibor elektromagnit va elektrostatik maydonlar ta'sirini kamaytirishga qaratilishi darkor. Bugungi kunda ish o'rnini yergonomik talablar asosida tashkil etish, komfort ish sharoitini tashkil etish dolzarb sosial masaladir. Olingan tadqiqotlar asosida ish stolining optimal o'lchamlari, qo'lay konfigurasiya, ish o'rnini to'g'ri tashkil etishda asosiy manbadir.

Komp'yuterlar joylashtirilgan xonalar va dasturlovchi o'rni yong'in xavfsizligi talablari bo'yicha «D» toifaga mansub bo'lib, tez yonuvchi materiallar kam hisoblanadi.

Elektr hisoblash mashinalari va aloqa vositalari joylashtirilgan xonalarda yong'in xavfi mavjud bo'lib, bu holat katta miqdorda tez eriydigan yengil materiallar. Bu xonalarda yong'in asosan yonuvchan moddalarning oksidlanishi, issiqlik ta'sirida erishi natijasida sodir bo'ladi. Mavjud xonalarda o'rnatilgan akustik va estetik talablarga mos keluvchi eshik, pol, kabellar izolyasiyasi va boshqalar yonuvchi komponentlar toifasiga kiradi.

Yong'inga qarshi kurash bu kompleks tashkiliy va texnik tadbirlar bo'lib, xodimlarni xavfsizligini ta'minlash, yong'inni samarali o'chirish uni tez tarqalishini oldini olishdan iborat. Uchqun chiqaruvchi manbalar esa, elektron sxemalar, elektr ta'minot tarmoqlari, turli elektron uskunalarning nosozligi, qisqa tutashuv, yer bilan notug'ri biriktirish, ulashdagi nosozliklar hisoblanadi.

Zamonaviy komp'yuterlar va EHMLarida elektron sxemalar va detallar zich joylashganligi va kabellar hamda ulash simlari yaqinligi sababli bu xavf ancha yuqoridir. Bulardan elektr tokini oqishi himoya qavatini erishiga olib keladi. Komp'yuterlarda bu kabi ortiqcha issiqlik maxsus shamolatish bloki orqali chiqarib yuboriladi. Bu blokni ishdan chiqishi yesa juda xavflidir. Bunday hollarda xavfsizlik

choralarini ko'rish va qo'shimcha shamollatish tizimini tashkil yetish maqsadga muvofiqdir.

Ma'lumki, yong'in holatini komp'yuter choralarda qo'llanilgan qurilish materiallarining yong'inga chidamligini, umumiy mustahkamlikni ta'minlaydi. Elektron uskunalarni yuqori narxini e'tiborga olgan holda bu turdagi xonalarni 1 yoki 2 toifali yong'inga chidamlilik deb darajalash mumkin.

Kichik hajmdagi yong'in o'choqlarini bartaraf etish uchun binolarda yong'in krani, o't o'chirish moslamasi, quruq qo'm, asbestli ko'rpa va boshqalarda foydalaniladi.

Belgilangan talablarga javob beradigan sharoit yaratish, ish sifatini va unumdorligini oshirish, dastur ishini yaxshilash, nuqsonsiz ishlashni ta'minlash imkonini beradi.

Komp'yuter bilan ishlash davomida insonlarning ko'rish organlarida yuqori darajada kuchlanish holati sodir bo'ladi.

Turli o'lchamdagi va hajmdagi grafik materiallar, kichik o'lchamlar, hajmini ortishi natijasida ko'zda surunkali jiddiy kuchlanish yoki zo'riqish sodir bo'ladi. Bu kabi salbiy holatlar ko'zni ko'rish qobiliyatini pasayishiga olib keladi.

Belgilangan me'yorlarga binoan foydalanuvchi operator ko'zi bilan yekran oralig'idagi masofa 600-700 mm bo'lishi lozim, Ayrim hollarda harf va raqamlar kattaligini e'tiborga olgan holda bu masafani 500 mm bo'lishiga ruxsat beriladi.

Ko'zni zo'riqishini kamaytirish maqsadida har 1 soatda 15 minutlik tanaffus qilish tavsiya etiladi.

Bugungi kunda bir qator olimlar tomonidan komp'yuter monitorlaridan tushayotgan turli nurlarni inson organizmiga ta'siri va ma'lum ionlanish sodir bo'lishi tasdiqdangan. Lekin, bu masalada to'laqonli belgilar mavjud emas.

Elektromagnit ionlashgan nurlarning monitordan tushadigan miqdori qo'yidagi jadvalda keltirilgan.

Ma'lumki, tegishli me'yorlarga asosan ish joyida komp'yuter monitoridan ajraladigan maksimal rengen nurlarning maksimal miqdori 10 mk ber/ch va ultrabinafsha va infraqizil nurlarning bu kabi miqdori yesa 10-100 mVt/m² bo'lishi aniqlangan.

XULOSA VA TAKLIFLAR

Bitiruv malakaviy ishida qaralgan masalalardan quyidagi natijalar va xulosalarga kelamiz:

1. Mantiqiy klassifikatorlar asosida qiyofalarni anglovchi tizimning asosiy tushunchalari va ta'riflari o'rganildi.

2. Qiyofalarni anglovchi tizimlarning turlari va obyektlar to'plamini yadro obyektlar yordamida sinflashtirish usullari o'rganildi va taxlil qilindi.

3. Mantiqiy klassifikatorlar asosida obyektlar to'plamini sinflashtirish masalasi formallashtirildi va ishning maqsadi aniqlandi.

4. Obyektlar to'plamini mantiqiy klassifikatorlar asosida sinflashtirish algoritmi ishlab chiqildi.

5. Yaratilgan algoritm asosida Delfi dasturlash tilida dasturiy ta'minot yaratildi.

6. Dasturiy ta'minot mantiqiy klassifikatorlar asosida obyektlar to'plamini sinflarga ajratish masalalariga doir misollarda sinovdvn o'tkazildi va olingan natigalar Del'fi tilida tuzilgan dasturning to'g'ri ishlashini tasdiqladi.

7. Dasturiy ta'minotni tibbiyot, texnika, geologiya, arxeologiya, qishloq xo'jaligi va boshqa sohalarda obyektlar to'plami ko'rinishda berilgan masalalarda ularning belgilariga qarab sinflarga ajratishda qo'llash mumkin

Foydalanilgan adabiyotlar

1. Ўзбекистон Республикаси Президентининг 2012 йил 21 мартдаги «Замонавий ахборот-коммуникация технологияларини янада жорий этиш ва ривожлантириш чора-тадбирлари тўғрисида» ПҚ-1730 сонли қарори.
2. Абдукаримов Р.Т. Алгоритмы распознавания, основанные на поиске признаков классов. - В. Сб.: Вопросы кибернетики, вы.88.Ташкент, 1976.
3. Журавлев Ю.И. Об алгоритмическом подходе к решению задач распознавания или классификации. - Пробл.кибернетики, вып.33, 1978. с.5-68.
4. В.И. Васильев. Распознающие системы. Киев.:Наукова Думка.1986.-415 с.
5. Васильев В.И. Проблема обучения распознаванию образов. Принципы, алгоритмы, реализация. - Киев: Высшая школа, 1989. - 64 с.
6. Вапник В.Н.,Червоненкис А.Я. Теория распознавания образов.- М.,Наука,1974,415 с.
7. Горелик А.А.,Скрипник В.А. Методы распознавания.- М.,Высшая школа,1977. 222 с.
8. Фор А. Восприятия распознавания образов. М.1989. 271 с.
9. Гаврилов А.В. Системы искусственного интеллекта. Ч.1. - Учебное пособие, Новосибирск: НГТУ, 2000
10. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. - М.: Горячая линия-Телеком, 2001.
11. Горбан А.Н., Россиев Д.А. Нейронные сети на персональном компьютере. – Новосибирск, Наука, 1996.
12. Арлазаров В.Л., Астахов А.Д., Троянker В.В., Котович Н. В. Адаптивное распознавание символов. В сб. "Интеллектуальные технологии ввода и обработки информации", М.: Эдиториал УРСС, 1998.

13. А.В.Мисюрёв. Использование искусственных нейронных сетей для распознавания рукопечатных символов. В сб. "Интеллектуальные технологии ввода и обработки информации", М.: Эдиториал УРСС, 1998.

14. В.В. Постников. Разработка методов наложения формы на графическое изображение документа. В сб. "Интеллектуальные технологии ввода и обработки информации", М.: Эдиториал УРСС, 1998

15. Бекмуратов К.А. Задача формирования логических классификаторов, обеспечивающих требуемую надежность распознавания. Меж. науч. конф. Инновация –2005. 20-21 октября 2005.

16. Бекмуродов Қ.А. Қисман прецедентли принципга асосланган танувчи тизимни яратиш алгоритми ва дастурий таъминоти. Математика, математик моделлаштириш ва ахборот технологияларининг долзарб масалалари. Рес.илмий конф. материаллари тўплами. Термиз. 21-22 ноябрь. 2012 й. 111-114 б.

17. Бекмуратов Д.Қ., Мамарауфов О. Синфларни оптимал қоплаш алгоритми ва дастурий воситаси. Математика, математик моделлаштириш ва ахборот технологияларининг долзарб масалалари. Рес.илмий конф. материаллари тўплами. Термиз. 21-22 ноябрь. 2012 й. 195-198 б.

18. Bekmurodov Q.A. Axborot - tanuvchi tizimlar: Ma'ruzalar kursi(to'ldirilgan qayta nashri). - Samarqand: TATU Samarqand filiali, 2012. - 159 bet.

19. Bekmurodov Q.A., Mamaraufov O.A., Bekmurodov D.Q. Axborot-tanuvchi tizimlar fanidan amaliy va laboratoriya mashg'ulotlar uchun uslubiy qo'llanma. - Samarqand: TATU Samarqand filiali, 2012. 162 bet.

20. Аҳолини ва ҳудудларни фавқулодда вазиятлардан ҳимоялаш, Ўқув қўлланма, ГСЧС, Тошкент, 2003.

21. Фукаро муҳофазаси меъёрлари ва қоидалари СНИП ИТМ ГЗ – 93, 1993.

1 – ilova. Oynalarni hosil qilish kodi

```
object Form1: TForm1
  Left = 155
  Top = 198
  Width = 769
  Height = 674
  Caption =
    'Belgilarni ketma-ket teshirish asosida sinflarni hosil qilish al' +
    'goritmi'
  Color = clAqua
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -11
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  OldCreateOrder = False
  OnCreate = FormCreate
  PixelsPerInch = 96
  TextHeight = 13
  object Label1: TLabel
    Left = 480
    Top = 32
    Width = 104
    Height = 20
    Caption = 'Obyektlar soni '
    Font.Charset = DEFAULT_CHARSET
    Font.Color = clWindowText
    Font.Height = -16
    Font.Name = 'MS Sans Serif'
    Font.Style = []
    ParentFont = False
  end
  object Label2: TLabel
    Left = 488
    Top = 80
    Width = 82
    Height = 20
    Caption = 'Belglar soni'
    Font.Charset = DEFAULT_CHARSET
    Font.Color = clWindowText
    Font.Height = -16
    Font.Name = 'MS Sans Serif'
    Font.Style = []
    ParentFont = False
  end
  object Label3: TLabel
```

```

Left = 128
Top = 8
Width = 176
Height = 24
Caption = 'Tnm tanlov qiymatlari'
Font.Charset = DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -19
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
object Label4: TLabel
Left = 24
Top = 192
Width = 485
Height = 24
Caption = 'Tayanch obyektlar asosida hosil qilingan belgilar va sinflar '
Font.Charset = DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -19
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
end
object StringGrid1: TStringGrid
Left = 8
Top = 40
Width = 457
Height = 137
DefaultColWidth = 25
DefaultRowHeight = 20
Options = [goFixedVertLine, goFixedHorzLine, goVertLine, goHorzLine, goRangeSelect,
goEditing]
TabOrder = 0
end
object SpinEdit1: TSpinEdit
Left = 584
Top = 24
Width = 121
Height = 22
MaxValue = 0
MinValue = 0
TabOrder = 1
Value = 18
OnChange = SpinEdit1Change

```

```

end
object SpinEdit2: TSpinEdit
  Left = 584
  Top = 72
  Width = 121
  Height = 22
  MaxValue = 0
  MinValue = 0
  TabOrder = 2
  Value = 25
  OnChange = SpinEdit2Change
end
object Button1: TButton
  Left = 536
  Top = 256
  Width = 209
  Height = 25
  Caption = 'Ma'#39'lumotlarni fayldan yuklash'
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -16
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  ParentFont = False
  TabOrder = 3
  OnClick = Button1Click
end
object Button2: TButton
  Left = 544
  Top = 408
  Width = 75
  Height = 25
  Caption = 'Hisoblash'
  Font.Charset = DEFAULT_CHARSET
  Font.Color = clWindowText
  Font.Height = -16
  Font.Name = 'MS Sans Serif'
  Font.Style = []
  ParentFont = False
  TabOrder = 4
  OnClick = Button2Click
end
object Button3: TButton
  Left = 640
  Top = 408
  Width = 75

```

```

Height = 25
Caption = 'Log'
Font.Charset = DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -16
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 5
OnClick = Button3Click
end
object Button4: TButton
Left = 536
Top = 328
Width = 209
Height = 25
Caption = 'Taxminiy qiymatlar kiritish'
Font.Charset = DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -16
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 6
OnClick = Button4Click
end
object Memo1: TMemo
Left = 16
Top = 224
Width = 225
Height = 345
Lines.Strings = (
    ")
ScrollBars = ssVertical
TabOrder = 7
end
object ProgressBar1: TProgressBar
Left = 40
Top = 584
Width = 441
Height = 17
TabOrder = 8
end
object Button5: TButton
Left = 544
Top = 136

```

```

Width = 193
Height = 25
Caption = 'Ma'#39'lumotlarni to'#39'ldirish'
Font.Charset = DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -16
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 9
OnClick = Button5Click
end
object Button6: TButton
Left = 600
Top = 544
Width = 75
Height = 25
Caption = 'Chiqish'
Font.Charset = DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -19
Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 10
OnClick = Button6Click
end
object Memo2: TMemo
Left = 272
Top = 224
Width = 225
Height = 345
Lines.Strings = (
  ")
ScrollBars = ssVertical
TabOrder = 11
end
object Button7: TButton
Left = 544
Top = 184
Width = 185
Height = 25
Caption = 'Ma'#39'lumotlarni tekshirish'
Font.Charset = DEFAULT_CHARSET
Font.Color = clWindowText
Font.Height = -16

```

```

Font.Name = 'MS Sans Serif'
Font.Style = []
ParentFont = False
TabOrder = 12
OnClick = Button7Click
end
end

```

2 – ilova. Dasturning kodi

```

unit main;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Grids, StdCtrls, Spin,Mask,ComCtrls;
type
  TMask = array of byte;
  TMaskList = array of TMask;
  TObyekt = array of byte;
  TKlass = array of TObyekt;
  TKlasslar = array of TKlass;
  TList = array of integer;
  TCheckList = array of TList;
  TArrayUnit = record
    index: integer;
    count: integer;
    objectset: set of 0..255;
  end;
  TForm1 = class(TForm)
    StringGrid1: TStringGrid;
    SpinEdit1: TSpinEdit;
    SpinEdit2: TSpinEdit;
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    Button4: TButton;
    Label1: TLabel;
    Label2: TLabel;
    Memo1: TMemo;
    ProgressBar1: TProgressBar;
    Button5: TButton;
    Label3: TLabel;
    Label4: TLabel;
    Button6: TButton;
    Memo2: TMemo;
    Button7: TButton;
    procedure Button1Click(Sender: TObject);
  end;

```

```

    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure SpinEdit1Change(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure SpinEdit2Change(Sender: TObject);
    procedure SpinEdit3Change(Sender: TObject);
    procedure Button6Click(Sender: TObject);
    procedure Button7Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
    colortable: array [1..5] of integer;
    SuperGlobalKlassList: TKlasslar;
    SuperGlobalUnknownObjects: TKlass;
end;
var
    Form1: TForm1;
    n, m, w: integer;
    log: string;
    log5:string;
    a: TKlasslar;
    unknown: TObyekt;
implementation
uses LogUnit,Math;
procedure AddToLog(const s: string; line: boolean = true); overload;
begin
    log := log + s;
    if line then
        log := log + #13#10;
end;
procedure AddToLog(const s: string; const List: TList; line: boolean = true); overload;
var
    i: integer;
begin
    log := log + s;
    for i := 0 to Length(List) - 1do
        log := log + 'S' + IntToStr(List[i] + 1) + #32;
    if line then
        log := log + #13#10;
end;
procedure AddToLog(const s: string; const List: TMask; line: boolean = true); overload;
var
    i: integer;

```

```

begin
  log := log + s;
  for i := 0 to Length(List) - 1 do
    log := log + IntToStr(List[i]) + #32;
  if line then
    log := log + #13#10;
  end;
procedure AddToLog(const s: string; const List: TObyekt; line: boolean = true); overload;
var
  i: integer;
begin
  log := log + s;
  for i := 0 to Length(List) - 1 do
    log := log + IntToStr(List[i]) + #32;
  if line then
    log := log + #13#10;
  end;
procedure AddToLog5(const s: string; line: boolean = true); overload;
begin
  log5 := log5 + s;
  if line then
    log5 := log5 + #13#10;
end;
{$R *.dfm}
procedure TForm1.Button1Click(Sender: TObject);
var
  x: textfile;
  b: byte;
  i,k,z: integer;
begin
  AssignFile(x, 'input.txt');
  Reset(x);
  Readln(x, w, n, m);
  SpinEdit2.Value := m;
  SpinEdit1.Value := w;
  SpinEdit1.Value := n;
  //DrawFixedRowsCols;
  for z := 0 to w - 1 do
    begin
      for i := 0 to n - 1 do
        begin
          for k := 0 to m - 1 do
            begin
              Read(x, b);
              StringGrid1.Cells[k + 1, z * n + i + 1] := chr(ord('0')+b);
            end;
          end;
        end;
      end;
    end;
  end;

```

```

    end;
end;
for k := 1 to m do
begin
    Read(x, b);
    // StringGrid2.Cells[k, 1] := chr(ord('0')+b);
end;
CloseFile(x);
end;
function Hisoblash(index: integer; Klass: integer; var c: integer): string;
var
    list: TList;
    i, k: integer;
    b: TMask;
    CheckList: TCheckList;
    GlobalMaskList: TMaskList;
    matches: boolean;    //1 2 3 5
                        //2 3 5 1
                        //3 5 1 2
                        //5 1 2 3
    procedure CreateCheckList(var checklist: TCheckList; FirstList: TList); //obyektlar
perestanovkasini yasovchi protsedura
    var
        i,k: integer;
        temp: integer;
    begin
        for i := 1 to n - 1 do
        begin
            temp := FirstList[0];
            for k := 0 to (n - 1) - 1 - 1 do
                FirstList[k] := FirstList[k+1];
            FirstList[n-2] := temp;
            temp := Length(checklist) + 1;
            SetLength(checklist, temp);
            SetLength(checklist[temp - 1], n - 1);
            for k := 0 to n - 2 do
                checklist[temp - 1][k] := FirstList[k];
            end;
        end;
    end;
function isSimilar(second: integer; ClassToCompare: TKlass; Mask: TMask):boolean;
//obyektlarni Mask da ko'rstailgan belgilari bilan o'xshashligini aniqlovchi protsedura
var
    i: integer;
begin
    result := true;
    for i := 0 to m - 1 do

```

```

    if Mask[i] = 1 then
        if a[Klass, index, i] <> ClassToCompare[second, i] then
            begin
                result := false;
                break;
            end;
        end;
    end;
    procedure DoXor(second: integer; Mask: TMask); //obyektlarning Mask da ko'rsatilgan belgilari
    bilan solishtiri, uni yetarli va yetarli emasligini bilgan holga yangi Mask ni qaytaruvchi protsedura
    label
    break1;
var
    resultMask: TMask;
    canBeAssigned: boolean;
    i,k: integer;
begin
    SetLength(resultMask, m);
    for i := 0 to m - 1 do
        if Mask[i] = 1 then
            begin
                if a[Klass, index, i] = a[Klass, second, i] then
                    resultMask[i] := 1
                else
                    resultMask[i] := 0;
            end
        else
            resultMask[i] := 0;
            AddToLog(#9#9' b = ', resultMask, false);
            canBeAssigned := true;
        // for k := 0 to w - 1 do
        for k := 0 to 0 do
            if a[k] <> a[Klass] then
                for i := 0 to n - 1 do
                    begin
                        if isSimilar(i, a[k], resultMask) then
                            begin
                                canBeAssigned := false;
                                // AddToLog(' - (Boshqa klassdagi ' + IntToStr(i+1) + ' obyektga o"xshash)');
                                goto break1;
                            end;
                        end;
                    end;
                break1:
                if canBeAssigned then
                    begin
                        for i := 0 to m - 1 do
                            Mask[i] := resultMask[i];

```

```

    AddToLog(' ');
end;
SetLength(resultMask, 0); //memory cleaning...
end;
function MakeString(Mask: TMask):string; // Maskda ko'rsatilgan belgilardan string turidagi
konuyunktiv shalkni yasocho protsedura
var
    i: integer; //9.06.20
begin
    result := "";
    for i := 0 to m - 1 do
        if Mask[i] = 1 then
            begin
                if a[Klass, index, i] = 1 then
                    result := result + 'x' + IntToStr(i+1) + '^'
                else
                    result := result + 'x' + IntToStr(i+1) + '^'
                end;
            end;
        result := Copy(result, 1, Length(result) - 1);
    end;
    procedure Rang(Mask: TMask);
    var
        i,k,z: integer;
        canBeDropped: boolean;
    begin
        for i := 0 to m - 1 do
            begin
                if Mask[i] = 0 then
                    continue;
                Mask[i] := 0;
                CanBeDropped := true;
                for z := 0 to w - 1 do
                    if z <> Klass then
                        for k := 0 to n - 1 do
                            if isSimilar(k, a[z], Mask) then
                                begin
                                    CanBeDropped := false;
                                    break;
                                end;
                            end;
                        if not canBeDropped then
                            Mask[i] := 1;
                        end;
                    end;
                end;
            end;
        procedure Interval(var MaskList: TMaskList);
        var
            CurrentObjectSet: set of 1..255;

```

```

FullObjectSet: set of 1..255;
au: array of TArrayUnit;
temp: TArrayUnit;
i,k: integer;
sorted: boolean;
NewMaskList: TMaskList;
begin
  AddToLog(#13#10'Hosil qilingan sinflar');
  SetLength(au, Length(MaskList));
  FullObjectSet := [];
  for i := 0 to Length(MaskList) - 1 do
    begin
      au[i].index := i;
      au[i].objectset := [];
      AddToLog(#9'K(' + IntToStr(i+1) + ') = {' , false);
      for k := 0 to n - 1 do
        if isSimilar(k, a[Klass], MaskList[i]) then
          begin
            inc(au[i].count);
            include(au[i].objectset, k);
            AddToLog('S' + IntToStr(k+1) + ' ', false);
          end;
      AddToLog('}');
    end;
    for k := 0 to n - 1 do
      Include(FullObjectSet, k);
    repeat
      sorted := true;
      for i := 0 to (Length(au) - 1) - 1 do
        if au[i].count < au[i+1].count then
          begin
            temp := au[i];
            au[i] := au[i+1];
            au[i+1] := temp;
            sorted := false;
          end;
      until sorted;
      CurrentObjectSet := au[0].objectset;
      for i := 1 to Length(au) - 1 do
        begin
          if CurrentObjectSet = FullObjectSet then
            break;
          if au[i].objectset <= currentobjectset then
            SetLength(MaskList[au[i].index], 0)
          else
            CurrentObjectSet := CurrentObjectSet + au[i].objectset;
        end;
      end;
    end;
  end;
end;

```

```

end;
SetLength(au, 0); //memory cleaning...
for i := 0 to Length(MaskList) - 1 do
  if Length(MaskList[i]) <> 0 then
    begin
      SetLength(NewMaskList, Length(NewMaskList) + 1);
      SetLength(NewMaskList[Length(NewMaskList)-1], m);
      for k := 0 to m - 1 do
        NewMaskList[Length(NewMaskList)-1][k] := MaskList[i][k];
      SetLength(MaskList[i], 0);
    end;
  SetLength(MaskList, 0);
  MaskList := NewMaskList;
  AddToLog("");
end;
procedure Interval5(var MaskList: TMaskList);
var
  CurrentObjectSet: set of 1..255;
  FullObjectSet: set of 1..255;
  au: array of TArrayUnit;
  temp: TArrayUnit;
  i,k: integer;
  sorted: boolean;
  NewMaskList: TMaskList;
  ssss,f:integer;
begin
  AddToLog5(#13#10'Tayanch Si obyekt asosida sinflarni hosil qilamiz');
  SetLength(au, Length(MaskList));
  FullObjectSet := [];
  for i := 0 to Length(MaskList) - 1 do
    begin
      au[i].index := i;
      au[i].objectset := [];
      AddToLog5(#9'K(' + IntToStr(i+1) + ') = {' , false);
      for k := 0 to n - 1 do
        if isSimilar(k, a[Klass], MaskList[i]) then
          begin
            inc(au[i].count);
            include(au[i].objectset, k);
            AddToLog5('S' + IntToStr(k+1) + ' ', false);
          end;
      AddToLog5('}');
    end;
  for k := 0 to n - 1 do
    Include(FullObjectSet, k);
  repeat

```

```

sorted := true;
for i := 0 to (Length(au) - 1) - 1 do
  if au[i].count < au[i+1].count then
    begin
      temp := au[i];
      au[i] := au[i+1];
      au[i+1] := temp;
      sorted := false;
    end;
until sorted;
CurrentObjectSet := au[0].objectset;
for i := 1 to Length(au) - 1 do
  begin
    if CurrentObjectSet = FullObjectSet then
      break;
    if au[i].objectset <= currentobjectset then
      SetLength(MaskList[au[i].index], 0)
    else
      CurrentObjectSet := CurrentObjectSet + au[i].objectset;
  end;
SetLength(au, 0); //memory cleaning...
for i := 0 to Length(MaskList) - 1 do
  if Length(MaskList[i]) <> 0 then
    begin
      SetLength(NewMaskList, Length(NewMaskList) + 1);
      SetLength(NewMaskList[Length(NewMaskList)-1], m);
      for k := 0 to m - 1 do
        NewMaskList[Length(NewMaskList)-1][k] := MaskList[i][k];
      SetLength(MaskList[i], 0);
    end;
SetLength(MaskList, 0);
MaskList := NewMaskList;
AddToLog5("");
end;
function ExistsInGlobalMaskList(Mask: TMask): boolean;
var
  i, k: integer;
  x: boolean;
begin
  result := false; // faraz qilamiz Mask GlobalMaskListda mavjud emas deb
  for i := 0 to Length(GlobalMaskList) - 1 do //GlobalMaskList dagi har bir TMask bilan
    tekshirib ko'ramiz
    begin
      x := true; //faraz qilamiz o'xshash deb
      for k := 0 to m - 1 do
        if GlobalMaskList[i][k] <> Mask[k] then //agar bitta faqr uchrasa...

```

```

begin
  x := false; //demak farazimiz noto'g'ri, bu TMask o'xshash emas
  break;
end;
if x then //agar faramiz to'g'ri bo'lib chiqsa, ya'ni o'xsha TMask topisa
begin
  result := true; //demak Mask GlobalMaskList da mavjud ekan, birinchi farazimiz noto'g'ri
ekan
  exit;
end;
// keyingi TMask bilan tekshiramiz
end;
//barcha TMasklar bilan tekshirdik, bironta ham o'xshash topilmadi, demak Mask
GlobalMaskList da mavjud emas degan farazimi to'g'ri ekan
end;
begin
  SetLength(GlobalMaskList, 0);
  AddToLog('Tayanch obyekt: S' + IntToStr(index+1));
  AddToLog("");
  SetLength(list, n - 1);
  k := 0;
  for i := 0 to n - 2 do
  begin
    if k = index then
      inc(k);
      list[i] := k;
      inc(k);
    end;
    CreateCheckList(CheckList, list);
    SetLength(list, 0); //memory cleaning...
    SetLength(b, m);
    for i := 0 to Length(CheckList) - 1 do
    begin
      AddToLog(#9'Solishtiriladigan obyektlar: ', CheckList[i]);
      AddToLog("");
      for k := 0 to m - 1 do
        b[k] := 1;
        AddToLog(#9#9' b = ', b);
        AddToLog("");
        for k := 0 to Length(CheckList[i]) - 1 do
        begin
          AddToLog(#9'Z* = S' + IntToStr(index+1) + ' = ', a[Klass][index]);
          AddToLog(#9#9'S'+IntToStr(checkList[i][k]+1) + ' = ', a[Klass][checkList[i][k]]);
          AddToLog(#9#9' -----', true);
          DoXor(checkList[i][k], b);
          AddToLog("");

```

```

end;
SetLength(CheckList[i], 0); //memory cleaning...
Rang(b);
if not ExistsInGlobalMaskList(b) then
begin
SetLength(GlobalMaskList, Length(GlobalMaskList) + 1);
SetLength(GlobalMaskList[Length(GlobalMaskList)-1], m);
for k := 0 to m - 1 do
GlobalMaskList[Length(GlobalMaskList)-1][k] := b[k];
end;
end;
SetLength(b, 0); //memory cleaning...
SetLength(CheckList, 0);
result := "";
for i := 0 to Length(GlobalMaskList) - 1 do
result := result + MakeString(GlobalMaskList[i]) + ' v ';
result := Copy(result, 1, Length(result) - 3);
AddToLog('D(S' + IntToStr(index+1) + ') = ' + result);
Interval(GlobalMaskList);
Interval5(GlobalMaskList);
result := "";
for i := 0 to Length(GlobalMaskList) - 1 do
begin
result := result + MakeString(GlobalMaskList[i]) + ' v ';
end;
matches := false;
for i := 0 to Length(GlobalMaskList) - 1 do
begin
matches := true;
for k := 0 to m - 1 do
if GlobalMaskList[i][k] = 1 then
if unknown[k] <> a[Klass][index][k] then
begin
matches := false;
break;
end;
SetLength(GlobalMaskList[i], 0); // memory cleaning...
if matches then
break;
end;
if matches then
inc(c);
SetLength(GlobalMaskList, 0); //memory cleaning...
result := Copy(result, 1, Length(result) - 3);
AddToLog('D(S' + IntToStr(index+1) + ') = ' + result + ' (intervallar usuli bilan
qisqartirilgandan so"ng)'#13#10);

```

```

end;
procedure TForm1.Button2Click(Sender: TObject);
var
  i, k, z: integer;
  x: TDateTime;
  uxshashliklar: array of integer;
begin
  x := Now;
  Log := "";
  TControl(Sender).Enabled := false;
  ProgressBar1.Visible := true;
  //Label6.Caption := 'Progress';
  ProgressBar1.Max := n * w;
  ProgressBar1.Position := 0;
  for i := 1 to n * w do
    for k := 1 to m do
      begin
        if StringGrid1.Cells[k,i] = " then
          StringGrid1.Cells[k,i] := '0';
        end;
      SetLength(a, w);
      SetLength(uxshashliklar, w);
      for z := 0 to w - 1 do
        begin
          SetLength(a[z], n);
          for i := 0 to n - 1 do
            begin
              SetLength(a[z][i], m);
              for k := 0 to m - 1 do
                begin
                  a[z][i][k] := StrToInt(StringGrid1.Cells[k + 1, z * n + i + 1]);
                end;
              end;
            end;
          end;
          setLength(unknown, m);
          for k := 0 to m - 1 do
            // unknown[k] := StrToIntDef(StringGrid2.Cells[k+1, 1], 0);
            Memo1.Lines.Clear;
            // for z := 0 to w - 1 do
            for z := 1 to 1 do
              begin
                // AddToLog('-----' + IntToStr(z + 1) + ' - inchi klass -----'#13#10);
                // Memo1.Lines.Add(IntToStr(z + 1) + ' - inchi klass ');
                for i := 0 to n - 1 do
                  begin
                    Memo1.Lines.Add('D(S' + IntToStr(i+1) + ') = ' + Hisoblash(i,z,uxshashliklar[z]));

```

```

    ProgressBar1.StepIt;
    Application.ProcessMessages;
end;
Memo1.Lines.Add("");
end;
i := 0;
for z := 1 to w - 1 do
    if uxshashliklar[z] > uxshashliklar[i] then
        i := z;
// Memo1.Lines.Add('Noma"lum obyekt ' + IntToStr(i+1) + ' - inchi klassga tegishli');
for z := 0 to w - 1 do
    begin
        for i := 0 to n - 1 do
            SetLength(a[z][i], 0);
            SetLength(a[z], 0);
        end;
        SetLength(a, 0);
        ProgressBar1.Visible := false;
//Label6.Caption := 'Hisoblashga sarflangan vaqt: ' + FormatDateTime('hh:nn:ss', Now - x);
        TControl(Sender).Enabled := true;
// Memo1.Lines.Add(Log5);
        Memo2.Lines.Add(Log5);
    end;
procedure TForm1.Button3Click(Sender: TObject);
begin
    TControl(Sender).Enabled := false;
    Button3.Caption := 'Kuting...';
    Form2.Memo1.Text := Log;
// Memo2.Text:= Log5;
    Memo1.Lines.Add(Log5);
    Form2.ShowModal;
    TControl(Sender).Enabled := true;
    Button3.Caption := 'Log';
end;
procedure TForm1.SpinEdit1Change(Sender: TObject);
begin
    n := SpinEdit1.Value;
    StringGrid1.Repaint;
end;
procedure TForm1.Button4Click(Sender: TObject);
var
    i,k: integer;
begin
    for i := 1 to m do
        for k := 1 to n * w do
            StringGrid1.Cells[i,k] := IntToStr(Random(2));

```

```

    for i := 1 to m do
// StringGrid2.Cells[i, 1] := IntToStr(Random(2));
    end;
procedure TForm1.FormCreate(Sender: TObject);
begin
    n := SpinEdit1.Value;
    m := SpinEdit2.Value;
    w := SpinEdit1.Value;
    Randomize;
    colortable[1] := $dfe5e6;
    colortable[2] := $d6dada;
    colortable[3] := $c5cbcc;
    colortable[4] := $cdcdcd;
    colortable[5] := $f0f0f0;
    end;
procedure TForm1.Button5Click(Sender: TObject);
var
    i, k: integer;
begin
    StringGrid1.ColWidths[0] := 40;
// StringGrid2.ColWidths[0] := 40;
    StringGrid1.ColCount := m + 1;
//StringGrid2.ColCount := m + 1;
// StringGrid1.RowCount := 1 + w * n;
    StringGrid1.RowCount := 1 + w;
    for i := 1 to m do
    begin
        StringGrid1.Cells[i, 0] := 'x' + IntToStr(i);
// StringGrid2.Cells[i, 0] := 'x' + IntToStr(i);
        for k := 1 to n * w do
StringGrid1.Cells[i,k] := IntToStr(Random(80));
        end;
//StringGrid2.Cells[0,1] := 'S*';
        for k := 0 to w - 1 do
            for i := 0 to n - 1 do
                StringGrid1.Cells[0, k * n + i + 1] := ' S' + IntToStr(i+1);
            for k := 1 to w - 1 do
                for i := 1 to n - 1 do
// StringGrid2.Cells[i, k] := IntToStr(Random(60));
                StringGrid1.Cells[i, k] := IntToStr(Random(60));
            end;
end;
procedure TForm1.SpinEdit2Change(Sender: TObject);
begin
    m := SpinEdit2.Value;
    StringGrid1.Repaint;
end;

```

```

procedure TForm1.SpinEdit3Change(Sender: TObject);
begin
  w := SpinEdit1.Value;
  StringGrid1.Repaint;
end;
procedure TForm1.Button6Click(Sender: TObject);
begin
  Form1.Close;
end;
procedure TForm1.Button7Click(Sender: TObject);
var
  i,k: integer;
begin
  for i := 1 to m do
    for k := 1 to n * w do
      if StrToInt(StringGrid1.Cells[i,k])<0 then
        ShowMessage('Malumotlar notogri kiritilgan');
      end;
    end.
end.

```