

Bölüm 6

C# Grafik-Dosya İşlemleri

Dr. Öğr. Üyesi Murat TAŞYÜREK (kayubmprogramlama1@gmail.com)

27 Mart 2024

Kayseri Üniversitesi, Bilgisayar Mühendisliği Bölümü

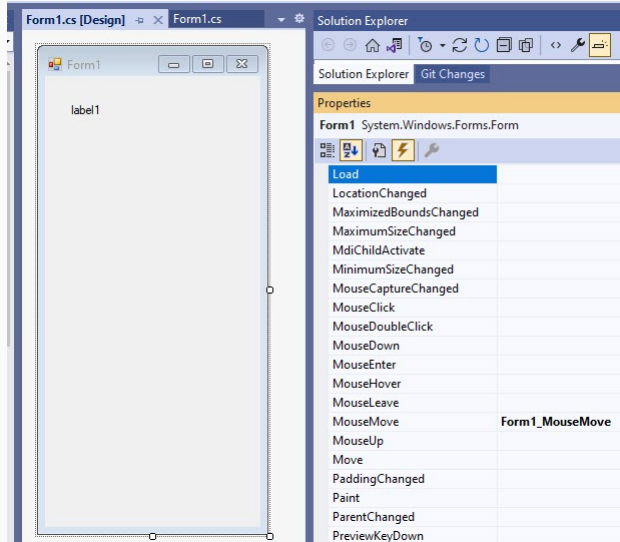
- Windows Form Mouse Hareketlerini Yakalama
- Mouse İle Şekil Çizme
- StreamWriter İle Yazma
- StreamReader İle Okuma
- Ödev

- **MouseEnter:** Mouse imleci bir kontrolün üzerine geldiğinde tetiklenir.
- **MouseLeave:** Mouse imleci bir kontrolün üzerinden çıkıp başka bir yere gittiğinde tetiklenir.
- *MouseMove:* Mouse imleci bir kontrolün üzerindeyken hareket ettiğinde tetiklenir.
- **MouseDown:** Mouse düğmesi basıldığında tetiklenir.

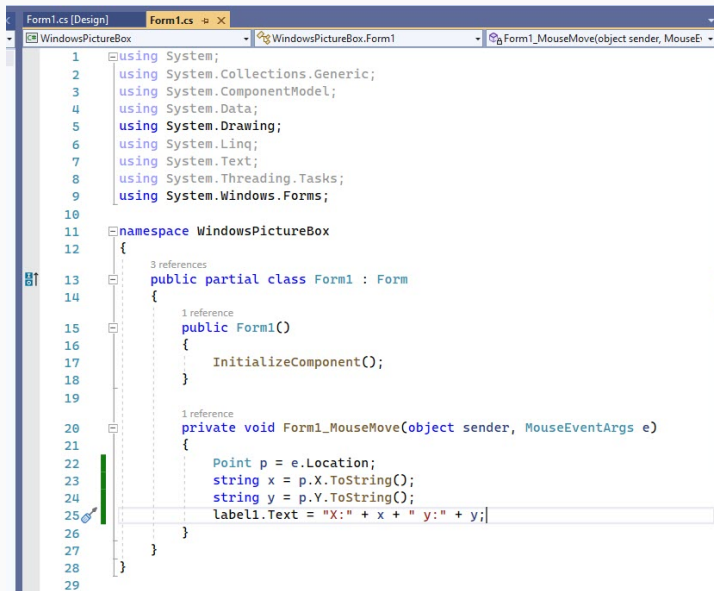
- **MouseUp:** Mouse düğmesi bırakıldığında tetiklenir.
- **MouseClicked:** Mouse düğmesine basılıp bırakıldığında tetiklenir.
- **MouseDoubleClick:** Mouse düğmesine çift tıkladığınızda tetiklenir.
- **MouseWheel:** Mouse tekerleği hareket ettirildiğinde tetiklenir.

- Mouse hareketini yakalayıp hareket ettiğini konumları ekrana yazdırmak istiyoruz.
- Bir adet label
- Mouse Fonksiyonları
- Mouse hareket edince çalışacak fonksiyon

Windows Form Tasarım Ekranı (Mouse Fonksiyonları)

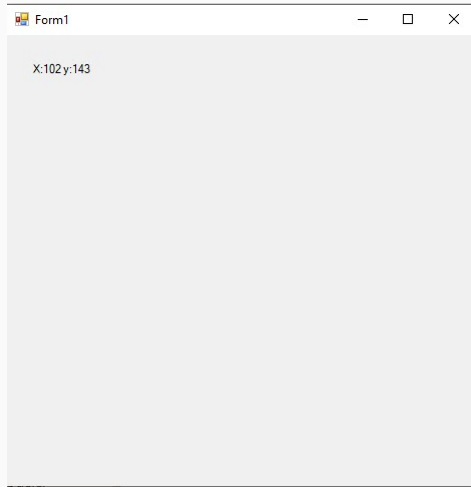


Windows Form Kaynak Kodu



```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 namespace WindowsPictureBox
12 {
13     3 references
14     public partial class Form1 : Form
15     {
16         1 reference
17         public Form1()
18         {
19             InitializeComponent();
20         }
21
22         1 reference
23         private void Form1_MouseMove(object sender, MouseEventArgs e)
24         {
25             Point p = e.Location;
26             string x = p.X.ToString();
27             string y = p.Y.ToString();
28             label1.Text = "X:" + x + " y:" + y;
29         }
30     }
31 }
```

Windows Form Çıktı

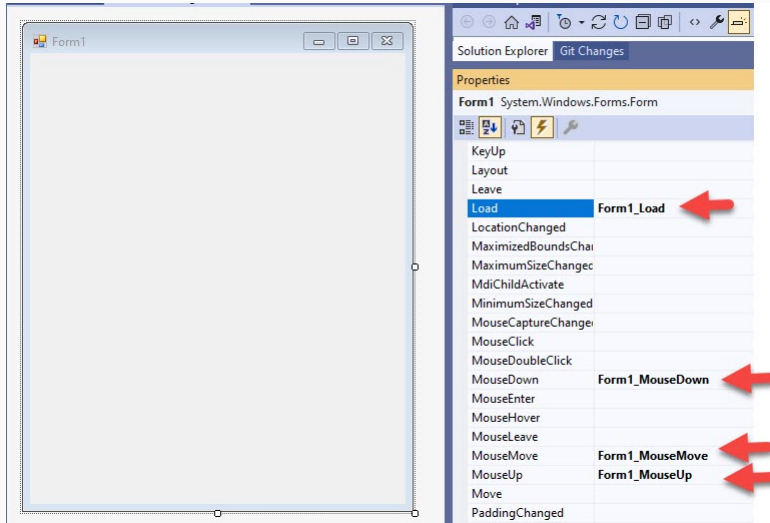


- **Mouse hareketi ile şekil çizdirelim.**
- Sıfırdan proje açıyoruz
- **MouseMove, MouseUp ve MouseDown** fonksiyonlarını kullanacağız.
- MouseDown fareyle basıldığında MouseUp ise fareden çekildiğinde çalışır.
- Mouse ile çizilen resmi ekrana çizmek.
- Formun arka planı mavi olsun.

C# Graphics, Point, Pen sınıfı

- **C# Graphics sınıfı**, .NET tabanlı masaüstü uygulamalarda grafikler oluşturmak, çizim yapmak, metin eklemek ve diğer çeşitli grafik işlemlerini gerçekleştirmek için kullanılan bir sınıftır.
- **C# Graphics sınıfı**, bir çizim yüzeyi üzerinde çeşitli grafik nesneleri oluşturmanıza ve bunlarla etkileşimde bulunmanıza olanak tanır.
- **C# Point sınıfı**, bir grafik nesnesinin (genellikle bir çizim yüzeyinde) belirli bir konumunu belirtmek için kullanılan bir sınıftır. Bu sınıf, x ve y koordinatlarından oluşan bir noktayı temsil eder.
- **C# Pen sınıfı**, çizim yüzeylerinde çizim yaparken kalem özelliklerini belirlemek ve çizgi stillerini kontrol etmek için kullanılan bir sınıftır

Windows Form Tasarım Ekranı (Fonksiyonlar)



Windows Form Kaynak Kodu

```
11 namespace MouseSekilCizdirme
12 {
13     3 references
14     public partial class Form1 : Form
15     {
16         1 reference
17         public Form1()
18         {
19             InitializeComponent();
20         }
21         bool cizimDevamEdecekmi = false; //çizim devam edecek mi
22         int baslangicX, baslangicY; //başlangıç noktası
23         1 reference
24         private void Form1_Load(object sender, EventArgs e)
25         {
26             this.BackColor = Color.Blue; //formun arka planını set ettik.
27         }
28         1 reference
29         private void Form1_MouseDown(object sender, MouseEventArgs e)
30         {
31             cizimDevamEdecekmi = true;
32             baslangicX = e.Location.X;
33             baslangicY = e.Location.Y;
34         }
35         1 reference
36         private void Form1_MouseUp(object sender, MouseEventArgs e)
37         {
38             cizimDevamEdecekmi = false;
39         }
40         1 reference
41         private void Form1_MouseMove(object sender, MouseEventArgs e)
42         {
43             Graphics g = this.CreateGraphics();
44             Pen p = new Pen(Color.Yellow, 5); // kalem oluşturduk. Renk ve kalınlık
45             Point point1 = new Point(baslangicX, baslangicY); //nokta oluşturduk önceki nokta
46             Point point2 = new Point(e.X, e.Y); //mevcut nokta
47             if (cizimDevamEdecekmi == true)
48             {
49                 g.DrawLine(p, point1, point2); //çizdirdik.
50                 baslangicX = e.X; //önceki noktanın X'ini güncelledik.
51                 baslangicY = e.Y; //önceki noktanın Y'sini güncelledik.
52             }
53         }
54     }
55 }
```

Windows Form Çıktı



Dosya Nedir?

- Bilgisayar bilimlerinde "**dosya**", bilgisayar sistemindeki verilerin saklandığı birim veya veri yapısıdır.
- **Bir dosya**, belirli bir dosya adı altında depolanan bilgilerin bir araya getirilmiş halidir.
- **Dosyalar**, çeşitli veri türlerini (metin, resim, video, ses vb.) depolayabilir ve bu verilere erişim sağlar.
- Örneğin, bir metin dosyası metin verilerini içerirken, bir resim dosyası görüntü verilerini içerir.
- **Dosyalar**, bilgisayar kullanıcıları tarafından oluşturulabilir, düzenlenebilir, taşınabilir ve silinebilir. Ayrıca, bir bilgisayar programı tarafından da oluşturulabilir ve yönetilebilir.

- C# da dosya işlemleri için yaygın olarak **StreamWriter** ve **StreamReader** kullanılır.
- Bu kütüphaneleri kullanmak için Projeye **System.IO** namespace'nin ekli olması gerekir.
- Açılan **stream** dosyalar ile işlem bittikten sonra **kapatılması** gerekir.
- Using komutu bu işlemleri otomatik yapar.
- **System.IO.FileMode.** ile hangi amaçla kullanılacağı belirtilir.

- **C# StreamWriter sınıfı**, metin tabanlı dosyalara veri yazmak için kullanılan bir sınıftır.
- **StreamWriter**, .NET Framework tarafından sağlanır ve metin dosyalarına veri yazmak için kullanılır.
- Bu sınıf, dosyaya metin verileri yazmak için çeşitli yöntemler sağlar ve genellikle dosya oluşturma, yazma ve güncelleme işlemleri için kullanılır.
- Yeni bir dosya oluşturabilir veya varolan bir dosyayı açarak içine veri yazabilir.

- **Veri Yazma:** **StreamWriter.Write()** ve **StreamWriter.WriteLine()** gibi yöntemler, dosyaya metin verileri yazmak için kullanılır. **Write()** yöntemi, belirtilen veriyi dosyaya yazar, **WriteLine()** yöntemi ise belirtilen veriyi dosyaya yazar ve ardından bir satır sonu karakteri ekler.
- **Dosya Kapatma ve Kaynak Temizleme:** **StreamWriter.Dispose()** yöntemi veya `using` bloğu kullanılarak **StreamWriter** nesnesinin kaynakları serbest bırakılır ve dosya kapatılır. Bu, dosya kaynaklarının verimli bir şekilde yönetilmesini sağlar.

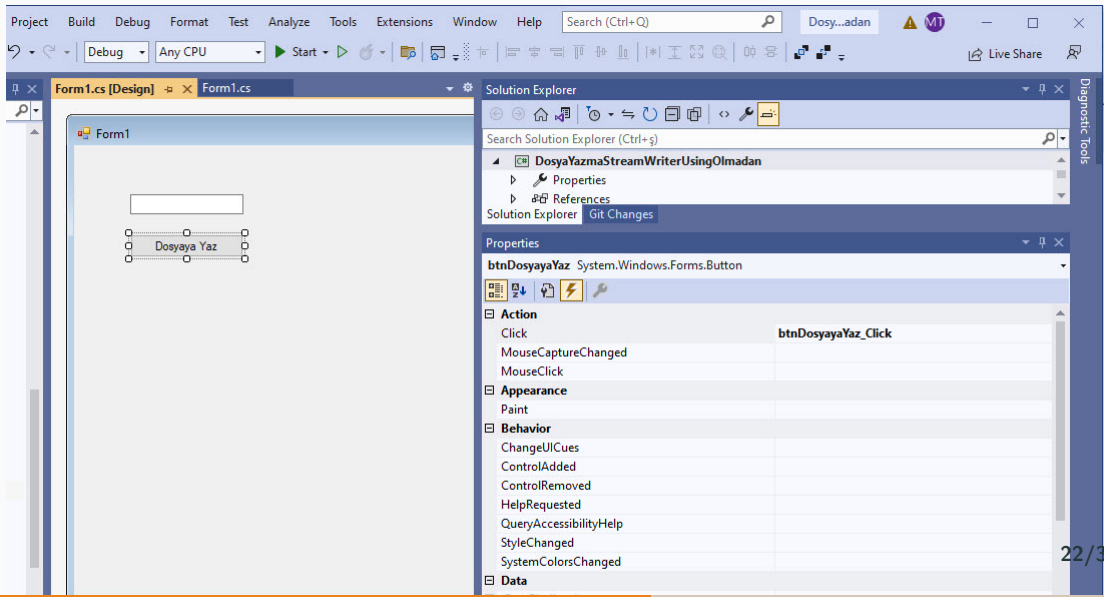
- **C# StreamReader sınıfı**, metin tabanlı dosyaları okumak için kullanılan bir sınıftır.
- **StreamReader sınıfı**, dosyadan veri okuma işlemlerini kolaylaştırmak için .NET Framework tarafından sağlanır.
- **StreamReader sınıfı**, bir dosyadan karakter tabanlı verileri okumak için kullanılan bir araçtır ve genellikle metin dosyalarını satır satır okumak için kullanılır.
- **StreamReader sınıfı**, Metin dosyalarını açar, okur ve dosyanın sonuna kadar verileri okuyabilir.

- **Satır Okuma: `StreamReader.ReadLine()`** yöntemi, dosyadan bir satır metni okur ve bu metni bir dize olarak döndürür. Bu yöntem genellikle bir döngü içinde kullanılarak dosyanın tüm satırları okunabilir.
- **Karakter Okuma: `StreamReader.Read()`** yöntemi, dosyadan bir karakter okur ve bu karakterin Unicode kodunu döndürür. Bu yöntem dosyanın herhangi bir yerinden karakter tabanlı okuma yapmak için kullanılabilir.
- **Kapatma ve Kaynak Temizleme: `StreamReader.Dispose()`** yöntemi veya `using` bloğu kullanılarak `StreamReader` nesnesinin kaynakları serbest bırakılır ve dosya kapatılır. Bu, dosya kaynaklarının verimli bir şekilde yönetilmesini sağlar.

- **C# FileStream sınıfı**, bir dosya üzerinde veri okuma ve yazma işlemlerini gerçekleştirmek için kullanılan bir sınıftır.
- **C# FileStream sınıfı**, .NET Framework tarafından sağlanır ve dosya işlemleri için düşük seviyeli bir arabirim sağlar.
- **FileStream sınıfı**, bir dosyaya doğrudan erişim sağlayarak dosya içeriğini okuma, yazma, güncelleme ve diğer dosya işlemlerini gerçekleştirmek için kullanılır.

- Windows Forms uygulaması oluşturalım
- Bir adet textbox, bir adet buton olsun
- Buton'a tıklayınca metin kutusunda yazılan ifadeler **E dizininde dosya.txt isimli dosyaya** kaydetsin

Windows Form Tasarım Ekranı

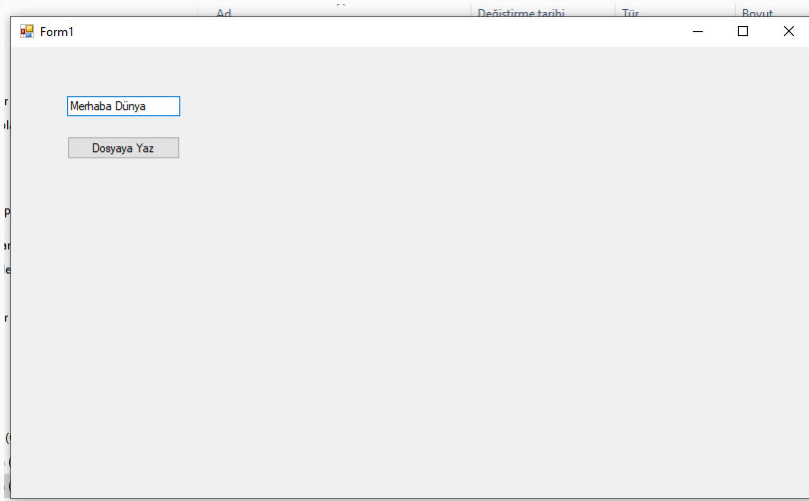


Windows Form Kaynak Kodu

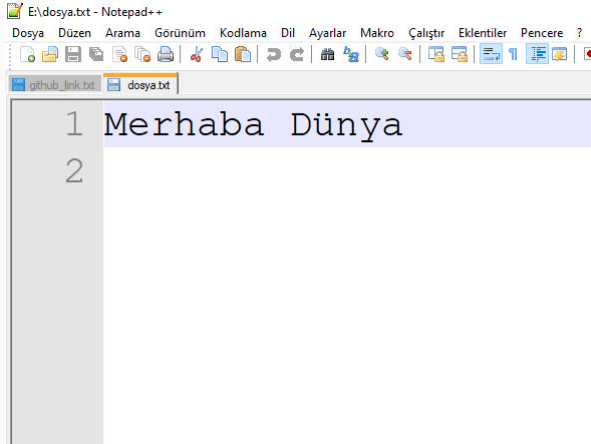
```
Form1.cs [Design]  Form1.cs -> X
DosyaYazmaStreamWriterUsingOlmadan -> DosyaYazmaStreamWriterUsingOlmadan.Form1 -> textBox1

7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10 using System.IO;
11
12 namespace DosyaYazmaStreamWriterUsingOlmadan
13 {
14     3 references
15     public partial class Form1 : Form
16     {
17         1 reference
18         public Form1()
19         {
20             InitializeComponent();
21         }
22
23         1 reference
24         private void btnDosyayaYaz_Click(object sender, EventArgs e)
25         {
26             FileStream fs = new FileStream(@"E:\\dosya.txt", FileMode.Open);
27             StreamWriter sr = new StreamWriter(fs, Encoding.UTF8);
28             sr.WriteLine(textBox1.Text);
29             sr.Close();
30             fs.Close();
31         }
32     }
33 }
```

Windows Form Çıktı



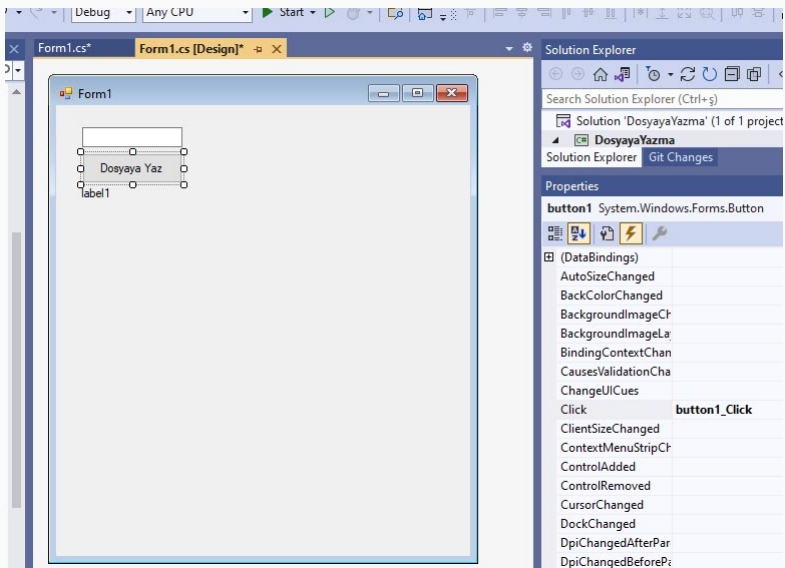
Windows Form Çıktı



Uygulama 4

- Metin kutusunda yazılan ifadeler input.txt isimli dosyaya kaydetsin
- Dosya yoksa oluştursun
- Var ise üzerine ekleyerek devam etsin
- Yazma işlemi bitince metin kutusunu temizlesin, Label'e de bilgi yazsın
- Dosya işlemi bitince otomatik olarak dosya kapatma tetiklensin.

Windows Form Tasarım Ekranı

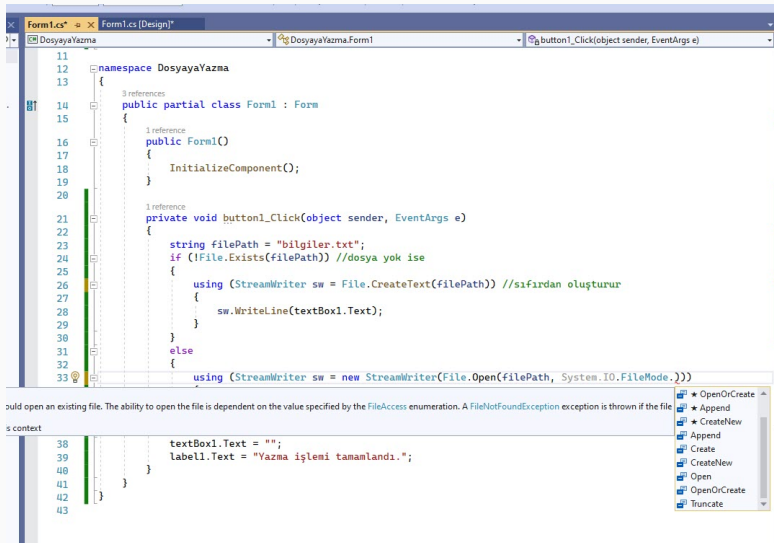


Windows Form Kaynak Kodu

```
3 references
public partial class Form1 : Form
{
    1 reference
    public Form1()
    {
        InitializeComponent();
    }

    1 reference
    private void btnDosyayaYaz_Click(object sender, EventArgs e)
    {
        string fileName = "input.txt";
        if (!File.Exists(fileName))
        {
            FileStream fs = new FileStream(fileName, FileMode.OpenOrCreate);
            using (StreamWriter sr = new StreamWriter(fs))
            {
                sr.WriteLine(textBox1.Text);
            }
        }
        else
        {
            FileStream fs = new FileStream(fileName, FileMode.Append);
            using (StreamWriter sr = new StreamWriter(fs))
            {
                sr.WriteLine(textBox1.Text);
            }
        }
        textBox1.Text = string.Empty;
        label1.Text = "Yazma işlemi başarılı";
    }
}
```

Windows Form FileMode.*



The screenshot shows a Visual Studio window with a Windows Form application named "DosyayaYazma". The code is in a file named "Form1.cs" and is in Design mode. The code defines a partial class `Form1` that inherits from `Form`. It includes a constructor `Form1()` that calls `InitializeComponent()`. A click event handler `button1_Click` is defined, which writes the text from `textBox1` to a file named `bilgiler.txt`. The code uses `FileMode.*` to create or append to the file. A context menu is open, showing the available `FileMode` options: `OpenOrCreate`, `Append`, `CreateNew`, `Create`, `Open`, `Truncate`, and `OpenOrCreate`.

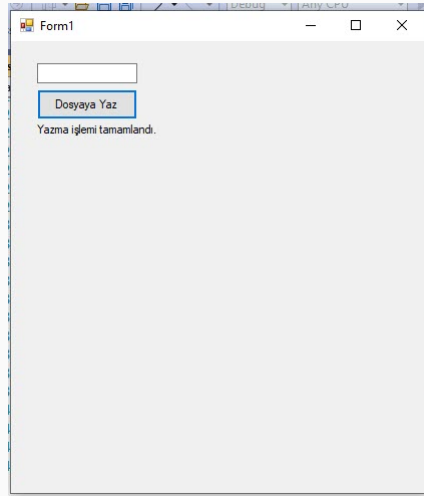
```
11
12 namespace DosyayaYazma
13 {
14     3 references
15     public partial class Form1 : Form
16     {
17         1 reference
18         public Form1()
19         {
20             InitializeComponent();
21         }
22
23         1 reference
24         private void button1_Click(object sender, EventArgs e)
25         {
26             string filePath = "bilgiler.txt";
27             if (!File.Exists(filePath)) //dosya yok ise
28             {
29                 using (StreamWriter sw = File.CreateText(filePath)) //sıfırdan oluşturur
30                 {
31                     sw.WriteLine(textBox1.Text);
32                 }
33             }
34             else
35             {
36                 using (StreamWriter sw = new StreamWriter(File.Open(filePath, System.IO.FileMode.*)))
37                 {
38                     textBox1.Text = "";
39                     label1.Text = "Yazma işlemi tamamlandı.";
40                 }
41             }
42         }
43     }
44 }
```

could open an existing file. The ability to open the file is dependent on the value specified by the `FileAccess` enumeration. A `FileNotFoundException` exception is thrown if the file

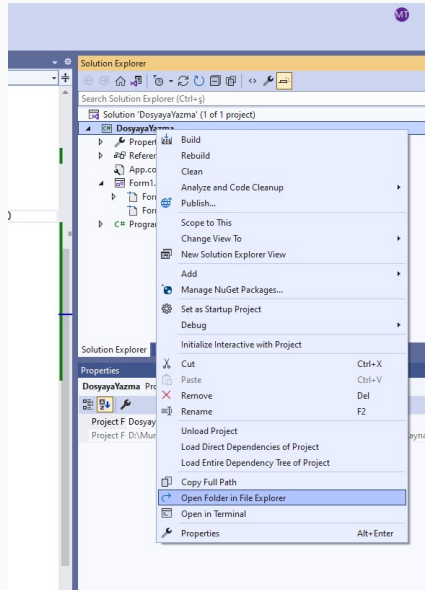
s context

- OpenOrCreate
- Append
- CreateNew
- Create
- Open
- Truncate
- OpenOrCreate

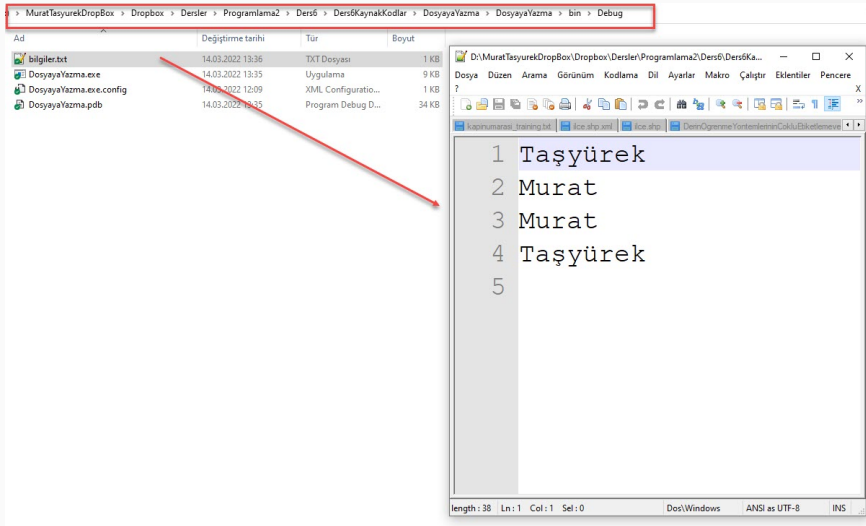
Windows Form Çıktı



Windows Form Dizini Açma

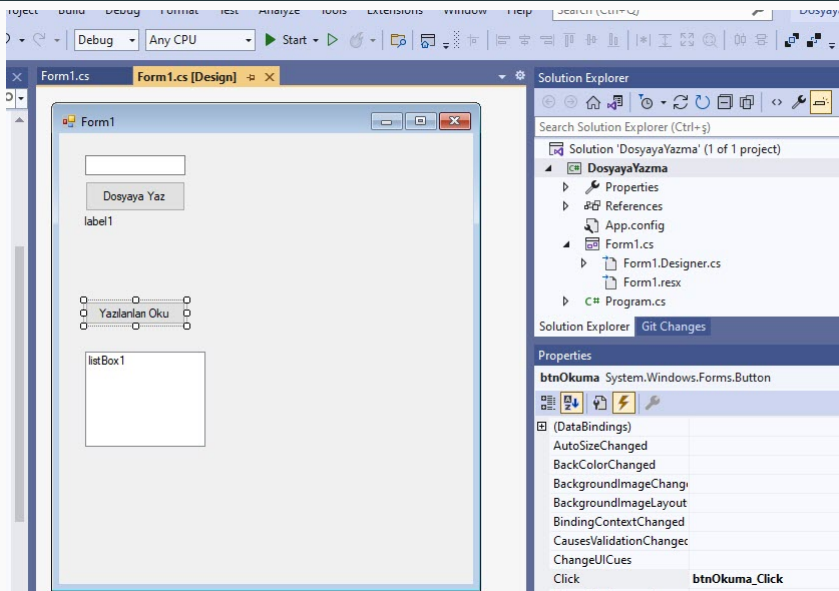


Windows Form Oluşturulan Dosya

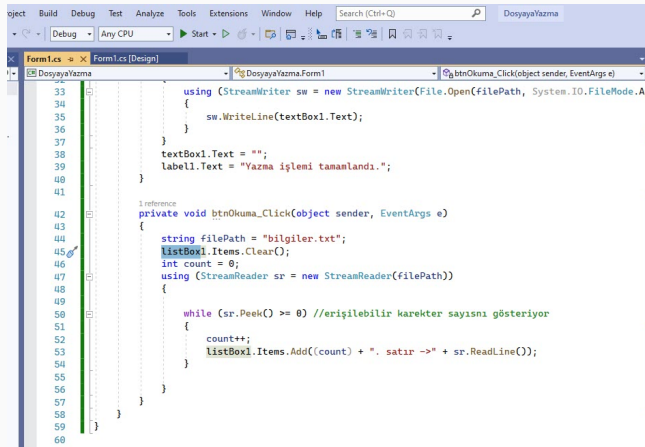


- Uygulama 4'de oluşturulan ve içine yazılan değerler okunup ekranda bulunan listBox içine satır satır yazsın
- Buton ve listBox ekleyelim

Windows Form Tasarım Ekranı

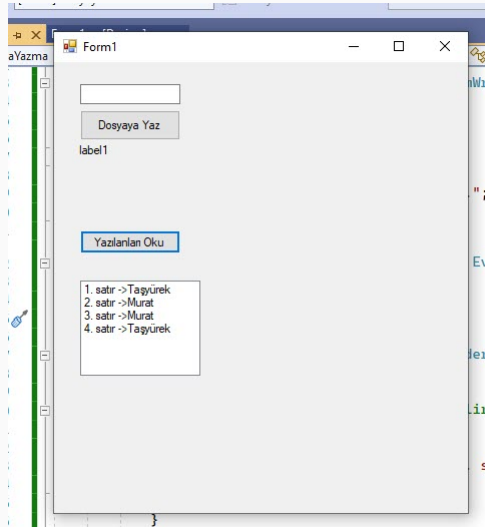


Windows Form Kaynak Kodu



```
33
34
35     using (StreamWriter sw = new StreamWriter(File.Open(filePath, System.IO.FileMode.Append)))
36     {
37         sw.WriteLine(textBox1.Text);
38     }
39     textBox1.Text = "";
40     label1.Text = "Yazma işlemi tamamlandı.";
41 }
42
43 1 reference
44 private void btnOkuma_Click(object sender, EventArgs e)
45 {
46     string filePath = "bilgiler.txt";
47     listBox1.Items.Clear();
48     int count = 0;
49     using (StreamReader sr = new StreamReader(filePath))
50     {
51         while (sr.Peek() >= 0) //erişilebilir karakter sayısını gösteriyor
52         {
53             count++;
54             listBox1.Items.Add((count) + ". satır ->" + sr.ReadLine());
55         }
56     }
57 }
58
59
60
```

Windows Form Çıktı



- Randevu defteri tasarlamanız isteniyor.
- Her gün **8** adet randevu olmalı.
- **Numarasının sonu tek olanlar 08.00-12.00, 13.00-17.00 arası, çift olanlar ise 08.30-12.30, 13.30-17.30 arasına göre yapmalı.**
- Form arayüzü üzerinden bilgiler girilmeli.
- Dosyada randevular tutulmalı.
- **Aynı saate randevu verilmek istendiğinde dolu olduğunu en yakın boş saatin hangi saat olduğunu göstermeli.**
- Özet butununa tıklayınca hangi günler hangi saatlerde kimin randevu bilgisinin olduğu topluca gösterilmeli.
- Özgün ödev olmalı.
- 3 Nisan 07.59'a kadar gönderilenler 8+2 puan ilave (kayubmprogramlama1@gmail.com). Bu tarihten sonra vizeye kadar gönderenler 8 puan.