Projektif Geometri ve Fiber Projektif Geometride C# Uygulamaları Üzerine

Hakkı Keskin

YÜKSEK LİSANS TEZİ
Matematik ve Bilgisayar Bilimleri Anabilim Dalı
Temmuz 2013

On the C# Applications in Projective Geometry and Fibered Projective Geometry

Hakkı Keskin

MASTER OF SCIENCE THESIS

Department of Mathematics and Computer Sciences

July 2013

Projektif Geometri ve Fiber Projektif Geometride C# Uygulamaları Üzerine

Hakkı Keskin

Eskişehir Osmangazi Üniversitesi

Fen Bilimleri Enstitüsü

Lisansüstü Yönetmeliği Uyarınca

Matematik ve Bilgisayar Bilimleri Anabilim Dalı

Geometri Bilim Dalında

YÜKSEK LİSANS TEZİ

Olarak Hazırlanmıştır

Danışman: Doç. Dr. Süheyla Ekmekçi

Temmuz 2013

ONAY

Matematik ve Bilgisayar Bilimleri Anabilim Dalı yüksek lisans öğrencisi Hakkı Keskin' in YÜKSEK LİSANS TEZİ olarak hazırladığı "**Projektif Geometri ve Fiber Projektif Geometride C# Uygulamaları Üzerine**" başlıklı bu çalışma, jürimizce lisans üstü yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

Danışman : Doç. Dr. Süheyla EKMEKÇİ

İkinci Danışman : –

Yüksek Lisans Tez Savunma Jürisi:

Üye : Prof. Dr. Münevver ÖZCAN

Üye : Prof. Dr. Ziya AKÇA

Üye : Doç. Dr. Ayşe BAYAR

Üye : Doç. Dr. Süheyla EKMEKÇİ

Üye : Doç. Dr. Aytaç KURTULUŞ

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..... tarih ve ..... sayılı kararıyla onaylanmıştır.

Prof. Dr. Nimetullah BURNAK

Enstitü Müdürü

ÖZET

Bu çalışmada, Galois cisimlerinden elde edilen sonlu projektif düzlemlerin nokta, doğru ve üzerinde bulunma bağıntısını veren bir C# uygulaması verildi. Daha sonra GF(2) ve GF(3) Galois cisimlerinden elde edilen projektif düzlemlerinin minimum operatörü yardımıyla fiberleştirme uygulamaları C# ortamında yapıldı.

Anahtar Kelimeler: Projektif Düzlem, C#, Fiber Projektif Düzlem

SUMMARY

In this study, a C# application is given that builds projective planes' points, lines and incidence relation, has been developed, planes are based on Galois fields. Also, an another C# application builds fibred projective planes by minimum operator which their base planes are GF(2) and GF(3).

Keywords: Projective Planes, C#, Fibred Projective Planes

TEŞEKKÜR

Yüksek Lisans çalışmalarında, gerek derslerimde ve gerekse tez çalışmalarında, bana danışmanlık ederek, beni yönlendiren ve her türlü olanağı sağlayan danışmanım

Doç. Dr. Süheyla Ekmekçi

başta olmak üzere

çalışmalarımda bana maddi manevi destek olan değerli,

Sevgül Ayan'a, **Hasbi Seçkin Akdemir**'e,

ve beni akademik alanda her zaman destekleyen,

sevgili babam Orhan Keskin'e

sonsuz saygı ve teşekkürlerimi sunarım.

anneme ithaf olunur...

İÇİNDEKİLER

BÖLÜM 0

Giriş

Bu çalışmada, sonlu projektif düzlemlerin ve bu düzlemler üzerine inşa edilen bazı fiber projektif düzlemlerin elemanlarını bilgisayar yardımıyla oluşturmak hedeflendi. Çalışmada .Net platformu üzerinde C# yazılım dili kullanılmış olup açık kaynak kodları bölümler halinde çalışmaların sonlarına eklendi. Çalışmada C# yazılım dilinin kullanılma sebebi olarak windows tabanlı olması, nesnel bir programlama dili olması ve çoğu yazılım geliştiriciler tarafından bilinmesi söylenebilir. Programın amacları arasında, Galois cisimleri ve fiber projektif düzlemler üzerine yapılacak çalışmalarda bilgisayar yardımıyla kolay hesaplamalar yapabilmek, programa farklı değerler atayarak yapılarda oluşan sonuçları gözlemlemek ve yapılan hesaplamaları kontrol etmek sayılabilir. Çalışma, alanında yapılan ilk bilgisayar programı olma özelliği taşımaktadır.

İlk olarak, birinci bölümde gerekli olan bazı temel kavramlara yer verilmiştir.

İkinci bölümde Galois cisimleriyle elde edilen projektif düzlemlerin noktalarını, doğrularını ve üzerinde bulunma bağıntısını veren C# uygulaması verilmiştir.

Üçüncü bölümde (Kuijken, 1999)'un minimum operatörüne göre fiberleştirdiği 2. mertebeden projektif düzlemi için C# uygulaması yapılmıştır.

Dördüncü bölümde yine minimum operatörüne göre 3. mertebeden projektif düzlem fiberleştirildi ve C# uygulaması yapılmıştır.

Sonuç bölümünde, programın yeterlilikleri ve eksik yönlerine değinildi ve bu çalışmayı baz alarak yapılabilecek yeni çalışmalar hakkında bilgiler verilmiştir.

BÖLÜM 1

Bazı Temel Kavramlar

1.1 Projektif Düzlem Hakkında Temel Kavramlar

1.1.1 Sonlu cisimler

Tanım 1.1 $F \neq \varnothing$ olmak üzere;

i. $(F,+)$ abel grup,

ii. $F^* = F - \{0\}$ olmak üzere, $(F^*,.)$ abel grup,

iii. $a.(b+c) = a.b + a.c$ ve $(a+b).c = a.c + b.c$ ise $(F,+,.)$ sistemine **cisim** denir.

Tanım 1.2 Verilen her p asal sayısı ve r pozitif tamsayısı için karakteristiği p olan p^r elemanlı bir tek **sonlu cisim** vardır. Bu cisme mertebesi p^r olan **Galois cismi** denir ve $q = p^r$ olmak üzere $GF(q)$ ile gösterilir.

1.1.2 Projektif düzlemler

Önce projektif geometrinin, bu çalışmada geçen, bazı temel kavramlarını kısaca verelim. Bu bölümde verilecek olan tanım, teorem ve ispatlar (Kaya R, 2005) ve (Kuijken, 1999) den alınmıştır.

Tanım 1.3 $\mathcal{N}$ ve $\mathcal{D}$ elemanları sırasıyla noktalar ve doğrular olan ayrık iki küme ve $\mathcal{N} \cap \mathcal{D} = \varnothing$ olsun. o da $\mathcal{N} \times \mathcal{D}$ kümesinde bir üzerinde bulunma bağıntısı olmak üzere aşağıdaki *P1*, *P2*, *P3* aksiyomlarını sağlayan $\mathbb{P} = (\mathcal{N}, \mathcal{D}, \circ)$ sistemine **projektif düzlem** denir (Kaya R, 2005).

P1) $\forall A, B \in \mathcal{N}, A \neq B$ için $A \circ d$ ve $B \circ d$ olacak şekilde bir tek $d \in \mathcal{D}$ doğrusu vardır.

P2) $\forall c, d \in \mathcal{D}$ için $A \circ c$ ve $A \circ d$ olacak şekilde en az bir $A \in \mathcal{N}$ noktası vardır.

P3) Herhangi üçü doğrudaş olmayan dört nokta vardır.

Teorem 1.4 Bir $\mathbb{P} = (\mathcal{N}, \mathcal{D}, \circ)$, projektif düzleminde farklı iki doğru bir tek noktada kesişirler.

Teorem 1.5 Her sonlu $\mathbb{P} = (\mathcal{N}, \mathcal{D}, \circ)$ projektif düzlemi için aşağıdaki koşulları saglayan bir n pozitif tamsayısı vardır. (Bu tam sayıya ilgili projektif düzlemin **mertebesi** denir):

i) $\mathbb{P}$ nin her doğrusu üzerinde $n+1$ nokta vardır.

ii) $\mathbb{P}$ nin her noktasından $n+1$ doğru geçer.

iii) $\mathbb{P}$ deki tüm noktaların sayısı n^2+n+1 dir.

iv) $\mathbb{P}$ deki tüm doğruların sayısı n^2+n+1 dir.

1.1.3 Galois cisimlerinden elde edilen projektif düzlemler

Teorem 1.6 Verilen her F cismi için nokta ve doğruları bu cismin elemanlarıyla belirtilebilen bir projektif düzlem vardır.

$$((p^r)^3-1)/(p^r-1) = (p^r)^2+(p^r)+1$$

nokta vardır. Bu da düzlemin mertebesinin p^r olduğunu gösterir. Yani, her r pozitif tamsayısı ve her p asal sayısı için mertebesi $n = p^r$ olan sonlu bir projektif düzlem vardır (Kaya R, 2005).

Teorem 1.7 $\mathbb{P} = (\mathcal{N}, \mathcal{D}, \circ)$ mertebesi p^r (p asal, $r \in \mathbb{Z}^+$) olan sonlu mertebeli bir projektif düzlem olsun ve $\mathbb{P}$ projektif düzleminin nokta ve doğrularını sırasıyla (x_1, x_2, x_3) ve $[a_1, a_2, a_3]$ şeklinde homojen temsille gösterelim. O halde $\mathbb{P}$ düzleminin nokta ve doğruları aşağıdaki şekilde ifade edilebilir;

$$\begin{aligned} \mathcal{N} &= (0,0,1) \cup \{(0,1,x_3) | x_3 \in GF(q)\} \cup \{(1,x_2,x_3) | x_2, x_3 \in GF(q)\} \\ \mathcal{D} &= [0,0,1] \cup \{[0,1,a_3] | a_3 \in GF(q)\} \cup \{[1,a_2,a_3] | a_2, a_3 \in GF(q)\} \end{aligned}$$

1.2 Fiber Projektif Düzlemler Hakkında Temel Kavramlar

1.2.1 Bulanık kümeler

Tanım 1.8 λ **bulanık kümesi**, X kümesi üzerinde

$$\begin{aligned} \lambda : \quad & X \to [0,1] \\ & x \to \lambda(x) \end{aligned}$$

şeklinde tanımlı dönüşümdür. $\lambda(x)$ sayısı λ daki x noktasının **üyelik derecesi** şeklinde adlandırılır. X kümesi üzerindeki λ ve μ gibi *iki fuzzy kümelerinin arakesiti*

$$\begin{aligned} \lambda \wedge \mu : \quad & X \to [0,1] \\ & x \to \lambda(x) \wedge \mu(x) \end{aligned}$$

şeklinde tanımlanan $\lambda \wedge \mu$ fuzzy kümesidir. Burada, $\wedge$ **minimum operatörünü** ifade eder, yani bu sayılardan minimum olanıalınır (Kuijken, 1999).

1.2.2 Fiber Noktalar ve Fiber Doğrular

Nokta kümesi P, doğru kümesi B olan nokta-doğru geometrisi $\mathcal{P} = (P, B, I)$ olsun. $\mathcal{P}$, yaklaşık lineer uzay olsun. Yani $\mathcal{P}$ de farklı iki nokta en fazla bir doğru belirtirken bir doğru üzerinde en az iki nokta vardır. p ve q noktalarının üzerinde bulunduğu ortak doğru $\langle p,q \rangle$ ile gösterilir ve p ile q **doğrudaş** denir. L ve M ortak bir noktada kesişen doğrular ise bu nokta $L \cap M$ ile gösterilip L ve M nin **kesişimi** olarak adlandırılır. Aşağıda f-nokta ve f-doğru şeklinde kısaca adlandırdığımız fiber doğruları ve fiber noktaları tanımlanmaktadır (Kuijken, 1999).

Tanım 1.9 $a \in P$ ve $\alpha \in (0,1]$ olsun.

$$\begin{aligned} (a,\alpha) : \quad P &\to [0,1] \\ a &\to \alpha \\ x &\to 0,\ x \in P \setminus \{a\} \end{aligned}$$

biçiminde tanımlanan (a,α) sıralı ikilisine f**-noktası** denir (Kuijken, 1999).

Tanım 1.10 Dual olarak , $L \in B$ ve $\beta \in (0,1]$ için

$$\begin{aligned} (L,\beta) : \quad B &\to [0,1] \\ L &\to \beta \\ X &\to 0,\ X \in B \setminus \{L\} \end{aligned}$$

biçiminde tanımlanan (L,β) sıralı ikilisine f-**doğrusu** denir. (Kuijken, 1999).

Bu tanımlamayla herhangi bir (a,α) f-noktası, $\mathcal{P}$ nin a noktasının sıfırdan farklı bir α üyelik derecesi almış bir noktasıdır. a, (a,α) f-noktasının **taban noktası** olarak adlandırılır. Farklı f-noktaları aynı taban noktasına sahip olabilir.

Dual olarak bir (L,β) f-doğrusu, $\mathcal{P}$ nin L doğrusunun sıfırdan farklı bir β üyelik derecesi almış bir doğrusudur. L ye (L,β) f-doğrusunun **taban doğrusu** denir. Benzer olarak farklı f-doğruları aynı taban doğrusuna sahip olabilir.

Tanım 1.11 (L,α) ve (M,β) **f-doğrularının arakesit noktası** $(L\cap M,\alpha\wedge\beta)$ şeklinde tanımlı bir tek f-noktasıdır (Kuijken, 1999).

(a,λ) ve (b,β) **f-noktalarının gerdiği f-doğrusu** $(\langle a,b\rangle,\lambda\wedge\beta)$ şeklinde tanımlanır ve bu doğru tektir (Kuijken, 1999).

1.3 Fiber Projektif Düzlemler

Tanım 1.12 $\mathcal{P}=(P,B,I)$ bir projektif düzlem olsun. $\mathcal{F}P$, $\mathcal{P}$ nin sıfırdan farklı en az bir taban noktasına sahip olan noktalar kümesi ve $\mathcal{F}B$, $\mathcal{P}$ nin sıfırdan farklı en az bir taban doğrusuna sahip olan doğrular kümesi olsun. Eğer aşağıdaki iki koşul sağlanırsa $(\mathcal{F}P,\mathcal{F}B)$ yapısına bir **fiber projektif düzlem** denir (Kuijken, 1999).

F1) Farklı taban noktalarından oluşan her f-nokta çifti yalnız bir f-doğrusunu gerer.

F2) Farklı taban doğrularından oluşan her f-doğru çifti yalnız bir f-noktasında kesişir.

1.3.1 Doğrudaş f-noktalar

Tanım 1.13 f- noktalarının her bir çifti aynı f- doğrusunu gererse bu f-noktalar kümesine **doğrudaş fiber noktalar** denir (Kuijken, 1999).

1.3.2 Kesişen f-doğrular

Tanım 1.14 İkişer ikişer aynı bir f-noktada kesişen f-doğrulara **noktadaş f-doğrular** ya da **kesişen f-doğrular** denir (Kuijken, 1999).

$\mathcal{P}$ projektif düzlemi, $\mathcal{F}\mathcal{P}$ nin **taban düzlemi** olarak adlandırılır. Aşağıdaki yolla bir fiber projektif düzlem üretilir. $P'\subseteq P$ ve $B'\subseteq B$ olsun. $P'\cup B'$ yi içeren tek kapalı konfigürasyon $P\cup B$ olacak şekilde, $P'\cup B'$ nin her bir x elemanı için $]0,1]$ in bir keyfi $\sum_x$ boş olmayan alt kümesini seçelim, bu alt kümenin elemanlarına x in **başlangıç değerleri** denir. $\mathcal{F}\mathcal{P}$ aşağıdaki gibi bir fiber projektif düzlem olarak tanımlanır. Her bir $x\in P'\cup B'$ ve her bir $\alpha\in\sum_x$ için (x,α) elemanı $\mathcal{F}\mathcal{P}$ ye aittir. Bu oluşumun ilk adımıdır. Şimdi i. adımını tanımlayalım, $i>1$ için elde ettiğimiz f-noktalarının herhangi bir çifti bu çiftle gerilmiş f-doğrusu da tanımla $\mathcal{F}\mathcal{P}$ ye aittir. Dual

olarak , f-doğrularının herhangi bir çifti için kesişim $\mathcal{FP}$ ye aittir. Sonlu sayıda adımla bu yolla oluşturulmuş bütün f doğruların ve noktaların kümesi, bir fiber projektif düzlem oluşturmak için kullanılır. Anlaşılıyor ki her fiber projektif düzlem yukarıda olduğu gibi inşa edilebilir. Aslında, herbir elaman için onun bütün denk değerleri başlangıç değerleri olarak her zaman alınabilir. Her $x \in P \cup B$ için $\sum_x$ tek bir küme olsun. Eğer $P' = P$ ve $B' = \varnothing$ ise fiber projektif düzlem **mono-point-generated** olarak adlandırılır. Eğer $P' = P$ ve $B' = B$ ise o zaman fiber projektif düzlem **mono-generated** olarak adlandırılır. $\mathcal{FP}$ sıradan bir projektif düzlemin her nokta ve doğrusuna $]0,1]$ değerler kümesinden değer verilerek elde edilen bir projektif düzlem olarak düşünülebilir(Kuijken, 1999).

BÖLÜM 2

Galois Cisimlerinden Elde Edilen Projektif Düzlem Elemanlarının C# ile Elde Edilmesi

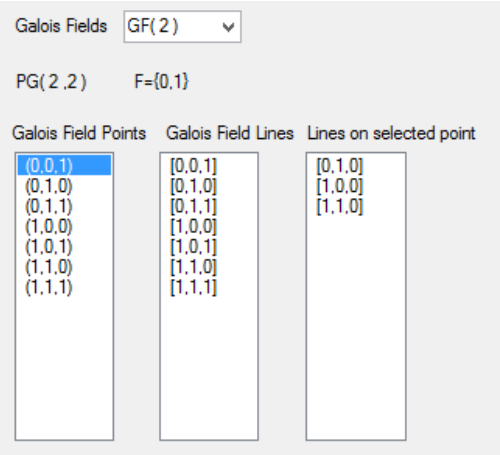
Bu bölümde, Bölüm 1 deki teoremde yer alan mertebesi p^r olan sonlu bir projektif düzlemin nokta ve doğru kümeleri .Net platformunda C# yazılım dili kullanılarak oluşturulmuştur. İkinci adımda ise elde edilen noktalardan birinin seçilmesi durumunda bu noktadan geçen doğru kümesi ve elde edilen doğrulardan birinin seçilmesi durumunda bu doğrunun üzerinde bulunan noktalar kümesi elde edilmiştir. Diğer bir ifadeyle var olan nokta ve doğru kümelerinde, üzerinde bulunma bağıntısı uygulanmış ve sağlayan nokta ya da doğru kümeleri listelenmiştir.

Programda $GF(2), GF(3), GF(5), GF(7), GF(11)$ cisimleri üzerine kurulan projektif düzlemler hakkında bazı kombinatöryel bilgiler Tablo 2.1'de verilmiştir.

	$GF(2)$	$GF(3)$	$GF(5)$	$GF(7)$	$GF(11)$
Nokta Sayısı	7	13	31	57	133
Doğru Sayısı	7	13	31	57	133
Noktadan Geçen Doğru Sayısı	3	4	6	8	12
Doğru Üzerinde Bulunan Nokta Sayısı	3	4	6	8	12

Tablo 2.1 Sonlu projektif düzlem elemanları

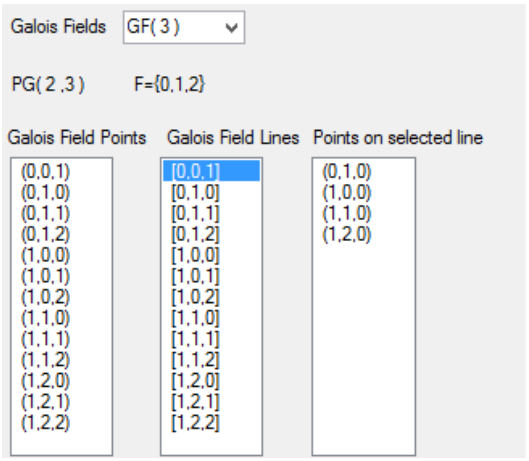
Şekil 2.1 de programın bir sayfa görüntüsüne yer verilmiştir. Programın üst kısmında görüldüğü üzere GF(2) cisminden elde edilen projektif düzlem seçilmiş ve alt kısımlarında cismin elemanları, noktaları ve doğruları listelenmiştir. İkinci adımda ise listenen nokta kümesinden ilk nokta seçilmiş ve "lines on selected point" kısmında bu nokta üzerindenden geçen 3 doğru belirmiştir. Bu işlem farklı noktalar için denendiğinde beliren doğrular değişmektedir.



Şekil 2.1. PG(2,2) Elemanlari

Şekil 2.2 de yine programın bir sayfa görüntüsüne yer verilmiştir. Programın üst kısmında

görüldüğü üzere bu sefer GF(3) cisminden elde edilen projektif düzlem seçilmiş ve alt kısımlarında cismin elemanları, noktaları ve doğruları listelenmiştir. Bu defa ikinci adımda listenen doğru kümesinden ilk nokta seçilmiş ve ”Points on selected line” kısmında bu doğru üzerinde yer alan 4 nokta belirmiştir. Bu işlem farklı doğrular için denendiğinde beliren noktalar değişmektedir.



Şekil 2.2. PG(2,3) Elemanlari

Programda kullanılan kodlar aşağıdaki çizelgede verilmiştir. Kodlar yardımıyla asal mertebeye

sahip tüm Galois cisimleri için nokta, doğru ve üzerinde bulunma kümeleri listelenebilir.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace Projectivator
{
public partial class Form1 : Form
{
public int galoisFieldDimension;
public int galoisFieldOrder;
public int[,] point_buffer = new int[900, 3];
public int point_bufferEndPoint = 0;
public int[,] point_buffer1 = new int[900, 3];
public int point_buffer1EndPoint = 0;
public int[,] line_buffer = new int[900, 3];
public int line_bufferEndPoint = 0;
public int[,] linesOnPoint_buffer = new int[900, 3];
public int linesOnPoint_bufferEndPoint = 0;
public int[,] pointsOnLine_buffer = new int[900, 3];
public int pointsOnLine_bufferEndPoint = 0;
public int function;
public Form1()
{
InitializeComponent();
}
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
string galoisField = cmbBoxGaloisFields.Text;
listBoxGaloisFieldsPoints.Items.Clear();
listBoxGaloisFieldsLines.Items.Clear();
listBoxOn.Items.Clear();
point_bufferEndPoint = 0;
point_buffer1EndPoint=0;
line_bufferEndPoint = 0;
galoisFieldDimension = Convert.ToInt16(galoisField.Substring(4, 2));
galoisFieldOrder = galoisFieldDimension;
lblPro.Text = ”PG( 2 ” + ”,” + galoisFieldOrder + ” )”;
if (galoisFieldOrder == 2) lblF.Text = ”F={0,1}”;
if (galoisFieldOrder == 3) lblF.Text = ”F={0,1,2}”;
if (galoisFieldOrder == 4) lblF.Text = ”F={0,1,a,a+1}”;
for (int firstPoints = 0; firstPoints <= 1; firstPoints++)
{
int secondValue;
```

```
if (firstPoints == 0) secondValue = 1;
else secondValue = galoisFieldDimension - 1;
for (int secondPoints = 0; secondPoints <= secondValue; secondPoints++)
{
int thirdValue;
if (firstPoints == 0 & secondPoints==0) thirdValue = 1;
else thirdValue = galoisFieldDimension - 1;
for (int thirdPoints = 0; thirdPoints <= thirdValue; thirdPoints++)
{
point_buffer[point_bufferEndPoint, 0] = firstPoints;
point_buffer[point_bufferEndPoint, 1] = secondPoints;
point_buffer[point_bufferEndPoint, 2] = thirdPoints;
if (firstPoints == 0 & secondPoints == 0 & thirdPoints == 0)
{ }
else point_bufferEndPoint++;
}
}
label4.Text = Convert.ToString(point_bufferEndPoint);
}
for (int lines = 0; lines < point_bufferEndPoint; lines++)
{
line_buffer[lines, 0] = point_buffer[lines, 0];
line_buffer[lines, 1] = point_buffer[lines, 1];
line_buffer[lines, 2] = point_buffer[lines, 2];
line_bufferEndPoint++;
}
for (int listPoints = 0; listPoints < point_bufferEndPoint; listPoints++)
{
listBoxGaloisFieldsPoints.Items.Add(" (" + point_buffer[listPoints, 0] + "," +
point_buffer[listPoints, 1] + "," + point_buffer[listPoints, 2] + ")");
}
for (int listLines = 0; listLines < line_bufferEndPoint; listLines++)
{
listBoxGaloisFieldsLines.Items.Add(" [" + line_buffer[listLines, 0] + "," +
line_buffer[listLines, 1] + "," + line_buffer[listLines, 2] + "]");
}
listBoxGaloisFieldsPoints.Visible = true;
listBoxGaloisFieldsLines.Visible = true;
lblLines.Visible = true;
lblPoints.Visible = true;
lblLines.Text = "Galois Field Lines";
lblPoints.Text = "Galois Field Points";
}
private void Form1_Load(object sender, EventArgs e)
```

```
{
listBoxGaloisFieldsLines.Visible = false;
listBoxGaloisFieldsPoints.Visible = false;
lblLines.Visible = false;
lblPoints.Visible = false;
listBoxOn.Visible = false;
lblOn.Visible = false;
}
private void btnExit_Click(object sender, EventArgs e)
{
this.Close();
}
private void listBoxGaloisFieldsPoints_SelectedIndexChanged(object sender, EventArgs
e) {
listBoxOn.Items.Clear();
linesOnPoint_bufferEndPoint = 0;
listBoxOn.Visible = true;
lblOn.Visible = true;
lblOn.Text = " Lines on selected point";
string selectedPoint = listBoxGaloisFieldsPoints.Text;
for (int linesOnPoint = 0; linesOnPoint < line_bufferEndPoint; linesOnPoint++)
{
int firstLine = line_buffer[linesOnPoint, 0];
int secondLine = line_buffer[linesOnPoint, 1];
int thirdLine = line_buffer[linesOnPoint, 2];
int myFirstPoint = Convert.ToInt16(selectedPoint.Substring(2, 1));
int mySecondPoint = Convert.ToInt16(selectedPoint.Substring(4, 1));
int myThirdPoint = Convert.ToInt16(selectedPoint.Substring(6, 1));
int result = firstLine * myFirstPoint + secondLine * mySecondPoint + thirdLine *
myThirdPoint;
checkModLoop:
int checkMod = result;
if (checkMod >= galoisFieldOrder)
{
result = result - galoisFieldOrder;
if (result >= galoisFieldOrder) goto checkModLoop;
}
if (result == 0)
{
linesOnPoint_buffer[linesOnPoint_bufferEndPoint, 0] = firstLine;
linesOnPoint_buffer[linesOnPoint_bufferEndPoint, 1] = secondLine;
linesOnPoint_buffer[linesOnPoint_bufferEndPoint, 2] = thirdLine;
linesOnPoint_bufferEndPoint++;
}
}
for (int listLinesOnPoint = 0; listLinesOnPoint < linesOnPoint_bufferEndPoint; listLi-
nesOnPoint++)
{
```

```
listBoxOn.Items.Add(” [” + linesOnPoint_buffer[listLinesOnPoint, 0] + ”,” + linesOn-
Point_buffer[listLinesOnPoint, 1] + ”,” + linesOnPoint_buffer[listLinesOnPoint, 2] + ”]”);
}
}
private void listBoxGaloisFieldsLines_SelectedIndexChanged(object sender, EventArgs
e)
{
listBoxOn.Visible = true;
lblOn.Visible = true;
lblOn.Text = ” Points on selected line”;
pointsOnLine_bufferEndPoint = 0;
listBoxOn.Items.Clear();
string selectedLine = listBoxGaloisFieldsLines.Text;
for (int pointsOnLine = 0; pointsOnLine < point_bufferEndPoint; pointsOnLine++)
{
int firstPoint = point_buffer[pointsOnLine, 0];
int secondPoint = point_buffer[pointsOnLine, 1];
int thirdPoint = point_buffer[pointsOnLine, 2];
int myFirstLine = Convert.ToInt16(selectedLine.Substring(2, 1));
int mySecondLine = Convert.ToInt16(selectedLine.Substring(4, 1));
int myThirdLine = Convert.ToInt16(selectedLine.Substring(6, 1));
int result = firstPoint * myFirstLine + secondPoint * mySecondLine + thirdPoint *
myThirdLine;
checkModLoop:
int checkMod = result;
if (checkMod >= galoisFieldOrder)
{
result = result - galoisFieldOrder;
if (result >= galoisFieldOrder) goto checkModLoop;
}
if (result == 0)
{
pointsOnLine_buffer[pointsOnLine_bufferEndPoint, 0] = firstPoint;
pointsOnLine_buffer[pointsOnLine_bufferEndPoint, 1] = secondPoint;
pointsOnLine_buffer[pointsOnLine_bufferEndPoint, 2] = thirdPoint;
pointsOnLine_bufferEndPoint++;
}
}
for (int listPointsOnLine = 0; listPointsOnLine < pointsOnLine_bufferEndPoint; list-
PointsOnLine++)
{
listBoxOn.Items.Add(” (” + pointsOnLine_buffer[listPointsOnLine, 0] + ”,” + pointsOn-
Line_buffer[listPointsOnLine, 1] + ”,” + pointsOnLine_buffer[listPointsOnLine, 2] + ”)”);
}
}
```

BÖLÜM 3

Minimum Operatörü Yardımıyla Fiberleştirilmiş Fano Düzleminin Nokta ve Doğrularının Üyelik Derecelerinin C# ile Uygulaması

Bu bölümde, bilinen en küçük projektif düzlem olan Fano düzleminin minimum operatörü ile fiberleştirilmesi ele alınmıştır. Aşağıda PG(2,2) düzleminin yedi noktası için Kuijken tarafından yapılmış olan örneğe yer verilmiştir.

Örnek 3.1 $\mathcal{F}=PG(2,2)$ düzlemini ele alalım. $\mathcal{F}$ taban düzlemi ile bir mono-point-generated fiber projektif düzlemi inşa edilecektir. $\mathcal{F}$ nin yedi noktası ve yedi doğrusu sırasıyla $\{a,b,c,d,e,f,g\}$ ve $\{A,B,C,D,E,F,G\}$ ile gösterilmektedir. Burada $A=\{a,b,c\}$, $B=\{c,d,e\}$, $C=\{e,f,a\}$, $D=\{a,g,d\}$, $E=\{b,g,e\}$, $F=\{c,g,f\}$, $G=\{b,d,f\}$ dir. 1. adımda P nin noktaları üzerinde $(a\ \ 0,9)$, $(b\ \ 0,8)$, $(c\ \ 0,7)$, $(d\ \ 0,6)$, $(e\ \ 0,3)$, $(f\ \ 0,4)$ ve $(g\ \ 0,5)$ f-noktalarını alalım. Böylece 0,9 0,8 0,7 0,6 0,3 0,4 ve 0,5 sırasıyla **a**, **b**, **c**, **d**, **e**, **f** ve **g** taban noktalarının başlangıç değerleridir. İlk oluşumda A doğrusu üzerindeki $(a\ \ 0,9)$, $(b\ \ 0,8)$, $(c\ \ 0,7)$ fiber noktaları taban doğrusu A olan ikişer ikişer farklı fiber doğrular oluşturur. Bu fiber doğruların üyelik dereceleri minimum operatörü kullanılarak $\{0,7\ \ 0,8\}$ olarak bulunur. Benzer şekilde tabanı B doğrusu olan fiber doğruların üyelik dereceleri $\{0,3\ \ 0,6\}$, tabanı C doğrusu olan fiber doğruların üyelik dereceleri $\{0,3\ \ 0,4\}$, tabanı D doğrusu olan fiber doğruların üyelik dereceleri $\{0,5\ \ 0,6\}$, tabanı E doğrusu olan fiber doğruların üyelik dereceleri $\{0,3\ \ 0,5\}$, tabanı F doğrusu olan fiber doğruların üyelik dereceleri $\{0,4\ \ 0,5\}$, tabanı G doğrusu olan fiber doğruların üyelik dereceleri $\{0,4\ \ 0,6\}$ dır. 2. aşamada tabanı a,b,c,d,e,f,g noktaları olan fiber noktaların aldığı üyelik dereceleri sırasıyla $\{0,3\ \ 0,4\ \ 0,5\ \ 0,6\}$, $\{0,3\ \ 0,4\ \ 0,5\ \ 0,6\}$, $\{0,3\ \ 0,4\ \ 0,5\ \ 0,6\}$, $\{0,3\ \ 0,4\ \ 0,5\ \ 0,6\}$, $\{0,3\ \ 0,4\ \ 0,5\}$, $\{0,3\ \ 0,4\ \ 0,5\}$, $\{0,3\ \ 0,4\ \ 0,5\}$ kümeleriyle verilebilir. Bunlar kullanılarak A,B,C,D,E,F,G tabanlı fiber doğruların aldığı üyelik dereceleri sırasıyla $\{0,3\ \ 0,4\ \ 0,5\ \ 0,6\ \ 0,7\ \ 0,8\}$, $\{0,3\ \ 0,4\ \ 0,5\ \ 0,6\}$, $\{0,3\ \ 0,4\ \ 0,5\}$, $\{0,3\ \ 0,4\ \ 0,5\ \ 0,6\}$, $\{0,3\ \ 0,4\ \ 0,5\}$, $\{0,3\ \ 0,4\ \ 0.5\}$, $\{0,3\ \ 0,4\ \ 0,5\ \ 0,6\}$ kümeleriyle verilebilir. 3. aşamada yeni fiber nokta ve doğru oluşmaz. Böylece nokta ve doğruları aşağıda verilen fiber projektif düzlemi oluşturulmuş olur (Kuijken, 1999).

$a=\{0,3\ \ 0,4\ \ 0,5\ \ 0,6\ \ 0,9\}$,

$b=\{0,3\ \ 0,4\ \ 0,5\ \ 0,6\ \ 0,8\}$,

$c = \{0,3\ \ 0,4\ \ 0,5\ \ 0,6\ \ 0,7\}$,

$d = \{0,3\ \ 0,4\ \ 0,5\ \ 0,6\}$,

$e = \{0,3\ \ 0,4\ \ 0,5\}$,

$f = \{0,3\ \ 0,4\ \ 0,5\}$,

$g = \{0,3\ \ 0,4\ \ 0,5\}$, ve

$A = \{0,3\ \ 0,4\ \ 0,5\ \ 0,6\ \ 0,7\ \ 0,8\}$,

$B = \{0,3\ \ 0,4\ \ 0,5\ \ 0,6\}$,

$C = \{0,3\ \ 0,4\ \ 0,5\}$,

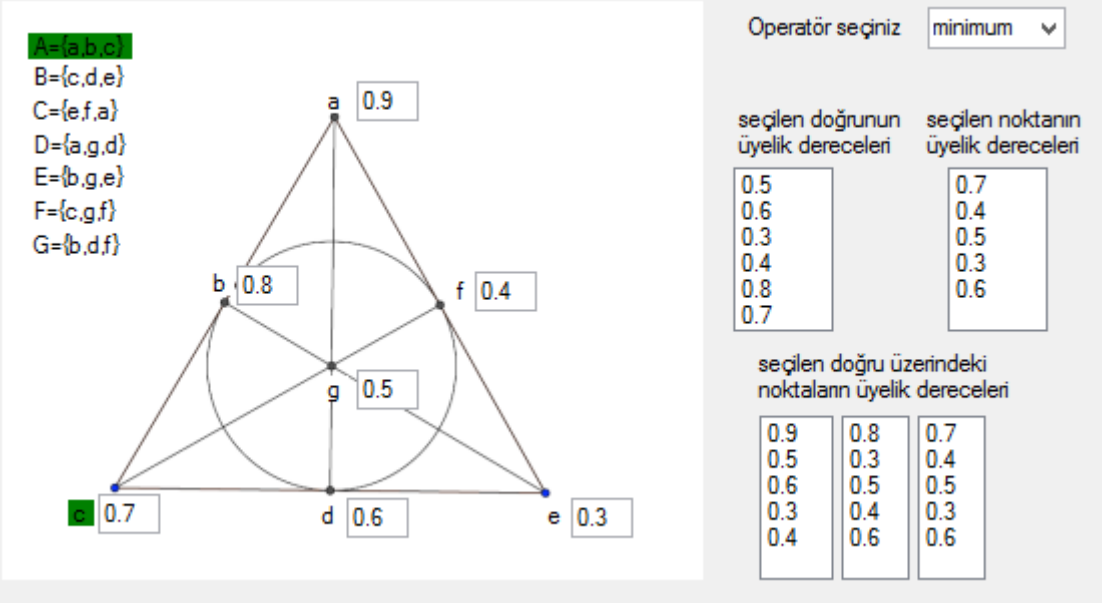
$D = \{0,3\ \ 0,4\ \ 0,5\ \ 0,6\}$,

$E = \{0,3\ \ 0,4\ \ 0,5\}$,

$F = \{0,3\ \ 0,4\ \ 0,5\}$,

$G = \{0,3\ \ 0,4\ \ 0,5\ \ 0,6\}$.

Şekil 3.1 de program görüntüsüne yer verilmiştir. Örnek 3.1 de yer alan başlangıç değerleri programa girilmiş ve *c* noktası seçilerek noktanın almış olduğu üyelik dereceleri ”seçilen noktanın üyelik dereceleri” kısmında belirmiştir. Yine *a, b, c* noktalarının üzerinde bulunduğu *A* doğrusu seçilmiş ve ”seçilen doğru üzerindeki noktaların üyelik dereceleri” kısmında sırasıyla *a, b, c* noktalarının üyelik dereceleri ve bu noktalar yardımıyla ”seçilen doğrunun üyelik dereceleri” kısmında *A* doğrusuna ait üyelik dereceleri belirmiştir.



Şekil 3.1. PG(2,2) de *A* Dogrusu Üyelik Dereceleri

Yukarıda, Kuijken tarafından verilen Fano düzleminin fiberleştirme örneği için C# kodu aşağıdaki çizelgede verilmiştir. Kod yardımıyla PG(2,2) düzlemine ait tüm nokta ve doğruların üyelik dereceleri keyfi başlangıç değerlerine göre belirlenebilir.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace Projectivator
{
public partial class Form1 : Form
{
public double[,] fuzzyLines_buffer = new double[90, 2];
public int fuzzyLines_bufferEndPoint = 0;
public int[,] listFuzzyPoints_buffer = new int[90, 1];
public int listFuzzyPoints_bufferEndPoint = 0;
public int[,] fibringPoints_buffer = new int[90, 1];
public int fibringPoints_bufferEndPoint = 0;
public int[,] fibringLines1_buffer = new int[150, 1];
public int fibringLines1_bufferEndPoint = 0;
public int[,] fibringLines2_buffer = new int[150, 1];
public int fibringLines2_bufferEndPoint = 0;
public int[,] fibringLines3_buffer = new int[150, 1];
public int fibringLines3_bufferEndPoint = 0;
public int[,] fibringLines5_buffer = new int[150, 1];
public int fibringLines5_bufferEndPoint = 0;
public int[,] fibringLines6_buffer = new int[150, 1];
public int fibringLines6_bufferEndPoint = 0;
private void lblPoint1_Click(object sender, EventArgs e)
{
listBoxFuzzyPoints.Items.Clear();
listFuzzyPoints_bufferEndPoint = 0;
lblPoint1.BackColor = Color.Green;
lblPoint2.BackColor = Color.White;
lblPoint3.BackColor = Color.White;
lblPoint4.BackColor = Color.White;
lblPoint5.BackColor = Color.White;
lblPoint6.BackColor = Color.White;
lblPoint7.BackColor = Color.White;
if (txtPoint1.Text == "" | txtPoint3.Text == "" | txtPoint4.Text == "" | txtPoint5.Text ==
"" | txtPoint6.Text == "" | txtPoint7.Text == "")
{
MessageBox.Show("lütfen bütün noktaları giriniz...");
}
else
{
int lenghtPoint1 = txtPoint1.Text.Length;
int lenghtPoint2 = txtPoint2.Text.Length;
int lenghtPoint3 = txtPoint3.Text.Length;
int lenghtPoint4 = txtPoint4.Text.Length;
```

```
int lenghtPoint5 = txtPoint5.Text.Length;
int lenghtPoint6 = txtPoint6.Text.Length;
int lenghtPoint7 = txtPoint7.Text.Length;
string strpoint1 = txtPoint1.Text.Substring(2, lenghtPoint1 - 2);
string strpoint2 = txtPoint2.Text.Substring(2, lenghtPoint2 - 2);
string strpoint3 = txtPoint3.Text.Substring(2, lenghtPoint3 - 2);
string strpoint4 = txtPoint4.Text.Substring(2, lenghtPoint4 - 2);
string strpoint5 = txtPoint5.Text.Substring(2, lenghtPoint5 - 2);
string strpoint6 = txtPoint6.Text.Substring(2, lenghtPoint6 - 2);
string strpoint7 = txtPoint7.Text.Substring(2, lenghtPoint7 - 2);
int point1 = Convert.ToInt16(strpoint1);
int point2 = Convert.ToInt16(strpoint2);
int point3 = Convert.ToInt16(strpoint3);
int point4 = Convert.ToInt16(strpoint4);
int point5 = Convert.ToInt16(strpoint5);
int point6 = Convert.ToInt16(strpoint6);
int point7 = Convert.ToInt16(strpoint7);
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = point1;
listFuzzyPoints_bufferEndPoint++;
if (cmBoxOperator.Text==”minimum”) function=1;
if (cmBoxOperator.Text == ”maximum”) function = 2;
if (function == 1)
{
int myFirstLineFirstFibredPoint = fibringFirstPoint(point3, point2, point1);
int myFirstLineSecondFibredPoint = fibringSecondPoint(point3, point2, point1);
int mySecondLineFirstFibredPoint = fibringFirstPoint(point6, point7, point1);
int mySecondLineSecondFibredPoint = fibringSecondPoint(point6, point7, point1);
int myThirdLineFirstFibredPoint = fibringFirstPoint(point1, point5, point4);
int myThirdLineSecondFibredPoint = fibringSecondPoint(point1, point5, point4);
fibringPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, mySecond-
LineFirstFibredPoint, mySecondLineSecondFibredPoint, myThirdLineFirstFibredPoint,
myThirdLineSecondFibredPoint);
}
if (function == 2)
{
int myFirstLineFirstFibredPoint = fibringMaxFirstPoint(point3, point2, point1);
int myFirstLineSecondFibredPoint = fibringMaxSecondPoint(point3, point2, point1);
int mySecondLineFirstFibredPoint = fibringMaxFirstPoint(point6, point7, point1);

int mySecondLineSecondFibredPoint = fibringMaxSecondPoint(point6, point7, point1);
int myThirdLineFirstFibredPoint = fibringMaxFirstPoint(point1, point5, point4);
int myThirdLineSecondFibredPoint = fibringMaxSecondPoint(point1, point5, point4);
fibringMaxPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, mySec-
ondLineFirstFibredPoint, mySecondLineSecondFibredPoint, myThirdLineFirstFibred-
Point, myThirdLineSecondFibredPoint);
}
}
}
```

```
private void lblPoint3_Click(object sender, EventArgs e)
{
listBoxFuzzyPoints.Items.Clear();
listFuzzyPoints_bufferEndPoint = 0;
if (txtPoint1.Text == "" | txtPoint3.Text == "" | txtPoint4.Text == "" | txtPoint5.Text ==
"" | txtPoint6.Text == "" | txtPoint7.Text == "")
{
MessageBox.Show("lütfen bütün noktaları giriniz...");
}
else
{
int lenghtPoint1 = txtPoint1.Text.Length;
int lenghtPoint2 = txtPoint2.Text.Length;
int lenghtPoint3 = txtPoint3.Text.Length;
int lenghtPoint4 = txtPoint4.Text.Length;
int lenghtPoint5 = txtPoint5.Text.Length;
int lenghtPoint6 = txtPoint6.Text.Length;
int lenghtPoint7 = txtPoint7.Text.Length;
string strpoint1 = txtPoint1.Text.Substring(2, lenghtPoint1 - 2);
string strpoint2 = txtPoint2.Text.Substring(2, lenghtPoint2 - 2);
string strpoint3 = txtPoint3.Text.Substring(2, lenghtPoint3 - 2);
string strpoint4 = txtPoint4.Text.Substring(2, lenghtPoint4 - 2);
string strpoint5 = txtPoint5.Text.Substring(2, lenghtPoint5 - 2);
string strpoint6 = txtPoint6.Text.Substring(2, lenghtPoint6 - 2);
string strpoint7 = txtPoint7.Text.Substring(2, lenghtPoint7 - 2);
int point1 = Convert.ToInt16(strpoint1);
int point2 = Convert.ToInt16(strpoint2);
int point3 = Convert.ToInt16(strpoint3);
int point4 = Convert.ToInt16(strpoint4);
int point5 = Convert.ToInt16(strpoint5);
int point6 = Convert.ToInt16(strpoint6);
int point7 = Convert.ToInt16(strpoint7);
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = point3;
listFuzzyPoints_bufferEndPoint++;
if (cmBoxOperator.Text=="minimum") function=1;
if (cmBoxOperator.Text == "maximum") function = 2;
if (function == 1)
{
int myFirstLineFirstFibredPoint = fibringFirstPoint(point3, point4, point6);
int myFirstLineSecondFibredPoint = fibringSecondPoint(point3, point4, point6);
int mySecondLineFirstFibredPoint = fibringFirstPoint(point3, point2, point1);
int mySecondLineSecondFibredPoint = fibringSecondPoint(point3, point2, point1);
int myThirdLineFirstFibredPoint = fibringFirstPoint(point3, point7, point5);
int myThirdLineSecondFibredPoint = fibringSecondPoint(point3, point7, point5);
fibringPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, mySecond-
LineFirstFibredPoint, mySecondLineSecondFibredPoint, myThirdLineFirstFibredPoint,
myThirdLineSecondFibredPoint);
}
```

```
if (function == 2)
{
int myFirstLineFirstFibredPoint = fibringMaxFirstPoint(point3, point4, point6);
int myFirstLineSecondFibredPoint = fibringMaxSecondPoint(point3, point4, point6);
int mySecondLineFirstFibredPoint = fibringMaxFirstPoint(point3, point2, point1);
int mySecondLineSecondFibredPoint = fibringMaxSecondPoint(point3, point2, point1);
int myThirdLineFirstFibredPoint = fibringMaxFirstPoint(point3, point7, point5);
int myThirdLineSecondFibredPoint = fibringMaxSecondPoint(point3, point7, point5);
fibringMaxPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibredPoint, myThirdLineFirstFibredPoint, myThirdLineSecondFibredPoint);
}
}
}
private void lblPoint5_Click(object sender, EventArgs e)
{
listBoxFuzzyPoints.Items.Clear();
listFuzzyPoints_bufferEndPoint = 0;
if (txtPoint1.Text == "" | txtPoint3.Text == "" | txtPoint4.Text == "" | txtPoint5.Text == "" | txtPoint6.Text == "" | txtPoint7.Text == "") {
MessageBox.Show("lütfen bütün noktaları giriniz...");
}
else
{
int lenghtPoint1 = txtPoint1.Text.Length;
int lenghtPoint2 = txtPoint2.Text.Length;
int lenghtPoint3 = txtPoint3.Text.Length;
int lenghtPoint4 = txtPoint4.Text.Length;
int lenghtPoint5 = txtPoint5.Text.Length;
int lenghtPoint6 = txtPoint6.Text.Length;
int lenghtPoint7 = txtPoint7.Text.Length;
string strpoint1 = txtPoint1.Text.Substring(2, lenghtPoint1 - 2);
string strpoint2 = txtPoint2.Text.Substring(2, lenghtPoint2 - 2);
string strpoint3 = txtPoint3.Text.Substring(2, lenghtPoint3 - 2);
string strpoint4 = txtPoint4.Text.Substring(2, lenghtPoint4 - 2);
string strpoint5 = txtPoint5.Text.Substring(2, lenghtPoint5 - 2);
string strpoint6 = txtPoint6.Text.Substring(2, lenghtPoint6 - 2);
string strpoint7 = txtPoint7.Text.Substring(2, lenghtPoint7 - 2);
int point1 = Convert.ToInt16(strpoint1);
int point2 = Convert.ToInt16(strpoint2);
int point3 = Convert.ToInt16(strpoint3);
int point4 = Convert.ToInt16(strpoint4);
int point5 = Convert.ToInt16(strpoint5);
int point6 = Convert.ToInt16(strpoint6);
int point7 = Convert.ToInt16(strpoint7);
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = point5;
listFuzzyPoints_bufferEndPoint++;
```

```
if (cmBoxOperator.Text==”minimum”) function=1;
if (cmBoxOperator.Text == ”maximum”) function = 2;
if (function == 1)
{
int myFirstLineFirstFibredPoint = fibringFirstPoint(point3, point7, point5);
int myFirstLineSecondFibredPoint = fibringSecondPoint(point3, point7, point5);
int mySecondLineFirstFibredPoint = fibringFirstPoint(point6, point2, point5);
```

BÖLÜM 4

Minimum Operatörü Yardımıyla Fiberleştirilmiş PG(2,3) Düzleminin Nokta ve Doğrularının Üyelik Derecelerinin C# ile Uygulaması

Bu bölümde PG(2,3) düzleminin minimum operatörü ile fiberleştirilmesi ele alınmıştır. PG(2,3) düzleminin on üç noktası için taban değerler atanmış ve bu durumlarda oluşan nokta ve doğruların üyelik dereceleri incelenmiştir.

Örnek 4.1 $\mathcal{F}=PG(2,3)$ düzlemi klasik projektif düzlem olsun. $\mathcal{F}$ taban düzlemi ile bir mono-point-generated fiber projektif düzlemi inşa edilecektir. $\mathcal{F}$ nin onüç noktası ve on üç doğrusu sırasıyla $\{a,b,c,d,e,f,g,h,\imath,j,k,l,m\}$ ve $\{A,B,C,D,E,F,G,H,I,J,K,L,M\}$ ile gösterilmektedir. Burada $A=\{a,d,g,l\}$, $B=\{a,b,c,k\}$, $C=\{c,h,\imath,l\}$, $D=\{c,d,j,m\}$, $E=\{b,e,j,l\}$, $F=\{c,e,f,g\}$ $G=\{g,\imath,j,k\}$, $H=\{b,g,h,m\}$, $I=\{b,d,f,\imath\}$, $J=\{c,d,j,m\}$, $K=\{d,e,h,k\}$, $L=\{f,k,l,m\}$, $M=\{a,e,\imath,m\}$ dir. 1. adımda P nin noktaları üzerinde $(a\ \ 0,9)$, $(b\ \ 0,8)$, $(c\ \ 0,7)$, $(d\ \ 0,6)$, $(e\ \ 0,3)$, $(f\ \ 0,4)$, $(\ g\ \ 0,45)$, $(\ h\ \ 0,35)$, $(\imath\ \ 0,55)$, $(j\ \ 0,2)$, $(k\ \ 0,1)$, $(l\ \ 0,25)$ ve $(m\ \ 0,5)$ f-noktalarını alalım. Böylece 0,9 0,8 0,7 0,6 0,3 0,4 0,45 0,35 0,55 0,2 0,1 0, 25 ve 0,5 sırasıyla $a,b,c,d,e,f,g,h,\imath,j,k,l$ ve g taban noktalarının başlangıç değerleridir. İlk oluşumda A doğrusu üzerindeki $(a\ 0,9)$, $(d\ 0,6)$, $(g\ 0,45)$, $(l\ 0,25)$ fiber noktaları taban doğrusu A olan üçer üçer farklı fiber doğrular oluşturur. Bu fiber doğruların üyelik dereceleri minimum operatörü kullanılarak $\{0,25\ \ 0,45\ \ 0,65\}$ olarak bulunur. Benzer şekilde tabanı B doğrusu olan fiber doğruların üyelik dereceleri $\{0,1\ \ 0,7\ \ 0,8\}$, tabanı C doğrusu olan fiber doğruların üyelik dereceleri $\{0,25\ \ 0,35\ \ 0,55\}$, tabanı D doğrusu olan fiber doğruların üyelik dereceleri $\{0,2\ \ 0,5\ \ 0,6\}$, tabanı E doğrusu olan fiber doğruların üyelik dereceleri $\{0,2\ \ 0,25\ \ 0,3\}$, tabanı F doğrusu olan fiber doğruların üyelik dereceleri $\{0,3\ \ 0,4\ \ 0,45\}$,tabanı G doğrusu olan fiber doğruların üyelik dereceleri $\{0,1\ \ 0,2\ \ 0,45\}$,tabanı H doğrusu olan fiber doğruların üyelik dereceleri $\{0,35\ \ 0,45\ \ 0,5\}$, tabanı I doğrusu olan fiber doğruların üyelik dereceleri $\{0,4\ \ 0,55\ \ 0,6\}$, tabanı J doğrusu olan fiber doğruların üyelik dereceleri $\{0,2\ \ 0,5\ \ 0,6\}$, tabanı K doğrusu olan fiber doğruların üyelik dereceleri $\{0,1\ \ 0,3\ \ 0,35\}$, tabanı L doğrusu olan fiber doğruların üyelik dereceleri $\{0,1\ \ 0,25\ \ 0,40\}$, tabanı M doğrusu olan fiber doğruların üyelik dereceleri $\{0,3\ \ 0,5\ \ 0,55\}$ dır. 2. aşamada tabanı $a,b,c,d,e,f,g,h,\imath,j,k,l,m$ noktaları olan fiber noktaların aldığı üyelik dereceleri sırasıyla

$\{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,5\ \ \ 0,55\ \ 0,6\}$,

$\{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,5\ \ 0,55\ \ 0,6\}$,

$\{0,1\ \ 0,2\ \ \ 0,25\ \ 0,3\ \ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,5\ \ 0,55\ \ 0,6\}$,

$\{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,5\ \ 0,55\}$,

$\{0,1\ \ 0,2\ \ 0,25,\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\}$,

$\{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\}$,

$\{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\}$,

$\{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\}$,

$\{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,5\}$,

$\{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\}$,

$\{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\}$,

$\{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,5\ \ 0,55\}$,

$\{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,350,4\ \ 0,45\ \ 0,5\ \ 0,6\}$.

3. aşamada yeni fiber nokta ve doğru oluşmaz. Böylece nokta ve doğruları aşağıda verilen mono-point-generated fiber projektif düzlemi oluşturulmuş olur.

$a = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4,\ \ 0,45\ \ 0,5\ \ 0,55\ \ 0,6\ \ 0,9\}$,

$b = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,5\ \ 0,55\ \ 0,6\ \ 0,8\}$,

$c = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,5\ \ 0,55\ \ 0,6\ \ 0,7\}$,

$d = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,5\ \ 0,55\ \ 0,6\}$,

$e = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\}$,

$f = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\}$,

$g = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\}$,

$h = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\}$,

$i = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,5\ \ 0,55\}$,

$j = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\}$,

$k = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\}$,

$l = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,55\}$,

$m = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,5\ \ 0,6\}$,ve

$A = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,5\ \ 0,55\ \ 0,6\}$,

$B = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,5\ \ 0,55\ \ 0,6\ \ 0,7\ \ 0,8\}$,

$C = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,55\}$,

$D = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,5\ \ 0,55\ \ 0,6\}$,

$E = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,55\}$,

$F = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\}$,

$G = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\}$,

$H = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,5\ \ 0,55\ \ 0,6\}$,

$I = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,5\ \ 0,55\ \ 0,6\}$,

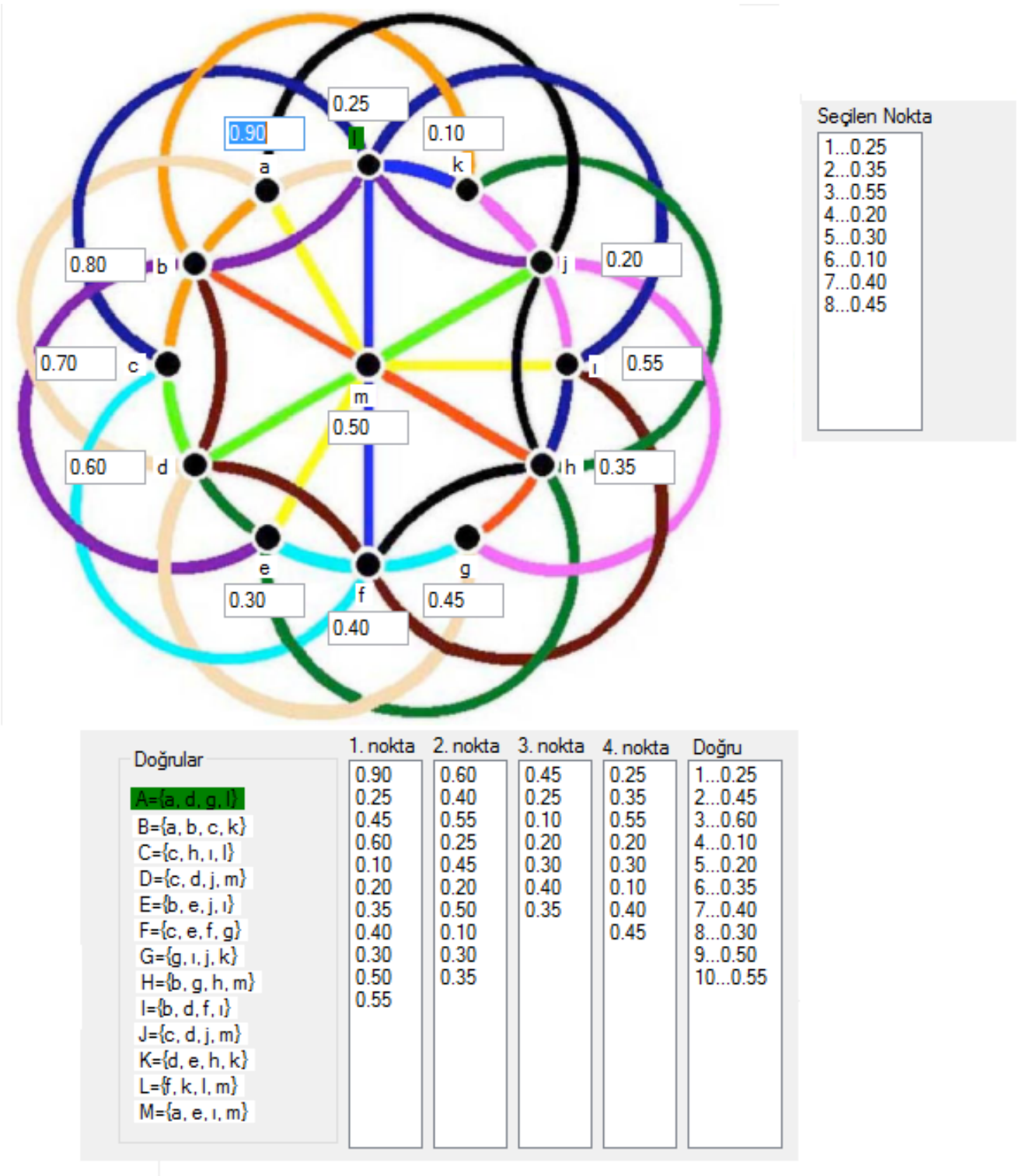
$J = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,55\}$,

$K = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\}$,

$L = \{0,1\ \ 0,2\ \ 0,25\ \ 0,3\ \ 0,35\ \ 0,4\ \ 0,45\ \ 0,55\}$,

$$M = \{0,1 \;\; 0,2 \;\; 0,25 \;\; 0,3 \;\; 0,35 \;\; 0,4 \;\; 0,45 \;\; 0,55 \;\; 0,6\},$$

Şekil 4.1 de program görüntüsüne yer verilmiştir. Örnek 4.1 de yer alan başlangıç değerleri programa girilmiş ve *l* noktası seçilerek noktanın almış olduğu üyelik dereceleri ”seçilen nokta” kısmında belirmiştir. Yine *a, d, g, l* noktalarının üzerinde bulunduğu *A* doğrusu seçilmiş ve ”1. nokta, 2. nokta, 3. nokta, 4. nokta” kısımlarında sırasıyla *a, d, g, l* noktalarının üyelik dereceleri ve bu noktalar yardımıyla ”doğru” kısmında *A* doğrusuna ait üyelik dereceleri belirmiştir.



Şekil 4.1. PG(2,3) de *A* Dogrusu Üyelik Dereceleri

Yukarıda,PG(2,3) düzleminin fiberleştirme örneği için C# kodu aşağıdaki çizelgede verilmiştir.

Kod yardımıyla PG(2,3) düzlemine ait tüm nokta ve doğruların üyelik dereceleri keyfi başlangıç değerlerine göre belirlenebilir.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
namespace Projectivator
{
public partial class Form2 : Form
{
public int[,] listFuzzyPoints_buffer = new int[50, 1];
public int listFuzzyPoints_bufferEndPoint = 0;
public int[,] fibringLines1_buffer = new int[150, 1];
public int fibringLines1_bufferEndPoint = 0;
public int[,] fibringLines2_buffer = new int[150, 1];
public int fibringLines2_bufferEndPoint = 0;
public int[,] fibringLines3_buffer = new int[150, 1];
public int fibringLines3_bufferEndPoint = 0;
public int[,] fibringLines4_buffer = new int[150, 1];
public int fibringLines4_bufferEndPoint = 0;
public int[,] fibringLines5_buffer = new int[5500, 1];
public int fibringLines5_bufferEndPoint = 0;
public int[,] fibringLines6_buffer = new int[1500, 1];
public int fibringLines6_bufferEndPoint = 0;
public int function = 0;
public Form2()
{
InitializeComponent();
}
private void button1_Click(object sender, EventArgs e)
{
this.Close();
}
private void lbl1_Click(object sender, EventArgs e)
{
lstBox13Points.Items.Clear();
if (txt1.Text == "" | txt3.Text == "" | txt4.Text == "" | txt5.Text == "" | txt6.Text == ""
| txt7.Text == "" | txt2.Text == "" | txt8.Text == "" | txt9.Text == "" | txt10.Text == "" |
txt11.Text == "" | txt12.Text "" | txt13.Text == "")
{
MessageBox.Show("lütfen bütün noktaları giriniz...");
}
else
{
int lenghtPoint1 = txt1.Text.Length;
int lenghtPoint2 = txt2.Text.Length;
int lenghtPoint3 = txt3.Text.Length;
```

```
int lenghtPoint4 = txt4.Text.Length;
int lenghtPoint5 = txt5.Text.Length;
int lenghtPoint6 = txt6.Text.Length;
int lenghtPoint7 = txt7.Text.Length;
int lenghtPoint8 = txt8.Text.Length;
int lenghtPoint9 = txt9.Text.Length;
int lenghtPoint10 = txt10.Text.Length;
int lenghtPoint11 = txt11.Text.Length;
int lenghtPoint12 = txt12.Text.Length;
int lenghtPoint13 = txt13.Text.Length;
string strpoint1 = txt1.Text.Substring(2, lenghtPoint1 - 2);
string strpoint2 = txt2.Text.Substring(2, lenghtPoint2 - 2);
string strpoint3 = txt3.Text.Substring(2, lenghtPoint3 - 2);
string strpoint4 = txt4.Text.Substring(2, lenghtPoint4 - 2);
string strpoint5 = txt5.Text.Substring(2, lenghtPoint5 - 2);
string strpoint6 = txt6.Text.Substring(2, lenghtPoint6 - 2);
string strpoint7 = txt7.Text.Substring(2, lenghtPoint7 - 2);
string strpoint8 = txt8.Text.Substring(2, lenghtPoint8 - 2);
string strpoint9 = txt9.Text.Substring(2, lenghtPoint9 - 2);
string strpoint10 = txt10.Text.Substring(2, lenghtPoint10- 2);
string strpoint11 = txt11.Text.Substring(2, lenghtPoint11- 2);
string strpoint12 = txt12.Text.Substring(2, lenghtPoint12- 2);
string strpoint13 = txt13.Text.Substring(2, lenghtPoint13- 2);
int point1 = Convert.ToInt16(strpoint1);
int point2 = Convert.ToInt16(strpoint2);
int point3 = Convert.ToInt16(strpoint3);
int point4 = Convert.ToInt16(strpoint4);
int point5 = Convert.ToInt16(strpoint5);
int point6 = Convert.ToInt16(strpoint6);
int point7 = Convert.ToInt16(strpoint7);
int point8 = Convert.ToInt16(strpoint8);
int point9 = Convert.ToInt16(strpoint9);
int point10 = Convert.ToInt16(strpoint10);
int point11 = Convert.ToInt16(strpoint11);
int point12 = Convert.ToInt16(strpoint12);
int point13 = Convert.ToInt16(strpoint13);
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = point1;
listFuzzyPoints_bufferEndPoint++;
if (cmBoxOperator.Text==”minimum”) function=1;
if (cmBoxOperator.Text == ”maximum”) function = 2;
if (function == 1)
{
int myFirstLineFirstFibredPoint = fibringFirstPoint(point1, point4, point7, point12);
int myFirstLineSecondFibredPoint = fibringSecondPoint(point1, point4, point7,
point12);
int myFirstLineThirdFibredPoint = fibringThirdPoint(point1, point4, point7, point12);
int mySecondLineFirstFibredPoint = fibringFirstPoint(point1, point2, point3, point11);
int mySecondLineSecondFibredPoint = fibringSecondPoint(point1, point2, point3,
point11);
```

```
int mySecondLineThirdFibredPoint = fibringThirdPoint(point1, point2, point3, point11);
int myThirdLineFirstFibredPoint = fibringFirstPoint(point1, point6, point8, point10);
int myThirdLineSecondFibredPoint = fibringSecondPoint(point1, point6, point8,
point10);
int myThirdLineThirdFibredPoint = fibringThirdPoint(point1, point6, point8, point10);
int myFourthLineFirstFibredPoint = fibringFirstPoint(point1, point5, point9, point13);
int myFourthLineSecondFibredPoint = fibringSecondPoint(point1, point5, point9,
point13);
int myFourthLineThirdFibredPoint = fibringThirdPoint(point1, point5, point9, point13);
fibringPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirst-
LineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibred-
Point, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLi-
neSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint,
myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
if (function == 2)
{
int myFirstLineFirstFibredPoint = fibringMaxFirstPoint(point1, point4, point7, point12);
int myFirstLineSecondFibredPoint = fibringMaxSecondPoint(point1, point4, point7,
point12);
int myFirstLineThirdFibredPoint = fibringMaxThirdPoint(point1, point4, point7,
point12);
int mySecondLineFirstFibredPoint = fibringMaxFirstPoint(point1, point2, point3,
point11);
int mySecondLineSecondFibredPoint = fibringMaxSecondPoint(point1, point2, point3,
point11);
int mySecondLineThirdFibredPoint = fibringMaxThirdPoint(point1, point2, point3,
point11);
int myThirdLineFirstFibredPoint = fibringMaxFirstPoint(point1, point6, point8,
point10);
int myThirdLineSecondFibredPoint = fibringMaxSecondPoint(point1, point6, point8,
point10);
int myThirdLineThirdFibredPoint = fibringMaxThirdPoint(point1, point6, point8,
point10);
int myFourthLineFirstFibredPoint = fibringMaxFirstPoint(point1, point5, point9,
point13);
int myFourthLineSecondFibredPoint = fibringMaxSecondPoint(point1, point5, point9,
point13);
int myFourthLineThirdFibredPoint = fibringMaxThirdPoint(point1, point5, point9,
point13);
fibringMaxPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, my-
FirstLineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFi-
bredPoint, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLi-
neSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint,
myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
}
}
private void lbl2_Click(object sender, EventArgs e)
{
```

```
listFuzzyPoints_bufferEndPoint = 0;
if (txt1.Text == "" | txt3.Text == "" | txt4.Text == "" | txt5.Text == "" | txt6.Text == ""
| txt7.Text == "" | txt2.Text == "" | txt8.Text == "" | txt9.Text == "" | txt10.Text == "" |
txt11.Text == "" | txt12.Text == "" | txt13.Text == "")
{
MessageBox.Show("lütfen bütün noktaları giriniz...");
}
else
{
int lenghtPoint1 = txt1.Text.Length;
int lenghtPoint2 = txt2.Text.Length;
int lenghtPoint3 = txt3.Text.Length;
int lenghtPoint4 = txt4.Text.Length;
int lenghtPoint5 = txt5.Text.Length;
int lenghtPoint6 = txt6.Text.Length;
int lenghtPoint7 = txt7.Text.Length;
int lenghtPoint8 = txt8.Text.Length;
int lenghtPoint9 = txt9.Text.Length;
int lenghtPoint10 = txt10.Text.Length;
int lenghtPoint11 = txt11.Text.Length;
int lenghtPoint12 = txt12.Text.Length;
int lenghtPoint13 = txt13.Text.Length;
string strpoint1 = txt1.Text.Substring(2, lenghtPoint1 - 2);
string strpoint2 = txt2.Text.Substring(2, lenghtPoint2 - 2);
string strpoint3 = txt3.Text.Substring(2, lenghtPoint3 - 2);
string strpoint4 = txt4.Text.Substring(2, lenghtPoint4 - 2);
string strpoint5 = txt5.Text.Substring(2, lenghtPoint5 - 2);
string strpoint6 = txt6.Text.Substring(2, lenghtPoint6 - 2);
string strpoint7 = txt7.Text.Substring(2, lenghtPoint7 - 2);
string strpoint8 = txt8.Text.Substring(2, lenghtPoint8 - 2);
string strpoint9 = txt9.Text.Substring(2, lenghtPoint9 - 2);
string strpoint10 = txt10.Text.Substring(2, lenghtPoint10 - 2);
string strpoint11 = txt11.Text.Substring(2, lenghtPoint11 - 2);
string strpoint12 = txt12.Text.Substring(2, lenghtPoint12 - 2);
string strpoint13 = txt13.Text.Substring(2, lenghtPoint13 - 2);
int point1 = Convert.ToInt16(strpoint1);
int point2 = Convert.ToInt16(strpoint2);
int point3 = Convert.ToInt16(strpoint3);
int point4 = Convert.ToInt16(strpoint4);
int point5 = Convert.ToInt16(strpoint5);
int point6 = Convert.ToInt16(strpoint6);
int point7 = Convert.ToInt16(strpoint7);
int point8 = Convert.ToInt16(strpoint8);
int point9 = Convert.ToInt16(strpoint9);
int point10 = Convert.ToInt16(strpoint10);
int point11 = Convert.ToInt16(strpoint11);
int point12 = Convert.ToInt16(strpoint12);
int point13 = Convert.ToInt16(strpoint13);
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = point2;
```

```
listFuzzyPoints_bufferEndPoint++;
if (cmBoxOperator.Text==”minimum”) function=1;
if (cmBoxOperator.Text == ”maximum”) function = 2;
if (function == 1)
{
int myFirstLineFirstFibredPoint = fibringFirstPoint(point2, point5, point10, point12);
int myFirstLineSecondFibredPoint = fibringSecondPoint(point2, point5, point10, point12);
int myFirstLineThirdFibredPoint = fibringThirdPoint(point2, point5, point10, point12);
int mySecondLineFirstFibredPoint = fibringFirstPoint(point1, point2, point3, point11);
int mySecondLineSecondFibredPoint = fibringSecondPoint(point1, point2, point3, point11);
int mySecondLineThirdFibredPoint = fibringThirdPoint(point1, point2, point3, point11);
int myThirdLineFirstFibredPoint = fibringFirstPoint(point2, point4, point6, point9);
int myThirdLineSecondFibredPoint = fibringSecondPoint(point2, point4, point6, point9);
int myThirdLineThirdFibredPoint = fibringThirdPoint(point2, point4, point6, point9);
int myFourthLineFirstFibredPoint = fibringFirstPoint(point2, point7, point8, point13);
int myFourthLineSecondFibredPoint = fibringSecondPoint(point2, point7, point8, point13);
int myFourthLineThirdFibredPoint = fibringThirdPoint(point2, point7, point8, point13);
fibringPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirst-
LineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibred-
Point, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLi-
neSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint,
myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
if (function == 2)
{
int myFirstLineFirstFibredPoint = fibringMaxFirstPoint(point2, point5, point10, point12);
int myFirstLineSecondFibredPoint = fibringMaxSecondPoint(point2, point5, point10, point12);
int myFirstLineThirdFibredPoint = fibringMaxThirdPoint(point2, point5, point10, point12);
int mySecondLineFirstFibredPoint = fibringMaxFirstPoint(point1, point2, point3, point11);
int mySecondLineSecondFibredPoint = fibringMaxSecondPoint(point1, point2, point3, point11);
int mySecondLineThirdFibredPoint = fibringMaxThirdPoint(point1, point2, point3, point11);
int myThirdLineFirstFibredPoint = fibringMaxFirstPoint(point2, point4, point6, point9);
int myThirdLineSecondFibredPoint = fibringMaxSecondPoint(point2, point4, point6, point9);
int myThirdLineThirdFibredPoint = fibringMaxThirdPoint(point2, point4, point6, point9);
int myFourthLineFirstFibredPoint = fibringMaxFirstPoint(point2, point7, point8, point13);
```

```
int myFourthLineSecondFibredPoint = fibringMaxSecondPoint(point2, point7, point8, point13);
int myFourthLineThirdFibredPoint = fibringMaxThirdPoint(point2, point7, point8, point13);
fibringMaxPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirstLineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibredPoint, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLineSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint, myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
}
}
private void lbl3_Click(object sender, EventArgs e)
{
listFuzzyPoints_bufferEndPoint = 0;
if (txt1.Text == "" | txt3.Text == "" | txt4.Text == "" | txt5.Text == "" | txt6.Text == "" | txt7.Text == "" | txt2.Text == "" | txt8.Text == "" | txt9.Text == "" | txt10.Text == "" | txt11.Text == "" | txt12.Text == "" | txt13.Text == "")
{
MessageBox.Show("lütfen bütün noktaları giriniz...");
}
else
{
int lenghtPoint1 = txt1.Text.Length;
int lenghtPoint2 = txt2.Text.Length;
int lenghtPoint3 = txt3.Text.Length;
int lenghtPoint4 = txt4.Text.Length;
int lenghtPoint5 = txt5.Text.Length;
int lenghtPoint6 = txt6.Text.Length;
int lenghtPoint7 = txt7.Text.Length;
int lenghtPoint8 = txt8.Text.Length;
int lenghtPoint9 = txt9.Text.Length;
int lenghtPoint10 = txt10.Text.Length;
int lenghtPoint11 = txt11.Text.Length;
int lenghtPoint12 = txt12.Text.Length;
int lenghtPoint13 = txt13.Text.Length;
string strpoint1 = txt1.Text.Substring(2, lenghtPoint1 - 2);
string strpoint2 = txt2.Text.Substring(2, lenghtPoint2 - 2);
string strpoint3 = txt3.Text.Substring(2, lenghtPoint3 - 2);
string strpoint4 = txt4.Text.Substring(2, lenghtPoint4 - 2);
string strpoint5 = txt5.Text.Substring(2, lenghtPoint5 - 2);
string strpoint6 = txt6.Text.Substring(2, lenghtPoint6 - 2);
string strpoint7 = txt7.Text.Substring(2, lenghtPoint7 - 2);
string strpoint8 = txt8.Text.Substring(2, lenghtPoint8 - 2);
string strpoint9 = txt9.Text.Substring(2, lenghtPoint9 - 2);
string strpoint10 = txt10.Text.Substring(2, lenghtPoint10 - 2);
string strpoint11 = txt11.Text.Substring(2, lenghtPoint11 - 2);
string strpoint12 = txt12.Text.Substring(2, lenghtPoint12 - 2);
string strpoint13 = txt13.Text.Substring(2, lenghtPoint13 - 2);
```

```
int point1 = Convert.ToInt16(strpoint1);
int point2 = Convert.ToInt16(strpoint2);
int point3 = Convert.ToInt16(strpoint3);
int point4 = Convert.ToInt16(strpoint4);
int point5 = Convert.ToInt16(strpoint5);
int point6 = Convert.ToInt16(strpoint6);
int point7 = Convert.ToInt16(strpoint7);
int point8 = Convert.ToInt16(strpoint8);
int point9 = Convert.ToInt16(strpoint9);
int point10 = Convert.ToInt16(strpoint10);
int point11 = Convert.ToInt16(strpoint11);
int point12 = Convert.ToInt16(strpoint12);
int point13 = Convert.ToInt16(strpoint13);
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = point3;
listFuzzyPoints_bufferEndPoint++;
if (cmBoxOperator.Text==”minimum”) function=1;
if (cmBoxOperator.Text == ”maximum”) function = 2;
if (function == 1)
{
int myFirstLineFirstFibredPoint = fibringFirstPoint(point1, point8, point9, point12);
int myFirstLineSecondFibredPoint = fibringSecondPoint(point1, point8, point9,
point12);
int myFirstLineThirdFibredPoint = fibringThirdPoint(point1, point8, point9, point12);
int mySecondLineFirstFibredPoint = fibringFirstPoint(point1, point2, point3, point11);
int mySecondLineSecondFibredPoint = fibringSecondPoint(point1, point2, point3,
point11);
int mySecondLineThirdFibredPoint = fibringThirdPoint(point1, point2, point3, point11);
int myThirdLineFirstFibredPoint = fibringFirstPoint(point3, point5, point6, point7);
int myThirdLineSecondFibredPoint = fibringSecondPoint(point3, point5, point6, point7);
int myThirdLineThirdFibredPoint = fibringThirdPoint(point3, point5, point6, point7);
int myFourthLineFirstFibredPoint = fibringFirstPoint(point3, point4, point10, point13);
int myFourthLineSecondFibredPoint = fibringSecondPoint(point3, point4, point10,
point13);
int myFourthLineThirdFibredPoint = fibringThirdPoint(point3, point4, point10,
point13);
fibringPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirst-
LineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibred-
Point, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLi-
neSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint,
myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
if (function == 2)
{
int myFirstLineFirstFibredPoint = fibringMaxFirstPoint(point1, point8, point9, point12);
int myFirstLineSecondFibredPoint = fibringMaxSecondPoint(point1, point8, point9,
point12);
int myFirstLineThirdFibredPoint = fibringMaxThirdPoint(point1, point8, point9,
point12);
```

```
int mySecondLineFirstFibredPoint = fibringMaxFirstPoint(point1, point2, point3, point11);
int mySecondLineSecondFibredPoint = fibringMaxSecondPoint(point1, point2, point3, point11);
int mySecondLineThirdFibredPoint = fibringMaxThirdPoint(point1, point2, point3, point11);
int myThirdLineFirstFibredPoint = fibringMaxFirstPoint(point3, point5, point6, point7);
int myThirdLineSecondFibredPoint = fibringMaxSecondPoint(point3, point5, point6, point7);
int myThirdLineThirdFibredPoint = fibringMaxThirdPoint(point3, point5, point6, point7);
int myFourthLineFirstFibredPoint = fibringMaxFirstPoint(point3, point4, point10, point13);
int myFourthLineSecondFibredPoint = fibringMaxSecondPoint(point3, point4, point10, point13);
int myFourthLineThirdFibredPoint = fibringMaxThirdPoint(point3, point4, point10, point13);
fibringMaxPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirstLineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibredPoint, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLineSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint, myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
}
}
private void lbl4_Click(object sender, EventArgs e) {
listFuzzyPoints_bufferEndPoint = 0;
if (txt1.Text == "" | txt3.Text == "" | txt4.Text == "" | txt5.Text == "" | txt6.Text == "" | txt7.Text == "" | txt2.Text == "" | txt8.Text == "" | txt9.Text == "" | txt10.Text == "" | txt11.Text == "" | txt12.Text == "" | txt13.Text == "")
{
MessageBox.Show("lütfen bütün noktaları giriniz...");
}
else
int lenghtPoint1 = txt1.Text.Length;
int lenghtPoint2 = txt2.Text.Length;
int lenghtPoint3 = txt3.Text.Length;
int lenghtPoint4 = txt4.Text.Length;
int lenghtPoint5 = txt5.Text.Length;
int lenghtPoint6 = txt6.Text.Length;
int lenghtPoint7 = txt7.Text.Length;
int lenghtPoint8 = txt8.Text.Length;
int lenghtPoint9 = txt9.Text.Length;
int lenghtPoint10 = txt10.Text.Length;
int lenghtPoint11 = txt11.Text.Length;
int lenghtPoint12 = txt12.Text.Length;
int lenghtPoint13 = txt13.Text.Length;
string strpoint1 = txt1.Text.Substring(2, lenghtPoint1 - 2);
string strpoint2 = txt2.Text.Substring(2, lenghtPoint2 - 2);
```

```
string strpoint3 = txt3.Text.Substring(2, lenghtPoint3 - 2);
string strpoint4 = txt4.Text.Substring(2, lenghtPoint4 - 2);
string strpoint5 = txt5.Text.Substring(2, lenghtPoint5 - 2);
string strpoint6 = txt6.Text.Substring(2, lenghtPoint6 - 2);
string strpoint7 = txt7.Text.Substring(2, lenghtPoint7 - 2);
string strpoint8 = txt8.Text.Substring(2, lenghtPoint8 - 2);
string strpoint9 = txt9.Text.Substring(2, lenghtPoint9 - 2);
string strpoint10 = txt10.Text.Substring(2, lenghtPoint10 - 2);
string strpoint11 = txt11.Text.Substring(2, lenghtPoint11 - 2);
string strpoint12 = txt12.Text.Substring(2, lenghtPoint12 - 2);
string strpoint13 = txt13.Text.Substring(2, lenghtPoint13 - 2);
int point1 = Convert.ToInt16(strpoint1);
int point2 = Convert.ToInt16(strpoint2);
int point3 = Convert.ToInt16(strpoint3);
int point4 = Convert.ToInt16(strpoint4);
int point5 = Convert.ToInt16(strpoint5);
int point6 = Convert.ToInt16(strpoint6);
int point7 = Convert.ToInt16(strpoint7);
int point8 = Convert.ToInt16(strpoint8);
int point9 = Convert.ToInt16(strpoint9);
int point10 = Convert.ToInt16(strpoint10);
int point11 = Convert.ToInt16(strpoint11);
int point12 = Convert.ToInt16(strpoint12);
int point13 = Convert.ToInt16(strpoint13);
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = point4;
listFuzzyPoints_bufferEndPoint++;
if (cmBoxOperator.Text==”minimum”) function=1;
if (cmBoxOperator.Text == ”maximum”) function = 2;
if (function == 1)
{
int myFirstLineFirstFibredPoint = fibringFirstPoint(point2, point4, point6, point9);
int myFirstLineSecondFibredPoint = fibringSecondPoint(point2, point4, point6, point9);
int myFirstLineThirdFibredPoint = fibringThirdPoint(point2, point4, point6, point9);
int mySecondLineFirstFibredPoint = fibringFirstPoint(point1, point4, point7, point12);
int mySecondLineSecondFibredPoint = fibringSecondPoint(point1, point4, point7,
point12);
int mySecondLineThirdFibredPoint = fibringThirdPoint(point1, point4, point7, point12);
int myThirdLineFirstFibredPoint = fibringFirstPoint(point3, point4, point10, point13);
int myThirdLineSecondFibredPoint = fibringSecondPoint(point3, point4, point10,
point13);
int myThirdLineThirdFibredPoint = fibringThirdPoint(point3, point4, point10, point13);
int myFourthLineFirstFibredPoint = fibringFirstPoint(point4, point5, point8, point11);
int myFourthLineSecondFibredPoint = fibringSecondPoint(point4, point5, point8,
point11);
int myFourthLineThirdFibredPoint = fibringThirdPoint(point4, point5, point8, point11);
```

```
fibringPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirst-
LineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibred-
Point, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLi-
neSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint,
myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
if (function == 2)
{
int myFirstLineFirstFibredPoint = fibringMaxFirstPoint(point2, point4, point6, point9);
int myFirstLineSecondFibredPoint = fibringMaxSecondPoint(point2, point4, point6,
point9);
int myFirstLineThirdFibredPoint = fibringMaxThirdPoint(point2, point4, point6,
point9);
int mySecondLineFirstFibredPoint = fibringMaxFirstPoint(point1, point4, point7,
point12);
int mySecondLineSecondFibredPoint = fibringMaxSecondPoint(point1, point4, point7,
point12);
int mySecondLineThirdFibredPoint = fibringMaxThirdPoint(point1, point4, point7,
point12);
int myThirdLineFirstFibredPoint = fibringMaxFirstPoint(point3, point4, point10,
point13);
int myThirdLineSecondFibredPoint = fibringMaxSecondPoint(point3, point4, point10,
point13);
int myThirdLineThirdFibredPoint = fibringMaxThirdPoint(point3, point4, point10,
point13);
int myFourthLineFirstFibredPoint = fibringMaxFirstPoint(point4, point5, point8,
point11);
int myFourthLineSecondFibredPoint = fibringMaxSecondPoint(point4, point5, point8,
point11);
int myFourthLineThirdFibredPoint = fibringMaxThirdPoint(point4, point5, point8,
point11);
fibringMaxPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, my-
FirstLineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFi-
bredPoint, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLi-
neSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint,
myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
}
}
private void lbl5_Click(object sender, EventArgs e)
{
lstBox13Points.Items.Clear();
listFuzzyPoints_bufferEndPoint = 0;
if (txt1.Text == "" | txt3.Text == "" | txt4.Text == "" | txt5.Text == "" | txt6.Text == ""
| txt7.Text == "" | txt2.Text == "" | txt8.Text == "" | txt9.Text == "" | txt10.Text == "" |
txt11.Text == "" | txt12.Text == "" | txt13.Text == "")
{
MessageBox.Show("lütfen bütün noktaları giriniz...");
```

```
}
else
{
int lenghtPoint1 = txt1.Text.Length;
int lenghtPoint2 = txt2.Text.Length;
int lenghtPoint3 = txt3.Text.Length;
int lenghtPoint4 = txt4.Text.Length;
int lenghtPoint5 = txt5.Text.Length;
int lenghtPoint6 = txt6.Text.Length;
int lenghtPoint7 = txt7.Text.Length;
int lenghtPoint8 = txt8.Text.Length;
int lenghtPoint9 = txt9.Text.Length;
int lenghtPoint10 = txt10.Text.Length;
int lenghtPoint11 = txt11.Text.Length;
int lenghtPoint12 = txt12.Text.Length;
int lenghtPoint13 = txt13.Text.Length;
string strpoint1 = txt1.Text.Substring(2, lenghtPoint1 - 2);
string strpoint2 = txt2.Text.Substring(2, lenghtPoint2 - 2);
string strpoint3 = txt3.Text.Substring(2, lenghtPoint3 - 2);
string strpoint4 = txt4.Text.Substring(2, lenghtPoint4 - 2);
string strpoint5 = txt5.Text.Substring(2, lenghtPoint5 - 2);
string strpoint6 = txt6.Text.Substring(2, lenghtPoint6 - 2);
string strpoint7 = txt7.Text.Substring(2, lenghtPoint7 - 2);
string strpoint8 = txt8.Text.Substring(2, lenghtPoint8 - 2);
string strpoint9 = txt9.Text.Substring(2, lenghtPoint9 - 2);
string strpoint10 = txt10.Text.Substring(2, lenghtPoint10 - 2);
string strpoint11 = txt11.Text.Substring(2, lenghtPoint11 - 2);
string strpoint12 = txt12.Text.Substring(2, lenghtPoint12 - 2);
string strpoint13 = txt13.Text.Substring(2, lenghtPoint13 - 2);
int point1 = Convert.ToInt16(strpoint1);
int point2 = Convert.ToInt16(strpoint2);
int point3 = Convert.ToInt16(strpoint3);
int point4 = Convert.ToInt16(strpoint4);
int point5 = Convert.ToInt16(strpoint5);
int point6 = Convert.ToInt16(strpoint6);
int point7 = Convert.ToInt16(strpoint7);
int point8 = Convert.ToInt16(strpoint8);
int point9 = Convert.ToInt16(strpoint9);
int point10 = Convert.ToInt16(strpoint10);
int point11 = Convert.ToInt16(strpoint11);
int point12 = Convert.ToInt16(strpoint12);
int point13 = Convert.ToInt16(strpoint13);
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = point5;
listFuzzyPoints_bufferEndPoint++;
if (cmBoxOperator.Text==”minimum”) function=1;
if (cmBoxOperator.Text == ”maximum”) function = 2;
if (function == 1)
{
```

```
int myFirstLineFirstFibredPoint = fibringFirstPoint(point2, point5, point10, point12);
int myFirstLineSecondFibredPoint = fibringSecondPoint(point2, point5, point10, point12);
int myFirstLineThirdFibredPoint = fibringThirdPoint(point2, point5, point10, point12);
int mySecondLineFirstFibredPoint = fibringFirstPoint(point1, point5, point9, point13);
int mySecondLineSecondFibredPoint = fibringSecondPoint(point1, point5, point9, point13);
int mySecondLineThirdFibredPoint = fibringThirdPoint(point1, point5, point9, point13);
int myThirdLineFirstFibredPoint = fibringFirstPoint(point3, point5, point6, point7);
int myThirdLineSecondFibredPoint = fibringSecondPoint(point3, point5, point6, point7);
int myThirdLineThirdFibredPoint = fibringThirdPoint(point3, point5, point6, point7);
int myFourthLineFirstFibredPoint = fibringFirstPoint(point4, point5, point8, point11);
int myFourthLineSecondFibredPoint = fibringSecondPoint(point4, point5, point8, point11);
int myFourthLineThirdFibredPoint = fibringThirdPoint(point4, point5, point8, point11);
fibringPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirstLineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibredPoint, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLineSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint, myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
if (function == 2)
{
int myFirstLineFirstFibredPoint = fibringMaxFirstPoint(point2, point5, point10, point12);
int myFirstLineSecondFibredPoint = fibringMaxSecondPoint(point2, point5, point10, point12);
int myFirstLineThirdFibredPoint = fibringMaxThirdPoint(point2, point5, point10, point12);
int mySecondLineFirstFibredPoint = fibringMaxFirstPoint(point1, point5, point9, point13);
int mySecondLineSecondFibredPoint = fibringMaxSecondPoint(point1, point5, point9, point13);
int mySecondLineThirdFibredPoint = fibringMaxThirdPoint(point1, point5, point9, point13);
int myThirdLineFirstFibredPoint = fibringMaxFirstPoint(point3, point5, point6, point7);
int myThirdLineSecondFibredPoint = fibringMaxSecondPoint(point3, point5, point6, point7);
int myThirdLineThirdFibredPoint = fibringMaxThirdPoint(point3, point5, point6, point7);
int myFourthLineFirstFibredPoint = fibringMaxFirstPoint(point4, point5, point8, point11);
int myFourthLineSecondFibredPoint = fibringMaxSecondPoint(point4, point5, point8, point11);
int myFourthLineThirdFibredPoint = fibringMaxThirdPoint(point4, point5, point8, point11);
```

```
fibringMaxPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirstLineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibredPoint, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLineSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint, myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
}
}
private void lbl6_Click(object sender, EventArgs e)
{
lstBox13Points.Items.Clear();
listFuzzyPoints_bufferEndPoint = 0;
if (txt1.Text == "" | txt3.Text == "" | txt4.Text == "" | txt5.Text == "" | txt6.Text == "" | txt7.Text == "" | txt2.Text == "" | txt8.Text == "" | txt9.Text == "" | txt10.Text == "" | txt11.Text == "" | txt12.Text == "" | txt13.Text == "")
{
MessageBox.Show("lütfen bütün noktaları giriniz...");
}
else
{
int lenghtPoint1 = txt1.Text.Length;
int lenghtPoint2 = txt2.Text.Length;
int lenghtPoint3 = txt3.Text.Length;
int lenghtPoint4 = txt4.Text.Length;
int lenghtPoint5 = txt5.Text.Length;
int lenghtPoint6 = txt6.Text.Length;
int lenghtPoint7 = txt7.Text.Length;
int lenghtPoint8 = txt8.Text.Length;
int lenghtPoint9 = txt9.Text.Length;
int lenghtPoint10 = txt10.Text.Length;
int lenghtPoint11 = txt11.Text.Length;
int lenghtPoint12 = txt12.Text.Length;
int lenghtPoint13 = txt13.Text.Length;
string strpoint1 = txt1.Text.Substring(2, lenghtPoint1 - 2);
string strpoint2 = txt2.Text.Substring(2, lenghtPoint2 - 2);
string strpoint3 = txt3.Text.Substring(2, lenghtPoint3 - 2);
string strpoint4 = txt4.Text.Substring(2, lenghtPoint4 - 2);
string strpoint5 = txt5.Text.Substring(2, lenghtPoint5 - 2);
string strpoint6 = txt6.Text.Substring(2, lenghtPoint6 - 2);
string strpoint7 = txt7.Text.Substring(2, lenghtPoint7 - 2);
string strpoint8 = txt8.Text.Substring(2, lenghtPoint8 - 2);
string strpoint9 = txt9.Text.Substring(2, lenghtPoint9 - 2);
string strpoint10 = txt10.Text.Substring(2, lenghtPoint10 - 2);
string strpoint11 = txt11.Text.Substring(2, lenghtPoint11 - 2);
string strpoint12 = txt12.Text.Substring(2, lenghtPoint12 - 2);
string strpoint13 = txt13.Text.Substring(2, lenghtPoint13 - 2);
```

```
int point1 = Convert.ToInt16(strpoint1);
int point2 = Convert.ToInt16(strpoint2);
int point3 = Convert.ToInt16(strpoint3);
int point4 = Convert.ToInt16(strpoint4);
int point5 = Convert.ToInt16(strpoint5);
int point6 = Convert.ToInt16(strpoint6);
int point7 = Convert.ToInt16(strpoint7);
int point8 = Convert.ToInt16(strpoint8);
int point9 = Convert.ToInt16(strpoint9);
int point10 = Convert.ToInt16(strpoint10);
int point11 = Convert.ToInt16(strpoint11);
int point12 = Convert.ToInt16(strpoint12);
int point13 = Convert.ToInt16(strpoint13);
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = point6;
listFuzzyPoints_bufferEndPoint++;
if (cmBoxOperator.Text==”minimum”) function=1;
if (cmBoxOperator.Text == ”maximum”) function = 2;
if (function == 1)
{
int myFirstLineFirstFibredPoint = fibringFirstPoint(point2, point4, point6, point9);
int myFirstLineSecondFibredPoint = fibringSecondPoint(point2, point4, point6, point9);
int myFirstLineThirdFibredPoint = fibringThirdPoint(point2, point4, point6, point9);
int mySecondLineFirstFibredPoint = fibringFirstPoint(point1, point6, point8, point10);
int mySecondLineSecondFibredPoint = fibringSecondPoint(point1, point6, point8,
point10);
int mySecondLineThirdFibredPoint = fibringThirdPoint(point1, point6, point8, point10);
int myThirdLineFirstFibredPoint = fibringFirstPoint(point3, point5, point6, point7);
int myThirdLineSecondFibredPoint = fibringSecondPoint(point3, point5, point6, point7);
int myThirdLineThirdFibredPoint = fibringThirdPoint(point3, point5, point6, point7);
int myFourthLineFirstFibredPoint = fibringFirstPoint(point6, point11, point12, point13);
int myFourthLineSecondFibredPoint = fibringSecondPoint(point6, point11, point12,
point13);
int myFourthLineThirdFibredPoint = fibringThirdPoint(point6, point11, point12,
point13);
fibringPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirst-
LineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibred-
Point, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLi-
neSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint,
myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
if (function == 2)
{
int myFirstLineFirstFibredPoint = fibringMaxFirstPoint(point2, point4, point6, point9);
int myFirstLineSecondFibredPoint = fibringMaxSecondPoint(point2, point4, point6,
point9);
int myFirstLineThirdFibredPoint = fibringMaxThirdPoint(point2, point4, point6, point9);
```

```
int mySecondLineFirstFibredPoint = fibringMaxFirstPoint(point1, point6, point8, point10);
int mySecondLineSecondFibredPoint = fibringMaxSecondPoint(point1, point6, point8, point10);
int mySecondLineThirdFibredPoint = fibringMaxThirdPoint(point1, point6, point8, point10);
int myThirdLineFirstFibredPoint = fibringMaxFirstPoint(point3, point5, point6, point7);
int myThirdLineSecondFibredPoint = fibringMaxSecondPoint(point3, point5, point6, point7);
int myThirdLineThirdFibredPoint = fibringMaxThirdPoint(point3, point5, point6, point7);
int myFourthLineFirstFibredPoint = fibringMaxFirstPoint(point6, point11, point12, point13);
int myFourthLineSecondFibredPoint = fibringMaxSecondPoint(point6, point11, point12, point13);
int myFourthLineThirdFibredPoint = fibringMaxThirdPoint(point6, point11, point12, point13);
fibringMaxPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirstLineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibredPoint, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLineSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint, myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
}
}
private void lbl7_Click(object sender, EventArgs e)
{
lstBox13Points.Items.Clear();
listFuzzyPoints_bufferEndPoint = 0;
if (txt1.Text == "" | txt3.Text == "" | txt4.Text == "" | txt5.Text == "" | txt6.Text == "" | txt7.Text == "" | txt2.Text == "" | txt8.Text == "" | txt9.Text == "" | txt10.Text == "" | txt11.Text == "" | txt12.Text == "" | txt13.Text == "")
{
MessageBox.Show("lütfen bütün noktaları giriniz...");
}
else
{
int lenghtPoint1 = txt1.Text.Length;
int lenghtPoint2 = txt2.Text.Length;
int lenghtPoint3 = txt3.Text.Length;
int lenghtPoint4 = txt4.Text.Length;
int lenghtPoint5 = txt5.Text.Length;
int lenghtPoint6 = txt6.Text.Length;
int lenghtPoint7 = txt7.Text.Length;
int lenghtPoint8 = txt8.Text.Length;
int lenghtPoint9 = txt9.Text.Length;
int lenghtPoint10 = txt10.Text.Length;
int lenghtPoint11 = txt11.Text.Length;
int lenghtPoint12 = txt12.Text.Length;
int lenghtPoint13 = txt13.Text.Length;
```

```
string strpoint1 = txt1.Text.Substring(2, lenghtPoint1 - 2);
string strpoint2 = txt2.Text.Substring(2, lenghtPoint2 - 2);
string strpoint3 = txt3.Text.Substring(2, lenghtPoint3 - 2);
string strpoint4 = txt4.Text.Substring(2, lenghtPoint4 - 2);
string strpoint5 = txt5.Text.Substring(2, lenghtPoint5 - 2);
string strpoint6 = txt6.Text.Substring(2, lenghtPoint6 - 2);
string strpoint7 = txt7.Text.Substring(2, lenghtPoint7 - 2);
string strpoint8 = txt8.Text.Substring(2, lenghtPoint8 - 2);
string strpoint9 = txt9.Text.Substring(2, lenghtPoint9 - 2);
string strpoint10 = txt10.Text.Substring(2, lenghtPoint10 - 2);
string strpoint11 = txt11.Text.Substring(2, lenghtPoint11 - 2);
string strpoint12 = txt12.Text.Substring(2, lenghtPoint12 - 2);
string strpoint13 = txt13.Text.Substring(2, lenghtPoint13 - 2);
int point1 = Convert.ToInt16(strpoint1);
int point2 = Convert.ToInt16(strpoint2);
int point3 = Convert.ToInt16(strpoint3);
int point4 = Convert.ToInt16(strpoint4);
int point5 = Convert.ToInt16(strpoint5);
int point6 = Convert.ToInt16(strpoint6);
int point7 = Convert.ToInt16(strpoint7);
int point8 = Convert.ToInt16(strpoint8);
int point9 = Convert.ToInt16(strpoint9);
int point10 = Convert.ToInt16(strpoint10);
int point11 = Convert.ToInt16(strpoint11);
int point12 = Convert.ToInt16(strpoint12);
int point13 = Convert.ToInt16(strpoint13);
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = point7;
listFuzzyPoints_bufferEndPoint++;
if (cmBoxOperator.Text==”minimum”) function=1;
if (cmBoxOperator.Text == ”maximum”) function = 2;
if (function == 1)
{
int myFirstLineFirstFibredPoint = fibringFirstPoint(point1, point4, point7, point12);
int myFirstLineSecondFibredPoint = fibringSecondPoint(point1, point4, point7,
point12);
int myFirstLineThirdFibredPoint = fibringThirdPoint(point1, point4, point7, point12);
int mySecondLineFirstFibredPoint = fibringFirstPoint(point7, point9, point11, point10);
int mySecondLineSecondFibredPoint = fibringSecondPoint(point7, point9, point11,
point10);
int mySecondLineThirdFibredPoint = fibringThirdPoint(point7, point9, point11,
point10);
int myThirdLineFirstFibredPoint = fibringFirstPoint(point3, point5, point6, point7);
int myThirdLineSecondFibredPoint = fibringSecondPoint(point3, point5, point6, point7);
int myThirdLineThirdFibredPoint = fibringThirdPoint(point3, point5, point6, point7);
int myFourthLineFirstFibredPoint = fibringFirstPoint(point2, point7, point8, point13);
int myFourthLineSecondFibredPoint = fibringSecondPoint(point6, point7, point8,
point13);
int myFourthLineThirdFibredPoint = fibringThirdPoint(point6, point7, point8, point13);
```

```
fibringPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirst-
LineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibred-
Point, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLi-
neSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint,
myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
if (function == 2)
{
int myFirstLineFirstFibredPoint = fibringMaxFirstPoint(point1, point4, point7, point12);
int myFirstLineSecondFibredPoint = fibringMaxSecondPoint(point1, point4, point7,
point12);
int myFirstLineThirdFibredPoint = fibringMaxThirdPoint(point1, point4, point7,
point12);
int mySecondLineFirstFibredPoint = fibringMaxFirstPoint(point7, point9, point11,
point10);
int mySecondLineSecondFibredPoint = fibringMaxSecondPoint(point7, point9, point11,
point10);
int mySecondLineThirdFibredPoint = fibringMaxThirdPoint(point7, point9, point11,
point10);
int myThirdLineFirstFibredPoint = fibringMaxFirstPoint(point3, point5, point6, point7);
int myThirdLineSecondFibredPoint = fibringMaxSecondPoint(point3, point5, point6,
point7);
int myThirdLineThirdFibredPoint = fibringMaxThirdPoint(point3, point5, point6,
point7);
int myFourthLineFirstFibredPoint = fibringMaxFirstPoint(point2, point7, point8,
point13);
int myFourthLineSecondFibredPoint = fibringMaxSecondPoint(point6, point7, point8,
point13);
int myFourthLineThirdFibredPoint = fibringMaxThirdPoint(point6, point7, point8,
point13);
fibringMaxPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, my-
FirstLineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFi-
bredPoint, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLi-
neSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint,
myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
}
}
private void lbl8_Click(object sender, EventArgs e)
{
lstBox13Points.Items.Clear();
listFuzzyPoints_bufferEndPoint = 0;
if (txt1.Text == "" | txt3.Text == "" | txt4.Text == "" | txt5.Text == "" | txt6.Text == ""
| txt7.Text == "" | txt2.Text == "" | txt8.Text == "" | txt9.Text == "" | txt10.Text == "" |
txt11.Text == "" | txt12.Text == "" | txt13.Text == "")
{
```

```
MessageBox.Show(”lütfen bütün noktaları giriniz...”);
}
else
{
int lenghtPoint1 = txt1.Text.Length;
int lenghtPoint2 = txt2.Text.Length;
int lenghtPoint3 = txt3.Text.Length;
int lenghtPoint4 = txt4.Text.Length;
int lenghtPoint5 = txt5.Text.Length;
int lenghtPoint6 = txt6.Text.Length;
int lenghtPoint7 = txt7.Text.Length;
int lenghtPoint8 = txt8.Text.Length;
int lenghtPoint9 = txt9.Text.Length;
int lenghtPoint10 = txt10.Text.Length;
int lenghtPoint11 = txt11.Text.Length;
int lenghtPoint12 = txt12.Text.Length;
int lenghtPoint13 = txt13.Text.Length;
string strpoint1 = txt1.Text.Substring(2, lenghtPoint1 - 2);
string strpoint2 = txt2.Text.Substring(2, lenghtPoint2 - 2);
string strpoint3 = txt3.Text.Substring(2, lenghtPoint3 - 2);
string strpoint4 = txt4.Text.Substring(2, lenghtPoint4 - 2);
string strpoint5 = txt5.Text.Substring(2, lenghtPoint5 - 2);
string strpoint6 = txt6.Text.Substring(2, lenghtPoint6 - 2);
string strpoint7 = txt7.Text.Substring(2, lenghtPoint7 - 2);
string strpoint8 = txt8.Text.Substring(2, lenghtPoint8 - 2);
string strpoint9 = txt9.Text.Substring(2, lenghtPoint9 - 2);
string strpoint10 = txt10.Text.Substring(2, lenghtPoint10 - 2);
string strpoint11 = txt11.Text.Substring(2, lenghtPoint11 - 2);
string strpoint12 = txt12.Text.Substring(2, lenghtPoint12 - 2);
string strpoint13 = txt13.Text.Substring(2, lenghtPoint13 - 2);
int point1 = Convert.ToInt16(strpoint1);
int point2 = Convert.ToInt16(strpoint2);
int point3 = Convert.ToInt16(strpoint3);
int point4 = Convert.ToInt16(strpoint4);
int point5 = Convert.ToInt16(strpoint5);
int point6 = Convert.ToInt16(strpoint6);
int point7 = Convert.ToInt16(strpoint7);
int point8 = Convert.ToInt16(strpoint8);
int point9 = Convert.ToInt16(strpoint9);
int point10 = Convert.ToInt16(strpoint10);
int point11 = Convert.ToInt16(strpoint11);
int point12 = Convert.ToInt16(strpoint12);
int point13 = Convert.ToInt16(strpoint13);
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = point8;
listFuzzyPoints_bufferEndPoint++;
if (cmBoxOperator.Text==”minimum”) function=1;
if (cmBoxOperator.Text == ”maximum”) function = 2;
if (function == 1)
{
```

```
int myFirstLineFirstFibredPoint = fibringFirstPoint(point1, point6, point8, point10);
int myFirstLineSecondFibredPoint = fibringSecondPoint(point1, point6, point8, point10);
int myFirstLineThirdFibredPoint = fibringThirdPoint(point1, point6, point8, point10);
int mySecondLineFirstFibredPoint = fibringFirstPoint(point4, point5, point8, point11);
int mySecondLineSecondFibredPoint = fibringSecondPoint(point4, point5, point8, point11);
int mySecondLineThirdFibredPoint = fibringThirdPoint(point4, point5, point8, point11);
int myThirdLineFirstFibredPoint = fibringFirstPoint(point3, point8, point9, point12);
int myThirdLineSecondFibredPoint = fibringSecondPoint(point3, point8, point9, point12);
int myThirdLineThirdFibredPoint = fibringThirdPoint(point3, point8, point9, point12);
int myFourthLineFirstFibredPoint = fibringFirstPoint(point2, point7, point8, point13);
int myFourthLineSecondFibredPoint = fibringSecondPoint(point6, point7, point8, point13);
int myFourthLineThirdFibredPoint = fibringThirdPoint(point6, point7, point8, point13);
fibringPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirstLineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibredPoint, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLineSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint, myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
if (function == 2)
{
int myFirstLineFirstFibredPoint = fibringMaxFirstPoint(point1, point6, point8, point10);
int myFirstLineSecondFibredPoint = fibringMaxSecondPoint(point1, point6, point8, point10);
int myFirstLineThirdFibredPoint = fibringMaxThirdPoint(point1, point6, point8, point10);
int mySecondLineFirstFibredPoint = fibringMaxFirstPoint(point4, point5, point8, point11);
int mySecondLineSecondFibredPoint = fibringMaxSecondPoint(point4, point5, point8, point11);
int mySecondLineThirdFibredPoint = fibringMaxThirdPoint(point4, point5, point8, point11);
int myThirdLineFirstFibredPoint = fibringMaxFirstPoint(point3, point8, point9, point12);
int myThirdLineSecondFibredPoint = fibringMaxSecondPoint(point3, point8, point9, point12);
int myThirdLineThirdFibredPoint = fibringMaxThirdPoint(point3, point8, point9, point12);
int myFourthLineFirstFibredPoint = fibringMaxFirstPoint(point2, point7, point8, point13);
int myFourthLineSecondFibredPoint = fibringMaxSecondPoint(point6, point7, point8, point13);
int myFourthLineThirdFibredPoint = fibringMaxThirdPoint(point6, point7, point8, point13);
```

```
fibringMaxPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirstLineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibredPoint, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLineSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint, myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
}
}
private void lbl9_Click(object sender, EventArgs e)
{
lstBox13Points.Items.Clear();
listFuzzyPoints_bufferEndPoint = 0;
if (txt1.Text == "" | txt3.Text == "" | txt4.Text == "" | txt5.Text == "" | txt6.Text == "" | txt7.Text == "" | txt2.Text == "" | txt8.Text == "" | txt9.Text == "" | txt10.Text == "" | txt11.Text == "" | txt12.Text == "" | txt13.Text == "")
{
MessageBox.Show("lütfen bütün noktaları giriniz...");
}
else
{
int lenghtPoint1 = txt1.Text.Length;
int lenghtPoint2 = txt2.Text.Length;
int lenghtPoint3 = txt3.Text.Length;
int lenghtPoint4 = txt4.Text.Length;
int lenghtPoint5 = txt5.Text.Length;
int lenghtPoint6 = txt6.Text.Length;
int lenghtPoint7 = txt7.Text.Length;
int lenghtPoint8 = txt8.Text.Length;
int lenghtPoint9 = txt9.Text.Length;
int lenghtPoint10 = txt10.Text.Length;
int lenghtPoint11 = txt11.Text.Length;
int lenghtPoint12 = txt12.Text.Length;
int lenghtPoint13 = txt13.Text.Length;
string strpoint1 = txt1.Text.Substring(2, lenghtPoint1 - 2);
string strpoint2 = txt2.Text.Substring(2, lenghtPoint2 - 2);
string strpoint3 = txt3.Text.Substring(2, lenghtPoint3 - 2);
string strpoint4 = txt4.Text.Substring(2, lenghtPoint4 - 2);
string strpoint5 = txt5.Text.Substring(2, lenghtPoint5 - 2);
string strpoint6 = txt6.Text.Substring(2, lenghtPoint6 - 2);
string strpoint7 = txt7.Text.Substring(2, lenghtPoint7 - 2);
string strpoint8 = txt8.Text.Substring(2, lenghtPoint8 - 2);
string strpoint9 = txt9.Text.Substring(2, lenghtPoint9 - 2);
string strpoint10 = txt10.Text.Substring(2, lenghtPoint10 - 2);
string strpoint11 = txt11.Text.Substring(2, lenghtPoint11 - 2);
string strpoint12 = txt12.Text.Substring(2, lenghtPoint12 - 2);
string strpoint13 = txt13.Text.Substring(2, lenghtPoint13 - 2);
```

```
int point1 = Convert.ToInt16(strpoint1);
int point2 = Convert.ToInt16(strpoint2);
int point3 = Convert.ToInt16(strpoint3);
int point4 = Convert.ToInt16(strpoint4);
int point5 = Convert.ToInt16(strpoint5);
int point6 = Convert.ToInt16(strpoint6);
int point7 = Convert.ToInt16(strpoint7);
int point8 = Convert.ToInt16(strpoint8);
int point9 = Convert.ToInt16(strpoint9);
int point10 = Convert.ToInt16(strpoint10);
int point11 = Convert.ToInt16(strpoint11);
int point12 = Convert.ToInt16(strpoint12);
int point13 = Convert.ToInt16(strpoint13);
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = point9;
listFuzzyPoints_bufferEndPoint++;
if (cmBoxOperator.Text==”minimum”) function=1;
if (cmBoxOperator.Text == ”maximum”) function = 2;
if (function == 1)
{
int myFirstLineFirstFibredPoint = fibringFirstPoint(point1, point5, point9, point13);
int myFirstLineSecondFibredPoint = fibringSecondPoint(point1, point5, point9,
point13);
int myFirstLineThirdFibredPoint = fibringThirdPoint(point1, point5, point9, point13);
int mySecondLineFirstFibredPoint = fibringFirstPoint(point2, point4, point6, point9);
int mySecondLineSecondFibredPoint = fibringSecondPoint(point2, point4, point6,
point9);
int mySecondLineThirdFibredPoint = fibringThirdPoint(point2, point4, point6, point9);
int myThirdLineFirstFibredPoint = fibringFirstPoint(point3, point8, point9, point12);
int myThirdLineSecondFibredPoint = fibringSecondPoint(point3, point8, point9,
point12);
int myThirdLineThirdFibredPoint = fibringThirdPoint(point3, point8, point9, point12);
int myFourthLineFirstFibredPoint = fibringFirstPoint(point7, point9, point10, point11);
int myFourthLineSecondFibredPoint = fibringSecondPoint(point7, point9, point10,
point11);
int myFourthLineThirdFibredPoint = fibringThirdPoint(point7, point9, point10,
point11);
fibringPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirst-
LineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibred-
Point, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLi-
neSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint,
myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
if (function == 2)
{
int myFirstLineFirstFibredPoint = fibringMaxFirstPoint(point1, point5, point9, point13);
int myFirstLineSecondFibredPoint = fibringMaxSecondPoint(point1, point5, point9,
point13);
int myFirstLineThirdFibredPoint = fibringMaxThirdPoint(point1, point5, point9,
point13);
```

```
int mySecondLineFirstFibredPoint = fibringMaxFirstPoint(point2, point4, point6, point9);
int mySecondLineSecondFibredPoint = fibringMaxSecondPoint(point2, point4, point6, point9);
int mySecondLineThirdFibredPoint = fibringMaxThirdPoint(point2, point4, point6, point9);
int myThirdLineFirstFibredPoint = fibringMaxFirstPoint(point3, point8, point9, point12);
int myThirdLineSecondFibredPoint = fibringMaxSecondPoint(point3, point8, point9, point12);
int myThirdLineThirdFibredPoint = fibringMaxThirdPoint(point3, point8, point9, point12);
int myFourthLineFirstFibredPoint = fibringMaxFirstPoint(point7, point9, point10, point11);
int myFourthLineSecondFibredPoint = fibringMaxSecondPoint(point7, point9, point10, point11);
int myFourthLineThirdFibredPoint = fibringMaxThirdPoint(point7, point9, point10, point11);
fibringMaxPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirstLineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibredPoint, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLineSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint, myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
}
}
private void lbl10_Click(object sender, EventArgs e)
{
lstBox13Points.Items.Clear();
listFuzzyPoints_bufferEndPoint = 0;
if (txt1.Text == "" | txt3.Text == "" | txt4.Text == "" | txt5.Text == "" | txt6.Text == "" | txt7.Text == "" | txt2.Text == "" | txt8.Text == "" | txt9.Text == "" | txt10.Text == "" | txt11.Text == "" | txt12.Text == "" | txt13.Text == "")
{
MessageBox.Show("lütfen bütün noktaları giriniz...");
}
else
{
int lenghtPoint1 = txt1.Text.Length;
int lenghtPoint2 = txt2.Text.Length;
int lenghtPoint3 = txt3.Text.Length;
int lenghtPoint4 = txt4.Text.Length;
int lenghtPoint5 = txt5.Text.Length;
int lenghtPoint6 = txt6.Text.Length;
int lenghtPoint7 = txt7.Text.Length;
int lenghtPoint8 = txt8.Text.Length;
int lenghtPoint9 = txt9.Text.Length;
int lenghtPoint10 = txt10.Text.Length;
int lenghtPoint11 = txt11.Text.Length;
```

```
int lenghtPoint12 = txt12.Text.Length;
int lenghtPoint13 = txt13.Text.Length;
string strpoint1 = txt1.Text.Substring(2, lenghtPoint1 - 2);
string strpoint2 = txt2.Text.Substring(2, lenghtPoint2 - 2);
string strpoint3 = txt3.Text.Substring(2, lenghtPoint3 - 2);
string strpoint4 = txt4.Text.Substring(2, lenghtPoint4 - 2);
string strpoint5 = txt5.Text.Substring(2, lenghtPoint5 - 2);
string strpoint6 = txt6.Text.Substring(2, lenghtPoint6 - 2);
string strpoint7 = txt7.Text.Substring(2, lenghtPoint7 - 2);
string strpoint8 = txt8.Text.Substring(2, lenghtPoint8 - 2);
string strpoint9 = txt9.Text.Substring(2, lenghtPoint9 - 2);
string strpoint10 = txt10.Text.Substring(2, lenghtPoint10 - 2);
string strpoint11 = txt11.Text.Substring(2, lenghtPoint11 - 2);
string strpoint12 = txt12.Text.Substring(2, lenghtPoint12 - 2);
string strpoint13 = txt13.Text.Substring(2, lenghtPoint13 - 2);
int point1 = Convert.ToInt16(strpoint1);
int point2 = Convert.ToInt16(strpoint2);
int point3 = Convert.ToInt16(strpoint3);
int point4 = Convert.ToInt16(strpoint4);
int point5 = Convert.ToInt16(strpoint5);
int point6 = Convert.ToInt16(strpoint6);
int point7 = Convert.ToInt16(strpoint7);
int point8 = Convert.ToInt16(strpoint8);
int point9 = Convert.ToInt16(strpoint9);
int point10 = Convert.ToInt16(strpoint10);
int point11 = Convert.ToInt16(strpoint11);
int point12 = Convert.ToInt16(strpoint12);
int point13 = Convert.ToInt16(strpoint13);
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = point10;
listFuzzyPoints_bufferEndPoint++;
if (cmBoxOperator.Text==”minimum”) function=1;
if (cmBoxOperator.Text == ”maximum”) function = 2;
if (function == 1)
{
int myFirstLineFirstFibredPoint = fibringFirstPoint(point1, point6, point8, point10);
int myFirstLineSecondFibredPoint = fibringSecondPoint(point1, point6, point8,
point10);
int myFirstLineThirdFibredPoint = fibringThirdPoint(point1, point6, point8, point10);
int mySecondLineFirstFibredPoint = fibringFirstPoint(point2, point5, point10, point12);
int mySecondLineSecondFibredPoint = fibringSecondPoint(point2, point5, point10,
point12);
int mySecondLineThirdFibredPoint = fibringThirdPoint(point2, point5, point10,
point12);
int myThirdLineFirstFibredPoint = fibringFirstPoint(point3, point4, point10, point13);
int myThirdLineSecondFibredPoint = fibringSecondPoint(point3, point4, point10,
point13);
int myThirdLineThirdFibredPoint = fibringThirdPoint(point3, point4, point10, point13);
int myFourthLineFirstFibredPoint = fibringFirstPoint(point7, point9, point10, point11);
```

```
int myFourthLineSecondFibredPoint = fibringSecondPoint(point7, point9, point10, point11);
int myFourthLineThirdFibredPoint = fibringThirdPoint(point7, point9, point10, point11);
fibringPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirstLineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibredPoint, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLineSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint, myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
if (function == 2)
{
int myFirstLineFirstFibredPoint = fibringMaxFirstPoint(point1, point6, point8, point10);
int myFirstLineSecondFibredPoint = fibringMaxSecondPoint(point1, point6, point8, point10);
int myFirstLineThirdFibredPoint = fibringMaxThirdPoint(point1, point6, point8, point10);
int mySecondLineFirstFibredPoint = fibringMaxFirstPoint(point2, point5, point10, point12);
int mySecondLineSecondFibredPoint = fibringMaxSecondPoint(point2, point5, point10, point12);
int mySecondLineThirdFibredPoint = fibringMaxThirdPoint(point2, point5, point10, point12);
int myThirdLineFirstFibredPoint = fibringMaxFirstPoint(point3, point4, point10, point13);
int myThirdLineSecondFibredPoint = fibringMaxSecondPoint(point3, point4, point10, point13);
int myThirdLineThirdFibredPoint = fibringMaxThirdPoint(point3, point4, point10, point13);
int myFourthLineFirstFibredPoint = fibringMaxFirstPoint(point7, point9, point10, point11);
int myFourthLineSecondFibredPoint = fibringMaxSecondPoint(point7, point9, point10, point11);
int myFourthLineThirdFibredPoint = fibringMaxThirdPoint(point7, point9, point10, point11); fibringMaxPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirstLineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibredPoint, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLineSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint, myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
}
}
private void lbl11_Click(object sender, EventArgs e) {
lstBox13Points.Items.Clear();
listFuzzyPoints_bufferEndPoint = 0;
if (txt1.Text == "" | txt3.Text == "" | txt4.Text == "" | txt5.Text == "" | txt6.Text == "" | txt7.Text == "" | txt2.Text == "" | txt8.Text == "" | txt9.Text == "" | txt10.Text == "" | txt11.Text == "" | txt12.Text == "" | txt13.Text == "")
{
MessageBox.Show("lütfen bütün noktaları giriniz...");
```

```
}
else
{
int lenghtPoint1 = txt1.Text.Length;
int lenghtPoint2 = txt2.Text.Length;
int lenghtPoint3 = txt3.Text.Length;
int lenghtPoint4 = txt4.Text.Length;
int lenghtPoint5 = txt5.Text.Length;
int lenghtPoint6 = txt6.Text.Length;
int lenghtPoint7 = txt7.Text.Length;
int lenghtPoint8 = txt8.Text.Length;
int lenghtPoint9 = txt9.Text.Length;
int lenghtPoint10 = txt10.Text.Length;
int lenghtPoint11 = txt11.Text.Length;
int lenghtPoint12 = txt12.Text.Length;
int lenghtPoint13 = txt13.Text.Length;
string strpoint1 = txt1.Text.Substring(2, lenghtPoint1 - 2);
string strpoint2 = txt2.Text.Substring(2, lenghtPoint2 - 2);
string strpoint3 = txt3.Text.Substring(2, lenghtPoint3 - 2);
string strpoint4 = txt4.Text.Substring(2, lenghtPoint4 - 2);
string strpoint5 = txt5.Text.Substring(2, lenghtPoint5 - 2);
string strpoint6 = txt6.Text.Substring(2, lenghtPoint6 - 2);
string strpoint7 = txt7.Text.Substring(2, lenghtPoint7 - 2);
string strpoint8 = txt8.Text.Substring(2, lenghtPoint8 - 2);
string strpoint9 = txt9.Text.Substring(2, lenghtPoint9 - 2);
string strpoint10 = txt10.Text.Substring(2, lenghtPoint10 - 2);
string strpoint11 = txt11.Text.Substring(2, lenghtPoint11 - 2);
string strpoint12 = txt12.Text.Substring(2, lenghtPoint12 - 2);
string strpoint13 = txt13.Text.Substring(2, lenghtPoint13 - 2);
int point1 = Convert.ToInt16(strpoint1);
int point2 = Convert.ToInt16(strpoint2);
int point3 = Convert.ToInt16(strpoint3);
int point4 = Convert.ToInt16(strpoint4);
int point5 = Convert.ToInt16(strpoint5);
int point6 = Convert.ToInt16(strpoint6);
int point7 = Convert.ToInt16(strpoint7);
int point8 = Convert.ToInt16(strpoint8);
int point9 = Convert.ToInt16(strpoint9);
int point10 = Convert.ToInt16(strpoint10);
int point11 = Convert.ToInt16(strpoint11);
int point12 = Convert.ToInt16(strpoint12);
int point13 = Convert.ToInt16(strpoint13);
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = point11;
listFuzzyPoints_bufferEndPoint++;
if (cmBoxOperator.Text==”minimum”) function=1;
if (cmBoxOperator.Text == ”maximum”) function = 2;
if (function == 1)
```

```
{
int myFirstLineFirstFibredPoint = fibringFirstPoint(point6, point11, point12, point13);
int myFirstLineSecondFibredPoint = fibringSecondPoint(point6, point11, point12,
point13);
int myFirstLineThirdFibredPoint = fibringThirdPoint(point6, point11, point12, point13);
int mySecondLineFirstFibredPoint = fibringFirstPoint(point1, point2, point3, point11);
int mySecondLineSecondFibredPoint = fibringSecondPoint(point1, point2, point3,
point11);
int mySecondLineThirdFibredPoint = fibringThirdPoint(point1, point2, point3, point11);
int myThirdLineFirstFibredPoint = fibringFirstPoint(point4, point5, point8, point11);
int myThirdLineSecondFibredPoint = fibringSecondPoint(point4, point5, point8,
point11);
int myThirdLineThirdFibredPoint = fibringThirdPoint(point4, point5, point8, point11);
int myFourthLineFirstFibredPoint = fibringFirstPoint(point7, point9, point10, point11);
int myFourthLineSecondFibredPoint = fibringSecondPoint(point7, point9, point10,
point11);
int myFourthLineThirdFibredPoint = fibringThirdPoint(point7, point9, point10,
point11);
fibringPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirst-
LineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibred-
Point, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLi-
neSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint,
myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
if (function == 2)
{
int myFirstLineFirstFibredPoint = fibringMaxFirstPoint(point6, point11, point12,
point13);
int myFirstLineSecondFibredPoint = fibringMaxSecondPoint(point6, point11, point12,
point13);
int myFirstLineThirdFibredPoint = fibringMaxThirdPoint(point6, point11, point12,
point13);
int mySecondLineFirstFibredPoint = fibringMaxFirstPoint(point1, point2, point3,
point11);
int mySecondLineSecondFibredPoint = fibringMaxSecondPoint(point1, point2, point3,
point11);
int mySecondLineThirdFibredPoint = fibringMaxThirdPoint(point1, point2, point3,
point11);
int myThirdLineFirstFibredPoint = fibringMaxFirstPoint(point4, point5, point8,
point11);
int myThirdLineSecondFibredPoint = fibringMaxSecondPoint(point4, point5, point8,
point11);
int myThirdLineThirdFibredPoint = fibringMaxThirdPoint(point4, point5, point8,
point11);
int myFourthLineFirstFibredPoint = fibringMaxFirstPoint(point7, point9, point10,
point11);
int myFourthLineSecondFibredPoint = fibringMaxSecondPoint(point7, point9, point10,
point11);
```

```
int myFourthLineThirdFibredPoint = fibringMaxThirdPoint(point7, point9, point10, point11);
fibringMaxPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirstLineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibredPoint, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLineSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint, myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
}
}
private void lbl12_Click(object sender, EventArgs e)
{
lstBox13Points.Items.Clear();
listFuzzyPoints_bufferEndPoint = 0;
if (txt1.Text == "" | txt3.Text == "" | txt4.Text == "" | txt5.Text == "" | txt6.Text == "" | txt7.Text == "" | txt2.Text == "" | txt8.Text == "" | txt9.Text == "" | txt10.Text == "" | txt11.Text == "" | txt12.Text == "" | txt13.Text == "")
{
MessageBox.Show("lütfen bütün noktaları giriniz...");
}
else
{
int lenghtPoint1 = txt1.Text.Length;
int lenghtPoint2 = txt2.Text.Length;
int lenghtPoint3 = txt3.Text.Length;
int lenghtPoint4 = txt4.Text.Length;
int lenghtPoint5 = txt5.Text.Length;
int lenghtPoint6 = txt6.Text.Length;
int lenghtPoint7 = txt7.Text.Length;
int lenghtPoint8 = txt8.Text.Length;
int lenghtPoint9 = txt9.Text.Length;
int lenghtPoint10 = txt10.Text.Length;
int lenghtPoint11 = txt11.Text.Length;
int lenghtPoint12 = txt12.Text.Length;
int lenghtPoint13 = txt13.Text.Length;
string strpoint1 = txt1.Text.Substring(2, lenghtPoint1 - 2);
string strpoint2 = txt2.Text.Substring(2, lenghtPoint2 - 2);
string strpoint3 = txt3.Text.Substring(2, lenghtPoint3 - 2);
string strpoint4 = txt4.Text.Substring(2, lenghtPoint4 - 2);
string strpoint5 = txt5.Text.Substring(2, lenghtPoint5 - 2);
string strpoint6 = txt6.Text.Substring(2, lenghtPoint6 - 2);
string strpoint7 = txt7.Text.Substring(2, lenghtPoint7 - 2);
string strpoint8 = txt8.Text.Substring(2, lenghtPoint8 - 2);
string strpoint9 = txt9.Text.Substring(2, lenghtPoint9 - 2);
string strpoint10 = txt10.Text.Substring(2, lenghtPoint10 - 2);
string strpoint11 = txt11.Text.Substring(2, lenghtPoint11 - 2);
string strpoint12 = txt12.Text.Substring(2, lenghtPoint12 - 2);
string strpoint13 = txt13.Text.Substring(2, lenghtPoint13 - 2);
int point1 = Convert.ToInt16(strpoint1);
int point2 = Convert.ToInt16(strpoint2);
```

```
int point3 = Convert.ToInt16(strpoint3);
int point4 = Convert.ToInt16(strpoint4);
int point5 = Convert.ToInt16(strpoint5);
int point6 = Convert.ToInt16(strpoint6);
int point7 = Convert.ToInt16(strpoint7);
int point8 = Convert.ToInt16(strpoint8);
int point9 = Convert.ToInt16(strpoint9);
int point10 = Convert.ToInt16(strpoint10);
int point11 = Convert.ToInt16(strpoint11);
int point12 = Convert.ToInt16(strpoint12);
int point13 = Convert.ToInt16(strpoint13);
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = point12;
listFuzzyPoints_bufferEndPoint++;
if (cmBoxOperator.Text==”minimum”) function=1;
if (cmBoxOperator.Text == ”maximum”) function = 2;
if (function == 1)
{
int myFirstLineFirstFibredPoint = fibringFirstPoint(point3, point8, point9, point12);
int myFirstLineSecondFibredPoint = fibringSecondPoint(point3, point8, point9,
point12);
int myFirstLineThirdFibredPoint = fibringThirdPoint(point3, point8, point9, point12);
int mySecondLineFirstFibredPoint = fibringFirstPoint(point2, point5, point10, point12);
int mySecondLineSecondFibredPoint = fibringSecondPoint(point2, point5, point10,
point12);
int mySecondLineThirdFibredPoint = fibringThirdPoint(point2, point5, point10,
point12);
int myThirdLineFirstFibredPoint = fibringFirstPoint(point6, point11, point12, point13);
int myThirdLineSecondFibredPoint = fibringSecondPoint(point6, point11, point12,
point13);
int myThirdLineThirdFibredPoint = fibringThirdPoint(point6, point11, point12, point13);
int myFourthLineFirstFibredPoint = fibringFirstPoint(point1, point4, point7, point12);
int myFourthLineSecondFibredPoint = fibringSecondPoint(point1, point4, point7,
point12);
int myFourthLineThirdFibredPoint = fibringThirdPoint(point1, point4, point7, point12);
fibringPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirst-
LineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibred-
Point, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLi-
neSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint,
myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
if (function == 2)
{
int myFirstLineFirstFibredPoint = fibringMaxFirstPoint(point3, point8, point9, point12);
int myFirstLineSecondFibredPoint = fibringMaxSecondPoint(point3, point8, point9,
point12);
int myFirstLineThirdFibredPoint = fibringMaxThirdPoint(point3, point8, point9,
point12);
int mySecondLineFirstFibredPoint = fibringMaxFirstPoint(point2, point5, point10,
point12);
```

```
int mySecondLineSecondFibredPoint = fibringMaxSecondPoint(point2, point5, point10, point12);
int mySecondLineThirdFibredPoint = fibringMaxThirdPoint(point2, point5, point10, point12);
int myThirdLineFirstFibredPoint = fibringMaxFirstPoint(point6, point11, point12, point13);
int myThirdLineSecondFibredPoint = fibringMaxSecondPoint(point6, point11, point12, point13);
int myThirdLineThirdFibredPoint = fibringMaxThirdPoint(point6, point11, point12, point13);
int myFourthLineFirstFibredPoint = fibringMaxFirstPoint(point1, point4, point7, point12);
int myFourthLineSecondFibredPoint = fibringMaxSecondPoint(point1, point4, point7, point12);
int myFourthLineThirdFibredPoint = fibringMaxThirdPoint(point1, point4, point7, point12);
fibringMaxPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirstLineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibredPoint, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLineSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint, myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
}
}
private void lbl13_Click(object sender, EventArgs e)
{
lstBox13Points.Items.Clear();
listFuzzyPoints_bufferEndPoint = 0;
if (txt1.Text == "" | txt3.Text == "" | txt4.Text == "" | txt5.Text == "" | txt6.Text == "" | txt7.Text == "" | txt2.Text == "" | txt8.Text == "" | txt9.Text == "" | txt10.Text == "" | txt11.Text == "" | txt12.Text == "" | txt13.Text == "")
{
MessageBox.Show("lütfen bütün noktaları giriniz...");
}
else
{
int lenghtPoint1 = txt1.Text.Length;
int lenghtPoint2 = txt2.Text.Length;
int lenghtPoint3 = txt3.Text.Length;
int lenghtPoint4 = txt4.Text.Length;
int lenghtPoint5 = txt5.Text.Length;
int lenghtPoint6 = txt6.Text.Length;
int lenghtPoint7 = txt7.Text.Length;
int lenghtPoint8 = txt8.Text.Length;
int lenghtPoint9 = txt9.Text.Length;
int lenghtPoint10 = txt10.Text.Length;
int lenghtPoint11 = txt11.Text.Length;
int lenghtPoint12 = txt12.Text.Length;
int lenghtPoint13 = txt13.Text.Length;
string strpoint1 = txt1.Text.Substring(2, lenghtPoint1 - 2);
```

```
string strpoint2 = txt2.Text.Substring(2, lenghtPoint2 - 2);
string strpoint3 = txt3.Text.Substring(2, lenghtPoint3 - 2);
string strpoint4 = txt4.Text.Substring(2, lenghtPoint4 - 2);
string strpoint5 = txt5.Text.Substring(2, lenghtPoint5 - 2);
string strpoint6 = txt6.Text.Substring(2, lenghtPoint6 - 2);
string strpoint7 = txt7.Text.Substring(2, lenghtPoint7 - 2);
string strpoint8 = txt8.Text.Substring(2, lenghtPoint8 - 2);
string strpoint9 = txt9.Text.Substring(2, lenghtPoint9 - 2);
string strpoint10 = txt10.Text.Substring(2, lenghtPoint10 - 2);
string strpoint11 = txt11.Text.Substring(2, lenghtPoint11 - 2);
string strpoint12 = txt12.Text.Substring(2, lenghtPoint12 - 2);
string strpoint13 = txt13.Text.Substring(2, lenghtPoint13 - 2);
int point1 = Convert.ToInt16(strpoint1);
int point2 = Convert.ToInt16(strpoint2);
int point3 = Convert.ToInt16(strpoint3);
int point4 = Convert.ToInt16(strpoint4);
int point5 = Convert.ToInt16(strpoint5);
int point6 = Convert.ToInt16(strpoint6);
int point7 = Convert.ToInt16(strpoint7);
int point8 = Convert.ToInt16(strpoint8);
int point9 = Convert.ToInt16(strpoint9);
int point10 = Convert.ToInt16(strpoint10);
int point11 = Convert.ToInt16(strpoint11);
int point12 = Convert.ToInt16(strpoint12);
int point13 = Convert.ToInt16(strpoint13);
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = point13;
listFuzzyPoints_bufferEndPoint++;
if (cmBoxOperator.Text==”minimum”) function=1;
if (cmBoxOperator.Text == ”maximum”) function = 2;
if (function == 1)
{
int myFirstLineFirstFibredPoint = fibringFirstPoint(point1, point4, point5, point13);
int myFirstLineSecondFibredPoint = fibringSecondPoint(point1, point4, point5,
point13);
int myFirstLineThirdFibredPoint = fibringThirdPoint(point1, point4, point5, point13);
int mySecondLineFirstFibredPoint = fibringFirstPoint(point3, point4, point10, point13);
int mySecondLineSecondFibredPoint = fibringSecondPoint(point3, point4, point10,
point13);
int mySecondLineThirdFibredPoint = fibringThirdPoint(point3, point4, point10,
point13);
int myThirdLineFirstFibredPoint = fibringFirstPoint(point6, point11, point12, point13);
int myThirdLineSecondFibredPoint = fibringSecondPoint(point6, point11, point12,
point13);
int myThirdLineThirdFibredPoint = fibringThirdPoint(point6, point11, point12, point13);
int myFourthLineFirstFibredPoint = fibringFirstPoint(point2, point8, point7, point13);
int myFourthLineSecondFibredPoint = fibringSecondPoint(point2, point8, point7,
point13);
int myFourthLineThirdFibredPoint = fibringThirdPoint(point2, point8, point7, point13);
```

```
fibringPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, myFirst-
LineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFibred-
Point, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLi-
neSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint,
myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
if (function == 2)
{
int myFirstLineFirstFibredPoint = fibringMaxFirstPoint(point1, point4, point5, point13);
int myFirstLineSecondFibredPoint = fibringMaxSecondPoint(point1, point4, point5,
point13);
int myFirstLineThirdFibredPoint = fibringMaxThirdPoint(point1, point4, point5,
point13);
int mySecondLineFirstFibredPoint = fibringMaxFirstPoint(point3, point4, point10,
point13);
int mySecondLineSecondFibredPoint = fibringMaxSecondPoint(point3, point4, point10,
point13);
int mySecondLineThirdFibredPoint = fibringMaxThirdPoint(point3, point4, point10,
point13);
int myThirdLineFirstFibredPoint = fibringMaxFirstPoint(point6, point11, point12,
point13);
int myThirdLineSecondFibredPoint = fibringMaxSecondPoint(point6, point11, point12,
point13);
int myThirdLineThirdFibredPoint = fibringMaxThirdPoint(point6, point11, point12,
point13);
int myFourthLineFirstFibredPoint = fibringMaxFirstPoint(point2, point8, point7,
point13);
int myFourthLineSecondFibredPoint = fibringMaxSecondPoint(point2, point8, point7,
point13);
int myFourthLineThirdFibredPoint = fibringMaxThirdPoint(point2, point8, point7,
point13);
fibringMaxPoint(myFirstLineFirstFibredPoint, myFirstLineSecondFibredPoint, my-
FirstLineThirdFibredPoint, mySecondLineFirstFibredPoint, mySecondLineSecondFi-
bredPoint, mySecondLineThirdFibredPoint, myThirdLineFirstFibredPoint, myThirdLi-
neSecondFibredPoint, myThirdLineThirdFibredPoint, myFourthLineFirstFibredPoint,
myFourthLineSecondFibredPoint, myFourthLineThirdFibredPoint);
}
}
}
public int fibringPoint(int fibringPoint1, int fibringPoint2, int fibringPoint3, int fib-
ringPoint4, int fibringPoint5, int fibringPoint6, int fibringPoint7, int fibringPoint8, int
fibringPoint9, int fibringPoint10, int fibringPoint11, int fibringPoint12)
{
int add = 0;
if (fibringPoint1 <= fibringPoint4) add++;
if (fibringPoint1 <= fibringPoint5) add++;
if (fibringPoint1 <= fibringPoint6) add++;
if (fibringPoint1 <= fibringPoint7) add++;
if (fibringPoint1 <= fibringPoint8) add++;
if (fibringPoint1 <= fibringPoint9) add++;
```

```
if (fibringPoint1 <= fibringPoint10) add++;
if (fibringPoint1 <= fibringPoint11) add++;
if (fibringPoint1 <= fibringPoint12) add++;
if (add > 0 & fibringPoint1 != listFuzzyPoints_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint1;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint2 <= fibringPoint4) add++;
if (fibringPoint2 <= fibringPoint5) add++;
if (fibringPoint2 <= fibringPoint6) add++;
if (fibringPoint2 <= fibringPoint7) add++;
if (fibringPoint2 <= fibringPoint8) add++;
if (fibringPoint2 <= fibringPoint9) add++;
if (fibringPoint2 <= fibringPoint10) add++;
if (fibringPoint2 <= fibringPoint11) add++;
if (fibringPoint2 <= fibringPoint12) add++;
if (add > 0 & fibringPoint2 != fibringPoint1 & fibringPoint2 != listFuzzyPoints_buffer[0,
0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint2;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint3 <= fibringPoint4) add++;
if (fibringPoint3 <= fibringPoint5) add++;
if (fibringPoint3 <= fibringPoint6) add++;
if (fibringPoint3 <= fibringPoint7) add++;
if (fibringPoint3 <= fibringPoint8) add++;
if (fibringPoint3 <= fibringPoint9) add++;
if (fibringPoint3 <= fibringPoint10) add++;
if (fibringPoint3 <= fibringPoint11) add++;
if (fibringPoint3 <= fibringPoint12) add++;
if (add > 0 & fibringPoint3 != fibringPoint1 & fibringPoint3 != fibringPoint2 & fibring-
Point3 != listFuzzyPoints_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint3;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint4 <= fibringPoint1) add++;
if (fibringPoint4 <= fibringPoint2) add++;
if (fibringPoint4 <= fibringPoint3) add++;
if (fibringPoint4 <= fibringPoint7) add++;
if (fibringPoint4 <= fibringPoint8) add++;
if (fibringPoint4 <= fibringPoint9) add++;
if (fibringPoint4 <= fibringPoint10) add++;
if (fibringPoint4 <= fibringPoint11) add++;
if (fibringPoint4 <= fibringPoint12) add++;
```

```
if (add > 0 & fibringPoint4 != fibringPoint3 & fibringPoint4 != fibringPoint2 & fibring-
Point4 != fibringPoint1 & fibringPoint4 != listFuzzyPoints_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint4;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint5 <= fibringPoint1) add++;
if (fibringPoint5 <= fibringPoint2) add++;
if (fibringPoint5 <= fibringPoint3) add++;
if (fibringPoint5 <= fibringPoint7) add++;
if (fibringPoint5 <= fibringPoint8) add++;
if (fibringPoint5 <= fibringPoint9) add++;
if (fibringPoint5 <= fibringPoint10) add++;
if (fibringPoint5 <= fibringPoint11) add++;
if (fibringPoint5 <= fibringPoint12) add++;
if (add > 0 & fibringPoint5 != fibringPoint4 & fibringPoint5 != fibringPoint3 & fibring-
Point5 != fibringPoint2 & fibringPoint5 != fibringPoint1 & fibringPoint5 != listFuzzy-
Points_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint5;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint6 <= fibringPoint1) add++;
if (fibringPoint6 <= fibringPoint2) add++;
if (fibringPoint6 <= fibringPoint3) add++;
if (fibringPoint6 <= fibringPoint7) add++;
if (fibringPoint6 <= fibringPoint8) add++;
if (fibringPoint6 <= fibringPoint9) add++;
if (fibringPoint6 <= fibringPoint10) add++;
if (fibringPoint6 <= fibringPoint11) add++;
if (fibringPoint6 <= fibringPoint12) add++;
if (add > 0 & fibringPoint6 != fibringPoint5 & fibringPoint6 != fibringPoint4 & fibring-
Point6 != fibringPoint3 & fibringPoint6 != fibringPoint2 & fibringPoint6 != fibringPoint1
& fibringPoint6 != listFuzzyPoints_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint6;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint7 <= fibringPoint1) add++;
if (fibringPoint7 <= fibringPoint2) add++;
if (fibringPoint7 <= fibringPoint3) add++;
if (fibringPoint7 <= fibringPoint4) add++;
if (fibringPoint7 <= fibringPoint5) add++;
if (fibringPoint7 <= fibringPoint6) add++;
if (fibringPoint7 <= fibringPoint10) add++;
if (fibringPoint7 <= fibringPoint11) add++;
if (fibringPoint7 <= fibringPoint12) add++;
```

```
if (add > 0 & fibringPoint7 != fibringPoint6 & fibringPoint7 != fibringPoint5 & fibring-
Point7 != fibringPoint4 & fibringPoint7 != fibringPoint3 & fibringPoint7 != fibringPoint2
& fibringPoint7 != fibringPoint1 & fibringPoint7 != listFuzzyPoints_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint7;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint8 <= fibringPoint1) add++;
if (fibringPoint8 <= fibringPoint2) add++;
if (fibringPoint8 <= fibringPoint3) add++;
if (fibringPoint8 <= fibringPoint4) add++;
if (fibringPoint8 <= fibringPoint5) add++;
if (fibringPoint8 <= fibringPoint6) add++;
if (fibringPoint8 <= fibringPoint10) add++;
if (fibringPoint8 <= fibringPoint11) add++;
if (fibringPoint8 <= fibringPoint12) add++;
if (add > 0 & fibringPoint8 != fibringPoint7 & fibringPoint8 != fibringPoint6 & fibring-
Point8 != fibringPoint5 & fibringPoint8 != fibringPoint4 & fibringPoint8 != fibringPoint3
& fibringPoint8 != fibringPoint2 & fibringPoint8 != fibringPoint1 & fibringPoint8 !=
listFuzzyPoints_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint8;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint9 <= fibringPoint1) add++;
if (fibringPoint9 <= fibringPoint2) add++;
if (fibringPoint9 <= fibringPoint3) add++;
if (fibringPoint9 <= fibringPoint4) add++;
if (fibringPoint9 <= fibringPoint5) add++;
if (fibringPoint9 <= fibringPoint6) add++;
if (fibringPoint9 <= fibringPoint10) add++;
if (fibringPoint9 <= fibringPoint11) add++;
if (fibringPoint9 <= fibringPoint12) add++;
if (add > 0 & fibringPoint9 != fibringPoint8 & fibringPoint9 != fibringPoint7 & fibring-
Point9 != fibringPoint6 & fibringPoint9 != fibringPoint5 & fibringPoint9 != fibringPoint4
& fibringPoint9 != fibringPoint3 & fibringPoint9 != fibringPoint2 & fibringPoint9 !=
fibringPoint1 & fibringPoint9 != listFuzzyPoints_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint9;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint10 <= fibringPoint1) add++;
if (fibringPoint10 <= fibringPoint2) add++;
if (fibringPoint10 <= fibringPoint3) add++;
if (fibringPoint10 <= fibringPoint7) add++;
if (fibringPoint10 <= fibringPoint8) add++;
if (fibringPoint10 <= fibringPoint9) add++;
```

```
if (fibringPoint10 <= fibringPoint4) add++;
if (fibringPoint10 <= fibringPoint5) add++;
if (fibringPoint10 <= fibringPoint6) add++;
if (add > 0 & fibringPoint10 != fibringPoint9 & fibringPoint10 != fibringPoint8 &
fibringPoint10 != fibringPoint7 & fibringPoint10 != fibringPoint6 & fibringPoint10 !=
fibringPoint5 & fibringPoint10 != fibringPoint4 & fibringPoint10 != fibringPoint3 &
fibringPoint10 != fibringPoint2 & fibringPoint10 != fibringPoint1 & fibringPoint10 !=
listFuzzyPoints_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint10;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint11 <= fibringPoint1) add++;
if (fibringPoint11 <= fibringPoint2) add++;
if (fibringPoint11 <= fibringPoint3) add++;
if (fibringPoint11 <= fibringPoint7) add++;
if (fibringPoint11 <= fibringPoint8) add++;
if (fibringPoint11 <= fibringPoint9) add++;
if (fibringPoint11 <= fibringPoint4) add++;
if (fibringPoint11 <= fibringPoint5) add++;
if (fibringPoint11 <= fibringPoint6) add++;
if (add > 0 & fibringPoint11 != fibringPoint10 & fibringPoint11 != fibringPoint9 &
fibringPoint11 != fibringPoint8 & fibringPoint11 != fibringPoint7 & fibringPoint11 !=
fibringPoint6 & fibringPoint11 != fibringPoint5 & fibringPoint11 != fibringPoint4 &
fibringPoint11 != fibringPoint3 & fibringPoint11 != fibringPoint2 & fibringPoint11 !=
fibringPoint1 & fibringPoint11 != listFuzzyPoints_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint11;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint12 <= fibringPoint1) add++;
if (fibringPoint12 <= fibringPoint2) add++;
if (fibringPoint12 <= fibringPoint3) add++;
if (fibringPoint12 <= fibringPoint7) add++;
if (fibringPoint12 <= fibringPoint8) add++;
if (fibringPoint12 <= fibringPoint9) add++;
if (fibringPoint12 <= fibringPoint4) add++;
if (fibringPoint12 <= fibringPoint5) add++;
if (fibringPoint12 <= fibringPoint6) add++;
if (add > 0 & fibringPoint12 != fibringPoint11 & fibringPoint12 != fibringPoint10 &
fibringPoint12 != fibringPoint9 & fibringPoint12 != fibringPoint8 & fibringPoint12
!= fibringPoint7 & fibringPoint12 != fibringPoint6 & fibringPoint12 != fibringPoint5
& fibringPoint12 != fibringPoint4 & fibringPoint12 != fibringPoint3 & fibringPoint12
!= fibringPoint2 & fibringPoint12 != fibringPoint1 & fibringPoint12!= listFuzzy-
Points_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint12;
listFuzzyPoints_bufferEndPoint++;
```

```
}
for (int fuzzyPoints = 0; fuzzyPoints < listFuzzyPoints_bufferEndPoint; fuzzyPoints++)
{
lstBox13Points.Items.Add((fuzzyPoints+1)+"..."+"0." + listFuzzy-
Points_buffer[fuzzyPoints, 0]);
}
return (fibringPoint1);
}
public int fibringMaxPoint(int fibringPoint1, int fibringPoint2, int fibringPoint3, int
fibringPoint4, int fibringPoint5, int fibringPoint6, int fibringPoint7, int fibringPoint8, int
fibringPoint9, int fibringPoint10, int fibringPoint11, int fibringPoint12)
{
int add = 0;
if (fibringPoint1 >= fibringPoint4) add++;
if (fibringPoint1 >= fibringPoint5) add++;
if (fibringPoint1 >= fibringPoint6) add++;
if (fibringPoint1 >= fibringPoint7) add++;
if (fibringPoint1 >= fibringPoint8) add++;
if (fibringPoint1 >= fibringPoint9) add++;
if (fibringPoint1 >= fibringPoint10) add++;
if (fibringPoint1 >= fibringPoint11) add++;
if (fibringPoint1 >= fibringPoint12) add++;
if (add > 0 & fibringPoint1 != listFuzzyPoints_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint1;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint2 >= fibringPoint4) add++;
if (fibringPoint2 >= fibringPoint5) add++;
if (fibringPoint2 >= fibringPoint6) add++;
if (fibringPoint2 >= fibringPoint7) add++;
if (fibringPoint2 >= fibringPoint8) add++;
if (fibringPoint2 >= fibringPoint9) add++;
if (fibringPoint2 >= fibringPoint10) add++;
if (fibringPoint2 >= fibringPoint11) add++;
if (fibringPoint2 >= fibringPoint12) add++;
if (add > 0 & fibringPoint2 != fibringPoint1 & fibringPoint2 != listFuzzyPoints_buffer[0,
0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint2;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint3 >= fibringPoint4) add++;
if (fibringPoint3 >= fibringPoint5) add++;
if (fibringPoint3 >= fibringPoint6) add++;
if (fibringPoint3 >= fibringPoint7) add++;
if (fibringPoint3 >= fibringPoint8) add++;
if (fibringPoint3 >= fibringPoint9) add++;
```

```
if (fibringPoint3 >= fibringPoint10) add++;
if (fibringPoint3 >= fibringPoint11) add++;
if (fibringPoint3 >= fibringPoint12) add++;
if (add > 0 & fibringPoint3 != fibringPoint1 & fibringPoint3 != fibringPoint2 & fibring-
Point3 != listFuzzyPoints_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint3;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint4 >= fibringPoint1) add++;
if (fibringPoint4 >= fibringPoint2) add++;
if (fibringPoint4 >= fibringPoint3) add++;
if (fibringPoint4 >= fibringPoint7) add++;
if (fibringPoint4 >= fibringPoint8) add++;
if (fibringPoint4 >= fibringPoint9) add++;
if (fibringPoint4 >= fibringPoint10) add++;
if (fibringPoint4 >= fibringPoint11) add++;
if (fibringPoint4 >= fibringPoint12) add++;
if (add > 0 & fibringPoint4 != fibringPoint3 & fibringPoint4 != fibringPoint2 & fibring-
Point4 != fibringPoint1 & fibringPoint4 != listFuzzyPoints_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint4;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint5 >= fibringPoint1) add++;
if (fibringPoint5 >= fibringPoint2) add++;
if (fibringPoint5 >= fibringPoint3) add++;
if (fibringPoint5 >= fibringPoint7) add++;
if (fibringPoint5 >= fibringPoint8) add++;
if (fibringPoint5 >= fibringPoint9) add++;
if (fibringPoint5 >= fibringPoint10) add++;
if (fibringPoint5 >= fibringPoint11) add++;
if (fibringPoint5 >= fibringPoint12) add++;
if (add > 0 & fibringPoint5 != fibringPoint4 & fibringPoint5 != fibringPoint3 &
fibringPoint5 != fibringPoint2 & fibringPoint5 != fibringPoint1 & fibringPoint5 !=
listFuzzyPoints_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint5;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint6 >= fibringPoint1) add++;
if (fibringPoint6 >= fibringPoint2) add++;
if (fibringPoint6 >= fibringPoint3) add++;
if (fibringPoint6 >= fibringPoint7) add++;
if (fibringPoint6 >= fibringPoint8) add++;
if (fibringPoint6 >= fibringPoint9) add++;
```

```
if (fibringPoint6 >= fibringPoint10) add++;
if (fibringPoint6 >= fibringPoint11) add++;
if (fibringPoint6 >= fibringPoint12) add++;
if (add > 0 & fibringPoint6 != fibringPoint5 & fibringPoint6 != fibringPoint4 & fibring-
Point6 != fibringPoint3 & fibringPoint6 != fibringPoint2 & fibringPoint6 != fibringPoint1
& fibringPoint6 != listFuzzyPoints_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint6;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint7 >= fibringPoint1) add++;
if (fibringPoint7 >= fibringPoint2) add++;
if (fibringPoint7 >= fibringPoint3) add++;
if (fibringPoint7 >= fibringPoint4) add++;
if (fibringPoint7 >= fibringPoint5) add++;
if (fibringPoint7 >= fibringPoint6) add++;
if (fibringPoint7 >= fibringPoint10) add++;
if (fibringPoint7 >= fibringPoint11) add++;
if (fibringPoint7 >= fibringPoint12) add++;
if (add > 0 & fibringPoint7 != fibringPoint6 & fibringPoint7 != fibringPoint5 & fibring-
Point7 != fibringPoint4 & fibringPoint7 != fibringPoint3 & fibringPoint7 != fibringPoint2
& fibringPoint7 != fibringPoint1 & fibringPoint7 != listFuzzyPoints_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint7;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint8 >= fibringPoint1) add++;
if (fibringPoint8 >= fibringPoint2) add++;
if (fibringPoint8 >= fibringPoint3) add++;
if (fibringPoint8 >= fibringPoint4) add++;
if (fibringPoint8 >= fibringPoint5) add++;
if (fibringPoint8 >= fibringPoint6) add++;
if (fibringPoint8 >= fibringPoint10) add++;
if (fibringPoint8 >= fibringPoint11) add++;
if (fibringPoint8 >= fibringPoint12) add++;
if (add > 0 & fibringPoint8 != fibringPoint7 & fibringPoint8 != fibringPoint6 & fibring-
Point8 != fibringPoint5 & fibringPoint8 != fibringPoint4 & fibringPoint8 != fibringPoint3
& fibringPoint8 != fibringPoint2 & fibringPoint8 != fibringPoint1 & fibringPoint8 !=
listFuzzyPoints_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint8;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint9 >= fibringPoint1) add++;
if (fibringPoint9 >= fibringPoint2) add++;
if (fibringPoint9 >= fibringPoint3) add++;
if (fibringPoint9 >= fibringPoint4) add++;
```

```
if (fibringPoint9 >= fibringPoint5) add++;
if (fibringPoint9 >= fibringPoint6) add++;
if (fibringPoint9 >= fibringPoint10) add++;
if (fibringPoint9 >= fibringPoint11) add++;
if (fibringPoint9 >= fibringPoint12) add++;
if (add > 0 & fibringPoint9 != fibringPoint8 & fibringPoint9 != fibringPoint7 & fibring-
Point9 != fibringPoint6 & fibringPoint9 != fibringPoint5 & fibringPoint9 != fibringPoint4
& fibringPoint9 != fibringPoint3 & fibringPoint9 != fibringPoint2 & fibringPoint9 !=
fibringPoint1 & fibringPoint9 != listFuzzyPoints_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint9;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint10 >= fibringPoint1) add++;
if (fibringPoint10 >= fibringPoint2) add++;
if (fibringPoint10 >= fibringPoint7) add++;
if (fibringPoint10 >= fibringPoint8) add++;
if (fibringPoint10 >= fibringPoint9) add++;
if (fibringPoint10 >= fibringPoint4) add++;
if (fibringPoint10 >= fibringPoint5) add++;
if (fibringPoint10 >= fibringPoint6) add++;
if (add > 0 & fibringPoint10 != fibringPoint9 & fibringPoint10 != fibringPoint8 &
fibringPoint10 != fibringPoint7 & fibringPoint10 != fibringPoint6 & fibringPoint10 !=
fibringPoint5 & fibringPoint10 != fibringPoint4 & fibringPoint10 != fibringPoint3 &
fibringPoint10 != fibringPoint2 & fibringPoint10 != fibringPoint1 & fibringPoint10 !=
listFuzzyPoints_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint10;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint11 >= fibringPoint1) add++;
if (fibringPoint11 >= fibringPoint2) add++;
if (fibringPoint11 >= fibringPoint3) add++;
if (fibringPoint11 >= fibringPoint7) add++;
if (fibringPoint11 >= fibringPoint8) add++;
if (fibringPoint11 >= fibringPoint9) add++;
if (fibringPoint11 >= fibringPoint4) add++;
if (fibringPoint11 >= fibringPoint5) add++;
if (fibringPoint11 >= fibringPoint6) add++;
if (add > 0 & fibringPoint11 != fibringPoint10 & fibringPoint11 != fibringPoint9 &
fibringPoint11 != fibringPoint8 & fibringPoint11 != fibringPoint7 & fibringPoint11 !=
fibringPoint6 & fibringPoint11 != fibringPoint5 & fibringPoint11 != fibringPoint4 &
fibringPoint11 != fibringPoint3 & fibringPoint11 != fibringPoint2 & fibringPoint11 !=
fibringPoint1 & fibringPoint11 != listFuzzyPoints_buffer[0, 0])
```

```
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint11;
listFuzzyPoints_bufferEndPoint++;
}
add = 0;
if (fibringPoint12 >= fibringPoint1) add++;
if (fibringPoint12 >= fibringPoint2) add++;
if (fibringPoint12 >= fibringPoint3) add++;
if (fibringPoint12 >= fibringPoint7) add++;
if (fibringPoint12 >= fibringPoint8) add++;
if (fibringPoint12 >= fibringPoint9) add++;
if (fibringPoint12 >= fibringPoint4) add++;
if (fibringPoint12 >= fibringPoint5) add++;
if (fibringPoint12 >= fibringPoint6) add++;
if (add > 0 & fibringPoint12 != fibringPoint11 & fibringPoint12 != fibringPoint10 &
fibringPoint12 != fibringPoint9 & fibringPoint12 != fibringPoint8 & fibringPoint12 !=
fibringPoint7 & fibringPoint12 != fibringPoint6 & fibringPoint12 != fibringPoint5 &
fibringPoint12 != fibringPoint4 & fibringPoint12 != fibringPoint3 & fibringPoint12 !=
fibringPoint2 & fibringPoint12 != fibringPoint1 & fibringPoint12
!= listFuzzyPoints_buffer[0, 0])
{
listFuzzyPoints_buffer[listFuzzyPoints_bufferEndPoint, 0] = fibringPoint12;
listFuzzyPoints_bufferEndPoint++;
}
for (int fuzzyPoints = 0; fuzzyPoints < listFuzzyPoints_bufferEndPoint; fuzzyPoints++)
{
lstBox13Points.Items.Add((fuzzyPoints + 1) + "..." + "0." + listFuzzy-
Points_buffer[fuzzyPoints, 0]);
}
return (fibringPoint1);
}
public int fibringFirstPoint(int point1, int point2, int point3, int point4)
{
int fibredPoint = 0;
if (point1 <= point2 & point1 <= point3 & point1 <= point4) fibredPoint = point1;
if (point2 <= point1 & point2 <= point3 & point2 <= point4) fibredPoint = point2;
if (point3 <= point1 & point3 <= point2 & point3 <= point4) fibredPoint = point3;
if (point4 <= point1 & point4 <= point2 & point4 <= point3) fibredPoint = point4;
return (fibredPoint);
}
public int fibringSecondPoint(int point1, int point2, int point3, int point4)
{;
int fibredPoint = 0;
if ((point1 <= point2 & point1 <= point3 & point1 >= point4) | (point1 >= point2 &
point1 <= point3 & point1 <= point4) | (point1 <= point2 & point1 >= point3 & point1
<= point4)) fibredPoint = point1;
if ((point2 <= point1 & point2 <= point3 & point2 >= point4) | (point2 >= point1 &
point2 <= point3 & point2 <= point4) | (point2 <= point1 & point2 >= point3 & point2
<= point4)) fibredPoint = point2;
```

```
if ((point3 <= point2 & point3 <= point1 & point3 >= point4) | (point3 >= point2 &
point3 <= point1 & point3 <= point4) | (point3 <= point2 & point3 >= point1 & point3
<= point4)) fibredPoint = point3;
if ((point4 <= point2 & point4 <= point1 & point4 >= point3) | (point4 >= point2 &
point4 <= point1 & point4 <= point3) | (point4 <= point2 & point4 >= point1 & point4
<= point3)) fibredPoint = point4;
return (fibredPoint);
}
public int fibringThirdPoint(int point1, int point2, int point3, int point4)
{
int fibredPoint = 0;
if ((point1 <= point2 & point1 >= point3 & point1 >= point4) | (point1 >= point2 &
point1 >= point3 & point1 <= point4) | (point1 >= point2 & point1 <= point3 & point1
>= point4)) fibredPoint = point1;
if ((point2 <= point1 & point2 >= point3 & point2 >= point4) | (point2 >= point1 &
point2 >= point3 & point2 <= point4) | (point2 >= point1 & point2 <= point3 & point2
>= point4)) fibredPoint =
point2;
if ((point3 <= point2 & point3 >= point1 & point3 >= point4) | (point3 >= point2 &
point3 >= point1 & point3 <= point4) | (point3 >= point2 & point3 <= point1 & point3
>= point4)) fibredPoint = point3;
if ((point4 <= point2 & point4 >= point1 & point4 >= point3) | (point4 >= point2 &
point4 >= point1 & point4 <= point3) | (point4 >= point2 & point4 <= point1 & point4
>= point3)) fibredPoint = point4;
return (fibredPoint);
}
public int fibringMaxFirstPoint(int point1, int point2, int point3, int point4)
{
int fibredPoint = 0;
if (point1 >= point2 & point1 >= point3 & point1 >= point4) fibredPoint = point1;
if (point2 >= point1 & point2 >= point3 & point2 >= point4) fibredPoint = point2;
if (point3 >= point1 & point3 >= point2 & point3 >= point4) fibredPoint = point3;
if (point4 >= point1 & point4 >= point2 & point4 >= point3) fibredPoint = point4;
return (fibredPoint);
}
public int fibringMaxSecondPoint(int point1, int point2, int point3, int point4)
{
int fibredPoint = 0;
if ((point1 >= point2 & point1 >= point3 & point1 <= point4) | (point1 <= point2 &
point1 >= point3 & point1 >= point4) | (point1 >= point2 & point1 <= point3 & point1
>= point4)) fibredPoint = point1;
if ((point2 >= point1 & point2 >= point3 & point2 <= point4) | (point2 <= point1 &
point2 >= point3 & point2 >= point4) | (point2 >= point1 & point2 <= point3 & point2
>= point4)) fibredPoint = point2;
if ((point3 >= point2 & point3 >= point1 & point3 <= point4) | (point3 <= point2 &
point3 >= point1 & point3 >= point4) | (point3 >= point2 & point3 <= point1 & point3
>= point4)) fibredPoint = point3;
```

```
if ((point4 >= point2 & point4 >= point1 & point4 <= point3) | (point4 <= point2 &
point4 >= point1 & point4 >= point3) | (point4 >= point2 & point4 <= point1 & point4
>= point3)) fibredPoint = point4;
return (fibredPoint);
}
public int fibringMaxThirdPoint(int point1, int point2, int point3, int point4)
{
int fibredPoint = 0;
if ((point1 >= point2 & point1 <= point3 & point1 <= point4) | (point1 <= point2 &
point1 <= point3 & point1 >= point4) | (point1 <= point2 & point1 >= point3 & point1
<= point4)) fibredPoint = point1;
if ((point2 >= point1 & point2 <= point3 & point2 <= point4) | (point2 <= point1 &
point2 <= point3 & point2 >= point4) | (point2 <= point1 & point2 >= point3 & point2
<= point4)) fibredPoint = point2;
if ((point3 >= point2 & point3 <= point1 & point3 <= point4) | (point3 <= point2 &
point3 <= point1 & point3 >= point4) | (point3 <= point2 & point3 >= point1 & point3
<= point4)) fibredPoint = point3;
if ((point4 >= point2 & point4 <= point1 & point4 <= point3) | (point4 <= point2 &
point4 <= point1 & point4 >= point3) | (point4 <= point2 & point4 >= point1 & point4
<= point3)) fibredPoint = point4;
return (fibredPoint);
}
private void LBLA_Click(object sender, EventArgs e)
{
lstFirstPoint.Items.Clear();
lstSecondPoint.Items.Clear();
lstThirdPoint.Items.Clear();
lstFourthPoint.Items.Clear();
this.lbl1_Click(this.lbl1, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines1_buffer[fibringLines1_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFirstPoint.Items.Add("0." + fibringLines1_buffer[fibringLines1_bufferEndPoint, 0]);
fibringLines1_bufferEndPoint++;
}
this.lbl4_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines2_buffer[fibringLines2_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstSecondPoint.Items.Add("0." + fibringLines2_buffer[fibringLines2_bufferEndPoint,
0]);
fibringLines2_bufferEndPoint++;
}
this.lbl7_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines3_buffer[fibringLines3_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstThirdPoint.Items.Add("0." + fibringLines3_buffer[fibringLines3_bufferEndPoint, 0]);
fibringLines3_bufferEndPoint++;
}
```

```
this.lbl12_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines4_buffer[fibringLines4_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFourthPoint.Items.Add("0." + fibringLines4_buffer[fibringLines4_bufferEndPoint, 0]);
fibringLines4_bufferEndPoint++;
}
listLines();
}
private void LBLB_Click(object sender, EventArgs e)
{
this.lbl1_Click(this.lbl1, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines1_buffer[fibringLines1_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFirstPoint.Items.Add("0." + fibringLines1_buffer[fibringLines1_bufferEndPoint, 0]);
fibringLines1_bufferEndPoint++;
}
this.lbl2_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines2_buffer[fibringLines2_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstSecondPoint.Items.Add("0." + fibringLines2_buffer[fibringLines2_bufferEndPoint,
0]);
fibringLines2_bufferEndPoint++;
}
this.lbl3_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines3_buffer[fibringLines3_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstThirdPoint.Items.Add("0." + fibringLines3_buffer[fibringLines3_bufferEndPoint, 0]);
fibringLines3_bufferEndPoint++;
}
this.lbl11_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines4_buffer[fibringLines4_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFourthPoint.Items.Add("0." + fibringLines4_buffer[fibringLines4_bufferEndPoint, 0]);
fibringLines4_bufferEndPoint++;
}
listLines();
}
private void LBLC_Click(object sender, EventArgs e)
{
this.lbl3_Click(this.lbl1, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines1_buffer[fibringLines1_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFirstPoint.Items.Add("0." + fibringLines1_buffer[fibringLines1_bufferEndPoint, 0]);
fibringLines1_bufferEndPoint++;
```

```
}
this.lbl8_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{ fibringLines2_buffer[fibringLines2_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstSecondPoint.Items.Add("0." + fibringLines2_buffer[fibringLines2_bufferEndPoint,
0]);
fibringLines2_bufferEndPoint++;
} this.lbl9_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines3_buffer[fibringLines3_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstThirdPoint.Items.Add("0." + fibringLines3_buffer[fibringLines3_bufferEndPoint, 0]);
fibringLines3_bufferEndPoint++;
}
this.lbl12_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines4_buffer[fibringLines4_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFourthPoint.Items.Add("0." + fibringLines4_buffer[fibringLines4_bufferEndPoint, 0]);
fibringLines4_bufferEndPoint++;
}
listLines();
}
private void LBLD_Click(object sender, EventArgs e)
{
this.lbl3_Click(this.lbl1, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines1_buffer[fibringLines1_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFirstPoint.Items.Add("0." + fibringLines1_buffer[fibringLines1_bufferEndPoint, 0]);
fibringLines1_bufferEndPoint++;
}
this.lbl4_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines2_buffer[fibringLines2_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstSecondPoint.Items.Add("0." + fibringLines2_buffer[fibringLines2_bufferEndPoint,
0]);
fibringLines2_bufferEndPoint++;
}
this.lbl10_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines3_buffer[fibringLines3_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstThirdPoint.Items.Add("0." + fibringLines3_buffer[fibringLines3_bufferEndPoint, 0]);
fibringLines3_bufferEndPoint++;
}
this.lbl13_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
```

```
{
fibringLines4_buffer[fibringLines4_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFourthPoint.Items.Add("0." + fibringLines4_buffer[fibringLines4_bufferEndPoint, 0]);
fibringLines4_bufferEndPoint++;
}
listLines();
}
private void LBLE_Click(object sender, EventArgs e)
{
this.lbl2_Click(this.lbl1, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines1_buffer[fibringLines1_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFirstPoint.Items.Add("0." + fibringLines1_buffer[fibringLines1_bufferEndPoint, 0]);
fibringLines1_bufferEndPoint++;
}
this.lbl5_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines2_buffer[fibringLines2_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstSecondPoint.Items.Add("0." + fibringLines2_buffer[fibringLines2_bufferEndPoint,
0]);
fibringLines2_bufferEndPoint++;
}
this.lbl10_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines3_buffer[fibringLines3_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstThirdPoint.Items.Add("0." + fibringLines3_buffer[fibringLines3_bufferEndPoint, 0]);
fibringLines3_bufferEndPoint++;
}
this.lbl12_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines4_buffer[fibringLines4_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFourthPoint.Items.Add("0." + fibringLines4_buffer[fibringLines4_bufferEndPoint, 0]);
fibringLines4_bufferEndPoint++;
}
listLines();
}
private void LBLF_Click(object sender, EventArgs e)
{
this.lbl3_Click(this.lbl1, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines1_buffer[fibringLines1_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFirstPoint.Items.Add("0." + fibringLines1_buffer[fibringLines1_bufferEndPoint, 0]);
fibringLines1_bufferEndPoint++;
```

```
}
this.lbl5_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines2_buffer[fibringLines2_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstSecondPoint.Items.Add("0." + fibringLines2_buffer[fibringLines2_bufferEndPoint,
0]);
fibringLines2_bufferEndPoint++;
}
this.lbl6_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines3_buffer[fibringLines3_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstThirdPoint.Items.Add("0." + fibringLines3_buffer[fibringLines3_bufferEndPoint, 0]);
fibringLines3_bufferEndPoint++;
}
this.lbl7_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines4_buffer[fibringLines4_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFourthPoint.Items.Add("0." + fibringLines4_buffer[fibringLines4_bufferEndPoint, 0]);
fibringLines4_bufferEndPoint++;
}
listLines();
}
private void LBLG_Click(object sender, EventArgs e)
{
this.lbl7_Click(this.lbl1, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines1_buffer[fibringLines1_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFirstPoint.Items.Add("0." + fibringLines1_buffer[fibringLines1_bufferEndPoint, 0]);
fibringLines1_bufferEndPoint++;
}
this.lbl9_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines2_buffer[fibringLines2_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstSecondPoint.Items.Add("0." + fibringLines2_buffer[fibringLines2_bufferEndPoint,
0]);
fibringLines2_bufferEndPoint++;
}
this.lbl10_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines3_buffer[fibringLines3_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstThirdPoint.Items.Add("0." + fibringLines3_buffer[fibringLines3_bufferEndPoint, 0]);
fibringLines3_bufferEndPoint++;
```

```
}
this.lbl11_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines4_buffer[fibringLines4_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFourthPoint.Items.Add("0." + fibringLines4_buffer[fibringLines4_bufferEndPoint, 0]);
fibringLines4_bufferEndPoint++;
}
listLines();
}
private void LBLH_Click(object sender, EventArgs e)
{
this.lbl2_Click(this.lbl1, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines1_buffer[fibringLines1_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFirstPoint.Items.Add("0." + fibringLines1_buffer[fibringLines1_bufferEndPoint, 0]);
fibringLines1_bufferEndPoint++;
}
this.lbl7_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines2_buffer[fibringLines2_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstSecondPoint.Items.Add("0." + fibringLines2_buffer[fibringLines2_bufferEndPoint,
0]);
fibringLines2_bufferEndPoint++;
}
this.lbl8_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines3_buffer[fibringLines3_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstThirdPoint.Items.Add("0." + fibringLines3_buffer[fibringLines3_bufferEndPoint, 0]);
fibringLines3_bufferEndPoint++;
}
this.lbl13_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines4_buffer[fibringLines4_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFourthPoint.Items.Add("0." + fibringLines4_buffer[fibringLines4_bufferEndPoint, 0]);
fibringLines4_bufferEndPoint++;
}
listLines();
}
private void LBLI_Click(object sender, EventArgs e)
{
this.lbl2_Click(this.lbl1, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
```

```
{
fibringLines1_buffer[fibringLines1_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFirstPoint.Items.Add("0." + fibringLines1_buffer[fibringLines1_bufferEndPoint, 0]);
fibringLines1_bufferEndPoint++;
}
this.lbl4_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines2_buffer[fibringLines2_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstSecondPoint.Items.Add("0." + fibringLines2_buffer[fibringLines2_bufferEndPoint,
0]);
fibringLines2_bufferEndPoint++;
}
this.lbl6_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines3_buffer[fibringLines3_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstThirdPoint.Items.Add("0." + fibringLines3_buffer[fibringLines3_bufferEndPoint, 0]);
fibringLines3_bufferEndPoint++;
}
this.lbl9_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines4_buffer[fibringLines4_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFourthPoint.Items.Add("0." + fibringLines4_buffer[fibringLines4_bufferEndPoint, 0]);
fibringLines4_bufferEndPoint++;
}
listLines();
}
private void LBLJ_Click(object sender, EventArgs e)
{
this.lbl3_Click(this.lbl1, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines1_buffer[fibringLines1_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFirstPoint.Items.Add("0." + fibringLines1_buffer[fibringLines1_bufferEndPoint, 0]);
fibringLines1_bufferEndPoint++;
}
this.lbl4_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines2_buffer[fibringLines2_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstSecondPoint.Items.Add("0." + fibringLines2_buffer[fibringLines2_bufferEndPoint,
0]);
fibringLines2_bufferEndPoint++;
}
```

```
this.lbl10_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines3_buffer[fibringLines3_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstThirdPoint.Items.Add("0." + fibringLines3_buffer[fibringLines3_bufferEndPoint, 0]);
fibringLines3_bufferEndPoint++;
}
this.lbl13_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines4_buffer[fibringLines4_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFourthPoint.Items.Add("0." + fibringLines4_buffer[fibringLines4_bufferEndPoint, 0]);
fibringLines4_bufferEndPoint++;
}
listLines();
}
private void LBLK_Click(object sender, EventArgs e)
{
this.lbl4_Click(this.lbl1, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines1_buffer[fibringLines1_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFirstPoint.Items.Add("0." + fibringLines1_buffer[fibringLines1_bufferEndPoint, 0]);
fibringLines1_bufferEndPoint++;
}
this.lbl5_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines2_buffer[fibringLines2_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstSecondPoint.Items.Add("0." + fibringLines2_buffer[fibringLines2_bufferEndPoint,
0]);
fibringLines2_bufferEndPoint++;
}
this.lbl8_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines3_buffer[fibringLines3_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstThirdPoint.Items.Add("0." + fibringLines3_buffer[fibringLines3_bufferEndPoint, 0]);
fibringLines3_bufferEndPoint++;
}
this.lbl11_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines4_buffer[fibringLines4_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFourthPoint.Items.Add("0." + fibringLines4_buffer[fibringLines4_bufferEndPoint, 0]);
fibringLines4_bufferEndPoint++;
}
```

```
listLines();
}
private void LBLL_Click(object sender, EventArgs e)
{
this.lbl6_Click(this.lbl1, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines1_buffer[fibringLines1_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFirstPoint.Items.Add("0." + fibringLines1_buffer[fibringLines1_bufferEndPoint, 0]);
fibringLines1_bufferEndPoint++;
}
this.lbl11_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines2_buffer[fibringLines2_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstSecondPoint.Items.Add("0." + fibringLines2_buffer[fibringLines2_bufferEndPoint,
0]);
fibringLines2_bufferEndPoint++;
}
this.lbl12_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines3_buffer[fibringLines3_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstThirdPoint.Items.Add("0." + fibringLines3_buffer[fibringLines3_bufferEndPoint, 0]);
fibringLines3_bufferEndPoint++;
}
this.lbl13_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines4_buffer[fibringLines4_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFourthPoint.Items.Add("0." + fibringLines4_buffer[fibringLines4_bufferEndPoint, 0]);
fibringLines4_bufferEndPoint++;
}
listLines();
}
private void LBLM_Click(object sender, EventArgs e)
{
this.lbl1_Click(this.lbl1, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines1_buffer[fibringLines1_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFirstPoint.Items.Add("0." + fibringLines1_buffer[fibringLines1_bufferEndPoint, 0]);
fibringLines1_bufferEndPoint++;
}
this.lbl5_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines2_buffer[fibringLines2_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
```

```
lstSecondPoint.Items.Add("0." + fibringLines2_buffer[fibringLines2_bufferEndPoint,
0]);
fibringLines2_bufferEndPoint++;
}
this.lbl9_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines3_buffer[fibringLines3_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstThirdPoint.Items.Add("0." + fibringLines3_buffer[fibringLines3_bufferEndPoint, 0]);
fibringLines3_bufferEndPoint++;
}
this.lbl13_Click(this.lbl2, e);
for (int a = 0; a < listFuzzyPoints_bufferEndPoint; a++)
{
fibringLines4_buffer[fibringLines4_bufferEndPoint, 0] = listFuzzyPoints_buffer[a, 0];
lstFourthPoint.Items.Add("0." + fibringLines4_buffer[fibringLines4_bufferEndPoint, 0]);
fibringLines4_bufferEndPoint++;
}
listLines();
}
public void listLines()
{
if (cmBoxOperator.Text=="minimum") function=1;
if (cmBoxOperator.Text == "maximum") function =2;
if (function == 1)
{
lstLines.Items.Clear();
int samePoint = 0;
int same = 0;
for (samePoint = 0; samePoint < fibringLines1_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines1_buffer[samePoint, 0];
for (same = 0; same < fibringLines2_bufferEndPoint; same++)
{
if (myPoint <= fibringLines2_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = my-
Point;fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines1_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines1_buffer[samePoint, 0];
for (same = 0; same < fibringLines3_bufferEndPoint; same++)
{
```

```
if (myPoint <= fibringLines3_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines1_bufferEndPoint; samePoint++)

{
int myPoint = fibringLines1_buffer[samePoint, 0];
for (same = 0; same < fibringLines4_bufferEndPoint; same++)
{
if (myPoint <= fibringLines4_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines2_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines2_buffer[samePoint, 0];
for (same = 0; same < fibringLines1_bufferEndPoint; same++)
{
if (myPoint <= fibringLines1_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines2_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines2_buffer[samePoint, 0];
for (same = 0; same < fibringLines3_bufferEndPoint; same++)
{
if (myPoint <= fibringLines3_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;
fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines2_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines2_buffer[samePoint, 0];
```

```
for (same = 0; same < fibringLines4_bufferEndPoint; same++)
{
if (myPoint <= fibringLines4_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines3_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines3_buffer[samePoint, 0];
for (same = 0; same < fibringLines1_bufferEndPoint; same++)
{
if (myPoint <= fibringLines1_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;
fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines3_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines3_buffer[samePoint, 0];
for (same = 0; same < fibringLines2_bufferEndPoint; same++)
{
if (myPoint <= fibringLines2_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;
fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines3_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines3_buffer[samePoint, 0];
for (same = 0; same < fibringLines4_bufferEndPoint; same++)
{
if (myPoint <= fibringLines4_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;
fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines4_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines4_buffer[samePoint, 0];
for (same = 0; same < fibringLines1_bufferEndPoint; same++)
```

```
{
if (myPoint <= fibringLines1_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;
fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines4_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines4_buffer[samePoint, 0];
for (same = 0; same < fibringLines2_bufferEndPoint; same++)
{
if (myPoint <= fibringLines2_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;
fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines4_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines4_buffer[samePoint, 0];
for (same = 0; same < fibringLines3_bufferEndPoint; same++)
{
if (myPoint <= fibringLines3_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;
fibringLines5_bufferEndPoint++;
}
}
}
int sameCount = 0;
for (int add = 0; add < fibringLines5_bufferEndPoint; add++)
{
if (add == 0)
{
fibringLines6_buffer[0, 0] = fibringLines5_buffer[0, 0];
fibringLines6_bufferEndPoint++;
}
else
{
int myPoint = fibringLines5_buffer[add, 0];
sameCount = 0;
for (int a = 0; a < fibringLines6_bufferEndPoint; a++)
```

```
{
if (myPoint == fibringLines6_buffer[a, 0])
{
sameCount++;
}
}
if (sameCount == 0)
{
fibringLines6_buffer[fibringLines6_bufferEndPoint, 0] = myPoint;
fibringLines6_bufferEndPoint++;
}
}
}
}
}
if (function == 2)
{
lstLines.Items.Clear();
int samePoint = 0;
int same = 0;
for (samePoint = 0; samePoint < fibringLines1_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines1_buffer[samePoint, 0];
for (same = 0; same < fibringLines2_bufferEndPoint; same++)
{
if (myPoint >= fibringLines2_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines1_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines1_buffer[samePoint, 0];
for (same = 0; same < fibringLines3_bufferEndPoint; same++)
{
if (myPoint >= fibringLines3_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;
fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines1_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines1_buffer[samePoint, 0];
for (same = 0; same < fibringLines4_bufferEndPoint; same++)
```

```
{
if (myPoint >= fibringLines4_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines2_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines2_buffer[samePoint, 0];
for (same = 0; same < fibringLines1_bufferEndPoint; same++)
{
if (myPoint >= fibringLines1_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;
fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines2_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines2_buffer[samePoint, 0];
for (same = 0; same < fibringLines3_bufferEndPoint; same++)
{
if (myPoint >= fibringLines3_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines2_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines2_buffer[samePoint, 0];
for (same = 0; same < fibringLines4_bufferEndPoint; same++)
{
if (myPoint >= fibringLines4_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;
fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines3_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines3_buffer[samePoint, 0];
```

```
for (same = 0; same < fibringLines1_bufferEndPoint; same++)
{
if (myPoint >= fibringLines1_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines3_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines3_buffer[samePoint, 0];
for (same = 0; same < fibringLines2_bufferEndPoint; same++)
{
if (myPoint >= fibringLines2_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;
fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines3_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines3_buffer[samePoint, 0];
for (same = 0; same < fibringLines4_bufferEndPoint; same++)
{
if (myPoint >= fibringLines4_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines4_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines4_buffer[samePoint, 0];
for (same = 0; same < fibringLines1_bufferEndPoint; same++)
{
if (myPoint >= fibringLines1_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;
fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines4_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines4_buffer[samePoint, 0];
for (same = 0; same < fibringLines2_bufferEndPoint; same++)
```

```
{
if (myPoint >= fibringLines2_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;
fibringLines5_bufferEndPoint++;
}
}
}
for (samePoint = 0; samePoint < fibringLines4_bufferEndPoint; samePoint++)
{
int myPoint = fibringLines4_buffer[samePoint, 0];
for (same = 0; same < fibringLines3_bufferEndPoint; same++)
{
if (myPoint >= fibringLines3_buffer[same, 0])
{
fibringLines5_buffer[fibringLines5_bufferEndPoint, 0] = myPoint;
fibringLines5_bufferEndPoint++;
}
}
}
int sameCount = 0;
for (int add = 0; add < fibringLines5_bufferEndPoint; add++)
{
if (add == 0)
{
fibringLines6_buffer[0, 0] = fibringLines5_buffer[0, 0];
fibringLines6_bufferEndPoint++;
}
else
{
int myPoint = fibringLines5_buffer[add, 0];
sameCount = 0;
for (int a = 0; a < fibringLines6_bufferEndPoint; a++)
{
if (myPoint == fibringLines6_buffer[a, 0])
{
sameCount++;
}
}
if (sameCount == 0)
{
fibringLines6_buffer[fibringLines6_bufferEndPoint, 0] = myPoint;
fibringLines6_bufferEndPoint++;
}
}
}
}
for (int a = 0; a < fibringLines6_bufferEndPoint; a++)
```

```
{
lstLines.Items.Add((a + 1) + ”...” + ”0.” + fibringLines6_buffer[a, 0]);
}
}
}
}
```

BÖLÜM 5

Sonuç ve Öneriler

Programın ilk aşaması olarak adlandırabileceğimiz Galois cisimlerinden projektif düzlemler elde etme kısmında, mertebesi asal olan düzlemler için başarılı sonuçlar elde edildi. Uygulama sayesinde elde edilen nokta, doğru ve üzerinde bulunma bağıntısı kümelerindeki elemanların doğruluğu sınandı. Programda, mertebesi bir asal sayının birden büyük bir kuvveti şeklinde olan kümeler ele alındığında nokta ve doğru kümeleri elde edilebilmesine rağmen üzerinde bulunma bağıntısı kümesinde bazı hatalarla karşılaşıldı ve bu kısım çalışmadan çıkarıldı.

Üçüncü bölümde yer alan (Kuijkeen, 1999) tarafından hazırlanmış olan fiber Fano düzlemi örneği programa aktarılarak örnekte yer alan aynı başlangıç üyelik dereceleri kullanıldı ve yapılan programlama ile (Kuijkeen, 1999) çalışmasındaki aynı sonuçlar elde edildi. Programın ayrıca, fiber Fano düzleminin üyelik derecelerinin değiştirilerek yeni fiber düzlemler elde etmeye olanak sağladığı görüldü.

Dördüncü bölümde PG(2,3) düzlemi üzerine ilk defa bir fiber projektif düzlem inşa edildi ve örneği verildi. Daha sonra bu projektif düzlem için programlama yapıldı, program üzerinde üyelik derecelerinin değiştirilerek birçok yeni fiber düzlem elde edilebileceği gibi bir takım kombinatöryel özellikleri hakkında bilgi verdiği gözlemlendi.

Bu çalışmadan hareketle, çalışmanın mertebesi bir asal sayının birden büyük bir kuvveti şeklinde olan projektif düzlemlerin üzerinde bulunma bağıntısını verebilecek şekilde geliştirilebileceği, daha büyük düzlemler üzerine fiber projektif düzlemler inşa eden yeni çalışmaların yapılabileceği ve fiber düzlemlerin üyelik derecelerinin atanması işinin belirli şartlara bağlanmasıyla oluşabilecek durumların incelenerek bazı çıktılara ulaşılabilineceği düşünülmektedir.

KAYNAKLAR DİZİNİ

Hirschfeld, J.W.P., 1979, Projective Geometries Over Finite Fields, Clarendon Press, 474 p.

Kaya, R., 2005, Projektif Geometri, Osmangazi Üniversitesi Yayınları, 392 s.

Kuijken L.1999, Fuzzy projective geometries, EUSFLAT-ESTYLY Joint Conf., 429-432