BURSA ULUDAĞ ÜNİVERSİTESİ BİLGİSAYAR

MÜHENDİSLİĞİ BÖLÜMÜ

2022 – 2023 Eğitim Öğretim Yılı Güz Öğretimi

Görsel Programlama Ödevi 2

ADI SOYADI: Adnan Topçu

NUMARA:032190007

e-mail:

032190007@ogr.uludag.edu.tr

SORU1:

24 (5X5 - 1) tane düğme

yaratarak bunları çalışma alanını kaplar hale getiriniz. Düğmeler kare

biçiminde olmalı ve üzerlerinde 1, 2, 3, … biçiminde sayılar bulunmalıdır.

(Dolayısıyla çalışma alanı da kare biçiminde olmalı ve

boyutlandırılamamalıdır). Bu düğmelerin ortasında bir düğmelik boş bir alan vardır.

Düğmenin üzerine tıklandığında eğer o düğme uygun bir pozisyondaysa (yani boş

alanın dört komşusundan biriyse) tıklanan düğmenin o boş bölgeye

taşınabilmesini sağlayınız. Başlangıçta düğmeler rastgele bir biçimde

dizilmişlerdir. Oyunun amacı onları 1-24 arasında düzenli olarak dizip boşluğun

en altta sağda kalmasını sağlamaktır.

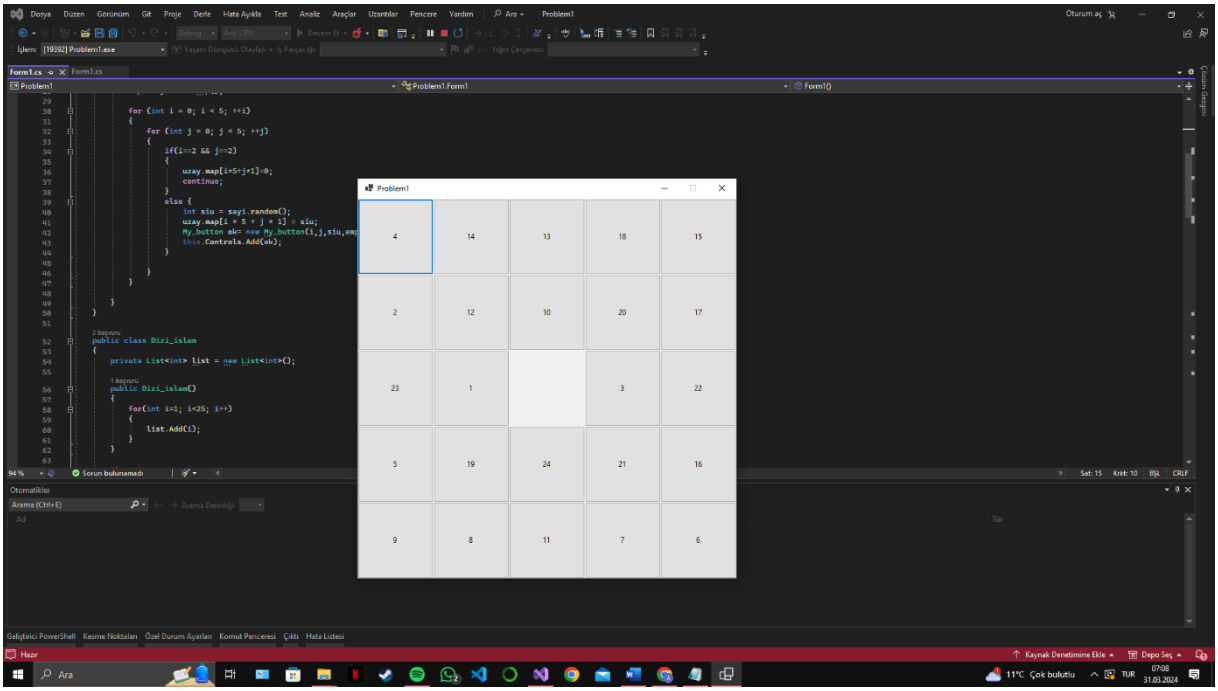
Kaynak kod:

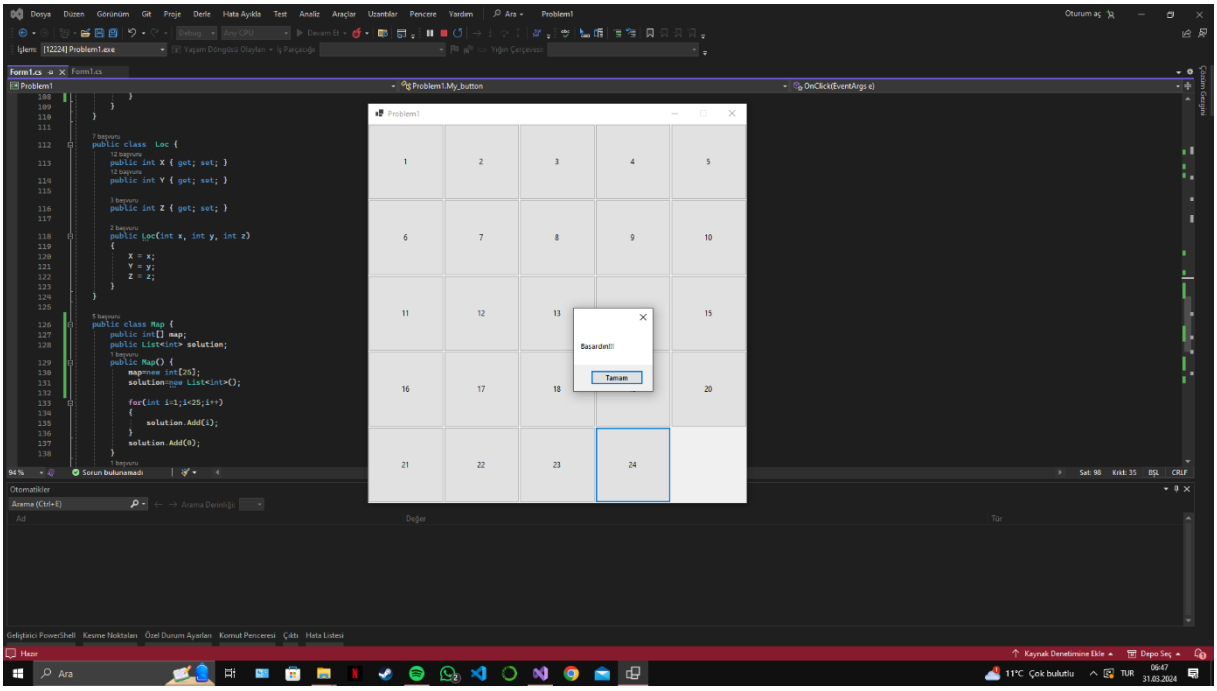
```
using System;
using System.CodeDom.Compiler;
using System.Data;
using System.Drawing;
using System.Security.Cryptography.X509Certificates;
using System.Security.Policy;
using System.Windows.Forms;
namespace Problem1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {

            this.StartPosition = FormStartPosition.CenterScreen;
            this.FormBorderStyle = FormBorderStyle.Fixed3D;
            this.MaximizeBox=false;
            this.Text = "Problem1";
            this.ClientSize = new Size(600, 600);

            Dizi_islem sayi = new Dizi_islem();
            Loc empty = new Loc(2, 2 ,0);
            Map uzay = new Map();

            for (int i = 0; i < 5; ++i)
            {
                for (int j = 0; j < 5; ++j)
                {
                    if(i==2 && j==2)
                    {
                        uzay.map[i*5+j*1]=0;
```

```
                        continue;
                    }
                    else {
                        int siu = sayi.random();
                        uzay.map[i * 5 + j * 1] = siu;
                        My_button ok= new My_button(i,j,siu,empty,uzay);
                        this.Controls.Add(ok);
                    }

                }
            }

        }
    }

    public class Dizi_islem
    {
        private List<int> list = new List<int>();

        public Dizi_islem()
        {
            for(int i=1; i<25; i++)
            {
                list.Add(i);
            }
        }

        public int random()
        {
            Random rnd = new Random();
            int random = rnd.Next(0,list.Count);
            random = list[random];
            list.Remove(random);
            return random;
        }
    }

    public class My_button : Button
    {
        public Loc konum ;
        public Loc empty;
        public Map map;

        public My_button(int x, int y, int k, Loc f,Map m)
        {
            this.Text = string.Format("{0}", k);
            this.Size = new Size(120, 120);
            this.Location = new Point(x * 120, y * 120);
            this.konum= new Loc(x, y,k);
            this.empty = f;
            this.map= m;

        }
        protected override void OnClick(EventArgs e)
        {
            if ((Math.Abs(konum.X - empty.X) == 1 && konum.Y == empty.Y) ||
                (Math.Abs(konum.Y - empty.Y) == 1 && konum.X == empty.X))
            {
                map.yer_degistirme(konum.X, konum.Y, konum.Z, empty.X, empty.Y,
empty.Z);
                int tempX = konum.X;
                int tempY = konum.Y;
                konum.X = empty.X;
                konum.Y = empty.Y;
                empty.X = tempX;
                empty.Y = tempY;
```

```
                this.Location = new Point(konum.X * 120, konum.Y * 120);
            }
            if (map.exp())
            {
                MessageBox.Show("Başardın!!!");
            }
        }
    }

    public class  Loc {
        public int X { get; set; }
        public int Y { get; set; }

        public int Z { get; set; }

        public Loc(int x, int y, int z)
        {
            X = x;
            Y = y;
            Z = z;
        }
    }

    public class Map {
        public int[] map;
        public List<int> solution;
        public Map() {
            map=new int[25];
            solution=new List<int>();

            for(int i=1;i<25;i++)
            {
                solution.Add(i);
            }
            solution.Add(0);
        }
        public bool exp()
        {
            for (int i = 0; i < 25; i++)
            {
                if (solution[i] != map[i])
                {
                    return false;
                }
            }
            return true;
        }

        public void yer_degistirme(int x1, int y1, int z1, int x2, int y2, int z2)
        {
            map[x2 * 5 + y2 * 1] = z1;
            map[x1 * 5 + y1 * 1] = z2;

        }

    }
}
```

Kod ekran görüntüsü:





Kod açıklama:

Form1: Bir Windows Forms formunu temsil eder. Formun yüklendiği anda, belirli bir boyutta ve belirli özelliklerle (orta konumda, sabit boyutlu(600,600), maksimize bottonu bloke etilmiştir) oluşturulur. Form yüklenirken, bir dizi işlem yapılır (Dizi_islem) ve bir harita (Map) oluşturulur. Ardından, 5x5 bir alan oluşturulur ve her hücreye rasgele sayılar atanır. Bu hücrelerden biri boş bırakılır ve hareket ettirilebilir.

Dizi_islem sınıfı rastgele sayılar üretmek için bir sınıf. 1'den 24'e kadar sayıları bir listeye ekler ve ardından listeden rasgele bir sayı seçer ve listeden çıkarır.

My_button sınıfı button sınıfından türetilmiştir . Windows Forms düğmelerini temsil eder. Her düğme bir hücreyi temsil eder. Oluşturulduğunda, konumu (x, y koordinatları), içeriği (rastgele sayı), boş konumu (empty) ve haritayı alır. Bir düğme tıklandığında, boş hücreyle yan yana ise, konumlarını değiştirir ve haritayı günceller.

Loc: Bir konum (x, y koordinatları) ve değer (z) içeren bir sınıf.

Map sınıfı bir haritayı temsil eder. Harita, 5x5'lik bir alanı depolamak için bir dizi kullanır. Ayrıca, doğru çözümü takip etmek için bir çözüm listesi bulundurur. Harita, tahtanın düzgün bir şekilde düzenlenip düzenlenmediğini kontrol eden bir 'exp()' metoduna ve iki hücre arasında değer değişimini sağlayan 'yer_degistirme()' metoduna sahiptir.

Soru 2: Birinci sorudaki programı değiştirerek pencereyi boyutlandırılabilir hale getiriniz. Pencere boyutlandırıldığında düğmelerin de yeniden ayarlanması gerekmektedir. Artık çalışma alanı ve dolayısıyla düğmeler kare biçiminde olmak zorunda değildir.

Kod:

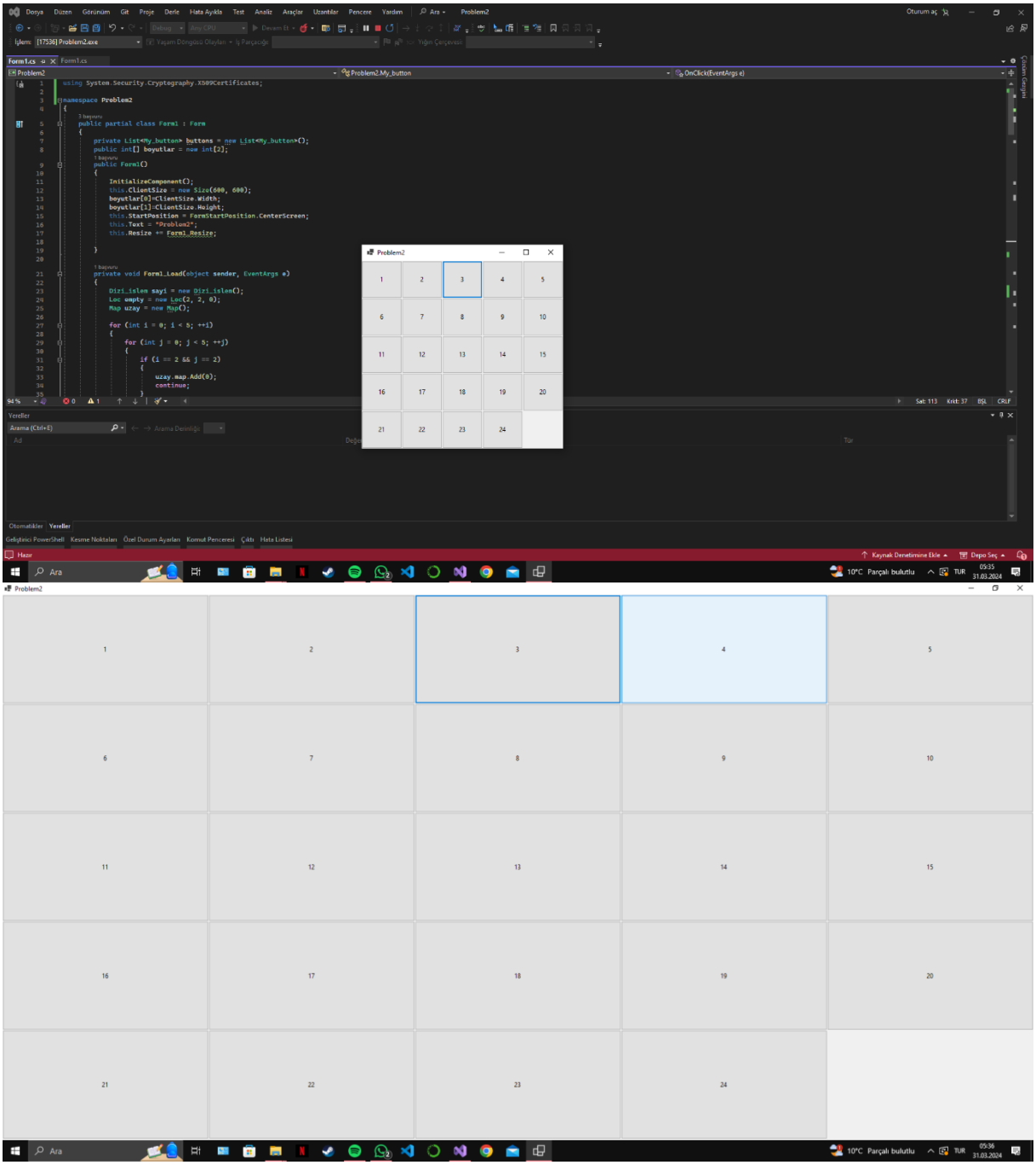
```
using System.Diagnostics.CodeAnalysis;
using System.Security.Cryptography.X509Certificates;

namespace Problem2
{
    public partial class Form1 : Form
    {
        private List<My_button> buttons = new List<My_button>();
        public int[] boyutlar = new int[2];
        public Form1()
        {
            InitializeComponent();
            this.ClientSize = new Size(600, 600);
            boyutlar[0]=ClientSize.Width;
            boyutlar[1]=ClientSize.Height;
            this.StartPosition = FormStartPosition.CenterScreen;
            this.Text = "Problem2";
            this.Resize += Form1_Resize;

        }

        private void Form1_Load(object sender, EventArgs e)
        {
            Dizi_islem sayi = new Dizi_islem();
            Loc empty = new Loc(2, 2, 0);
            Map uzay = new Map();

            for (int i = 0; i < 5; ++i)
            {
                for (int j = 0; j < 5; ++j)
                {
                    if (i == 2 && j == 2)
                    {
                        uzay.map.Add(0);
                        continue;
                    }
                    else
                    {
                        int siu = sayi.random();
                        uzay.map.Add(siu);
                        My_button ok = new My_button(i, j, siu, empty, uzay,boyutlar);
                        buttons.Add(ok);
                        this.Controls.Add(ok);
                    }

                }
            }
        }
```

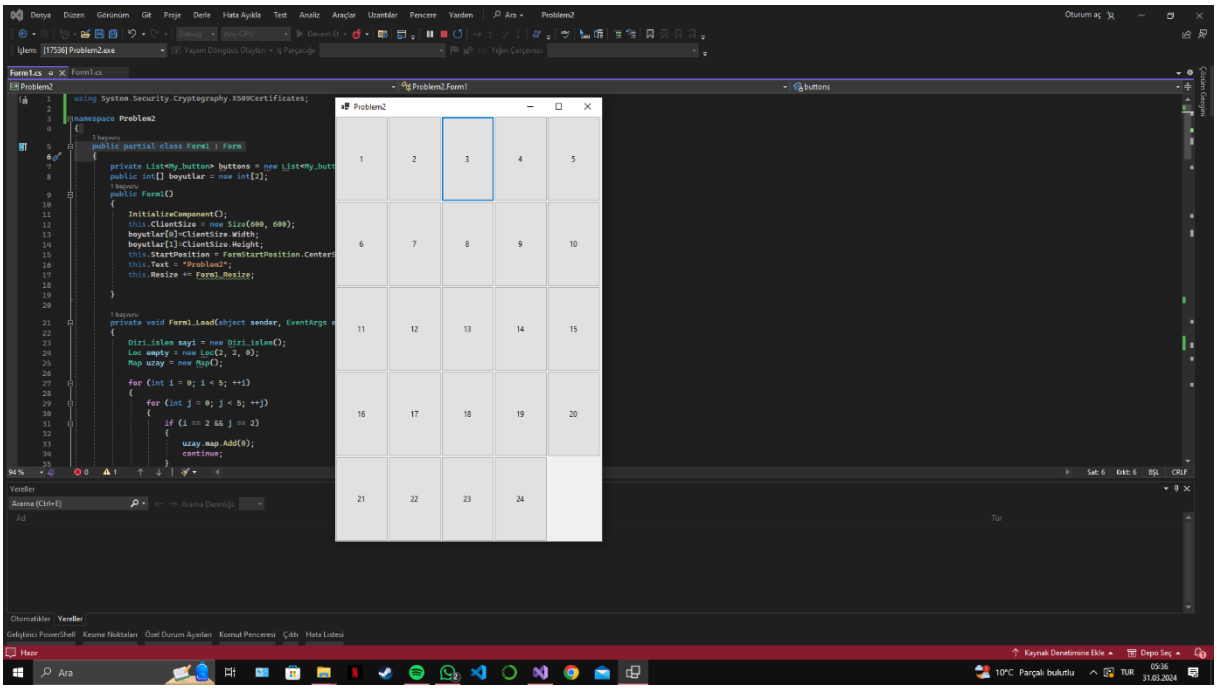
```
        private void Form1_Resize(object sender, EventArgs e)
        {
            boyutlar[0]=ClientSize.Width;
            boyutlar[1]=ClientSize.Height;
            ResizeButtons();
        }

        private void ResizeButtons()
        {
            int buttonwidth= ClientSize.Width / 5;
            int buttonheight = ClientSize.Height / 5;
            foreach (My_button button in buttons)
            {
                button.Size = new Size(buttonwidth, buttonheight);
                button.Location = new Point(button.konum.X * buttonwidth,
button.konum.Y * buttonheight);
            }
        }
    }

    public class Dizi_islem
    {
        private List<int> list = new List<int>();

        public Dizi_islem()
        {
            for (int i = 1; i < 25; i++)
            {
                list.Add(i);
            }
        }

        public int random()
        {
            Random rnd = new Random();
            int random = rnd.Next(0, list.Count);
            random = list[random];
            list.Remove(random);
            return random;
        }
    }

    public class My_button : Button
    {
        public Loc konum;
        public Loc empty;
        public Map map;
        public int[] boyut;

        public My_button(int x, int y, int k, Loc f, Map m, int[] boyut)
        {
            this.Text = string.Format("{0}", k);
            this.Size = new Size(boyut[0] / 5, boyut[1] / 5);
            this.Location = new Point(x * boyut[0] / 5, y* boyut[1] / 5);
            this.konum = new Loc(x, y, k);
            this.empty = f;
            this.map = m;
            this.boyut = boyut;
        }
        protected override void OnClick(EventArgs e)
        {
            if ((Math.Abs(konum.X - empty.X) == 1 && konum.Y == empty.Y) ||
                (Math.Abs(konum.Y - empty.Y) == 1 && konum.X == empty.X))
            {
```

```
                this.map.yer_degistirme(konum.X, konum.Y, konum.Z, empty.X, empty.Y,
empty.Z);
                int tempX = konum.X;
                int tempY = konum.Y;
                konum.X = empty.X;
                konum.Y = empty.Y;
                empty.X = tempX;
                empty.Y = tempY;
                this.Location = new Point(konum.X * boyut[0] / 5, konum.Y * boyut[1] /
5);
            }

            if (map.exp())
            {
                MessageBox.Show("Başardın!!!");

            }
        }
    }

    public class Loc
    {
        public int X { get; set; }
        public int Y { get; set; }

        public int Z { get; set; }

        public Loc(int x, int y, int z)
        {
            X = x;
            Y = y;
            Z = z;
        }
    }

    public class Map
    {
        public List<int> map;
        public List<int> solution;
        public Map()
        {
            map = new List<int>();
            solution = new List<int>();

            for (int i = 1; i < 25; i++)
            {
                solution.Add(i);
            }
            solution.Add(0);
        }
        public bool exp()
        {
            for (int i = 0; i < 25; i++)
            {
                if (solution[i] == map[i])
                {
                    continue;
                }
                else
                {
                    return false;
                }
            }
            return true;
        }
```

```
        public void yer_degistirme(int x1, int y1, int z1, int x2, int y2, int z2)
        {
            map[x2 * 5 + y2 * 1] = z1;
            map[x1 * 5 + y1 * 1] = z2;
        }
    }
}
```

Ekran görüntüsü:





Kod açıklama:

Form1 bir Windows Forms formunu temsil eder. Formun yüklendiği anda, belirli bir boyutta (600x600) ve belirli özelliklerle (orta konumda, yeniden boyutlandırılabilir) oluşturulur. Form yüklendiğinde, bir dizi işlem yapılır (Dizi_islem) ve bir harita (Map) oluşturulur. Ayrıca, formun yeniden boyutlandırılması durumunda düğmelerin boyutlarını ve konumlarını güncellemek için bir olay dinleyici eklenir.

Form1_Resize metottu formun yeniden boyutlandırılması durumunda çağrılan bir yöntemdir. Yeni boyutlar alınır ve düğmelerin boyutları ve konumları yeniden ayarlanır.

My_button sınıfı bir önceki örnekte olduğu gibi, Windows Forms düğmelerini temsil eder. Ancak bu sefer, düğmelerin boyutları ve konumları dinamik olarak formun boyutlarına bağlı olarak belirlenir.

Map ve Loc sınıfları önceki örnekle aynıdır.