



Bilgisayar Programcılığı

Öğr. Gör. Cansu AYVAZ GÜVEN

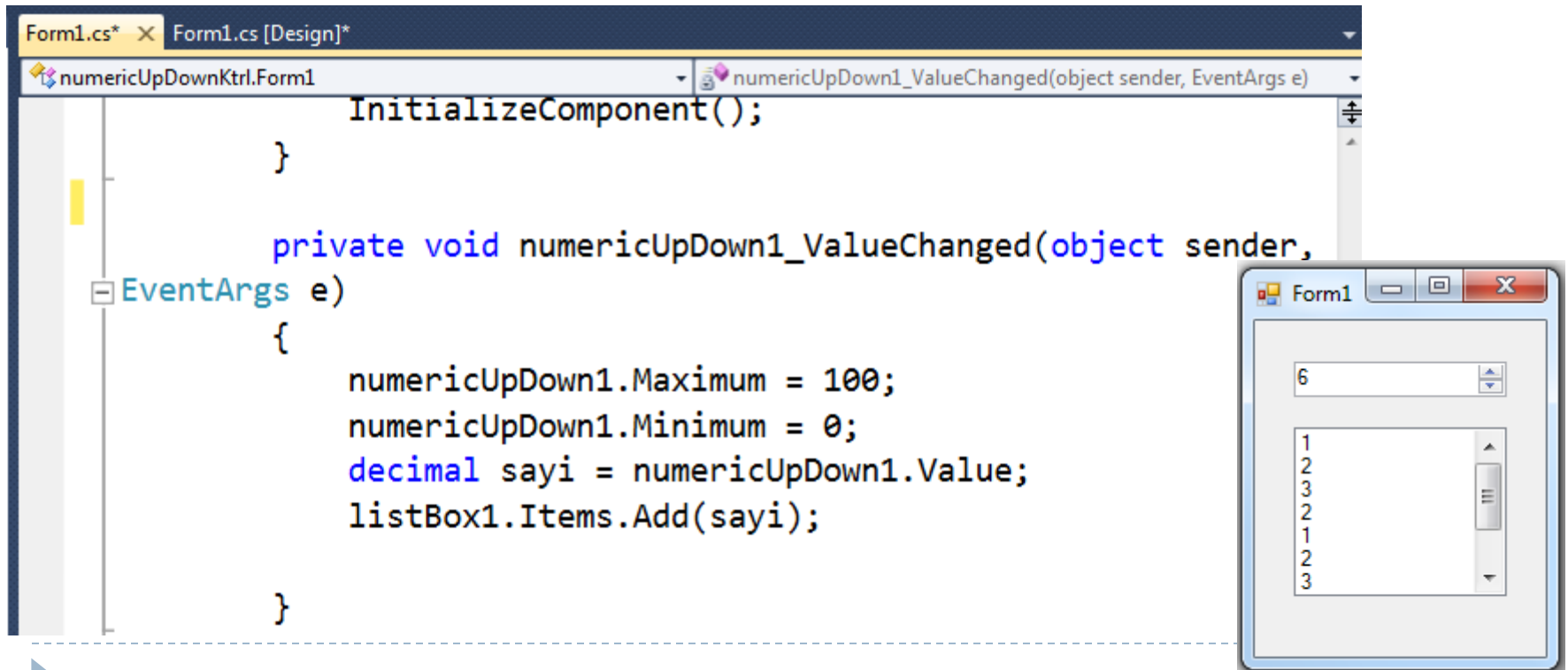
KONTROLLER-II

- ▶ NumericUpDown Kontrolü
- ▶ PictureBox Kontrolü
- ▶ RichTextBox Kontrolü
- ▶ ToolTip Kontrolü
- ▶ Web Browser Kontrolü
- ▶ ContextMenuStrip Kontrolü
- ▶ MenuStrip Kontrolü
- ▶ ToolStrip Kontrolü
- ▶ Timer Kontrolü
- ▶ ProgressBar Kontrolü



NumericUpDown Kontrolü: Aşağı yukarı okları kullanılarak değerin artırılıp azaltıldığı ve sayısal bir değer alınan bir kontroldür.

- ▶ Value: içerisindeki değer decimal tipindedir. Kontrol içindeki değeri okumamızı sağlar.
- ▶ Minimum: Girilebilecek minimum değer
- ▶ Maximum: Girilebilecek maximum değer

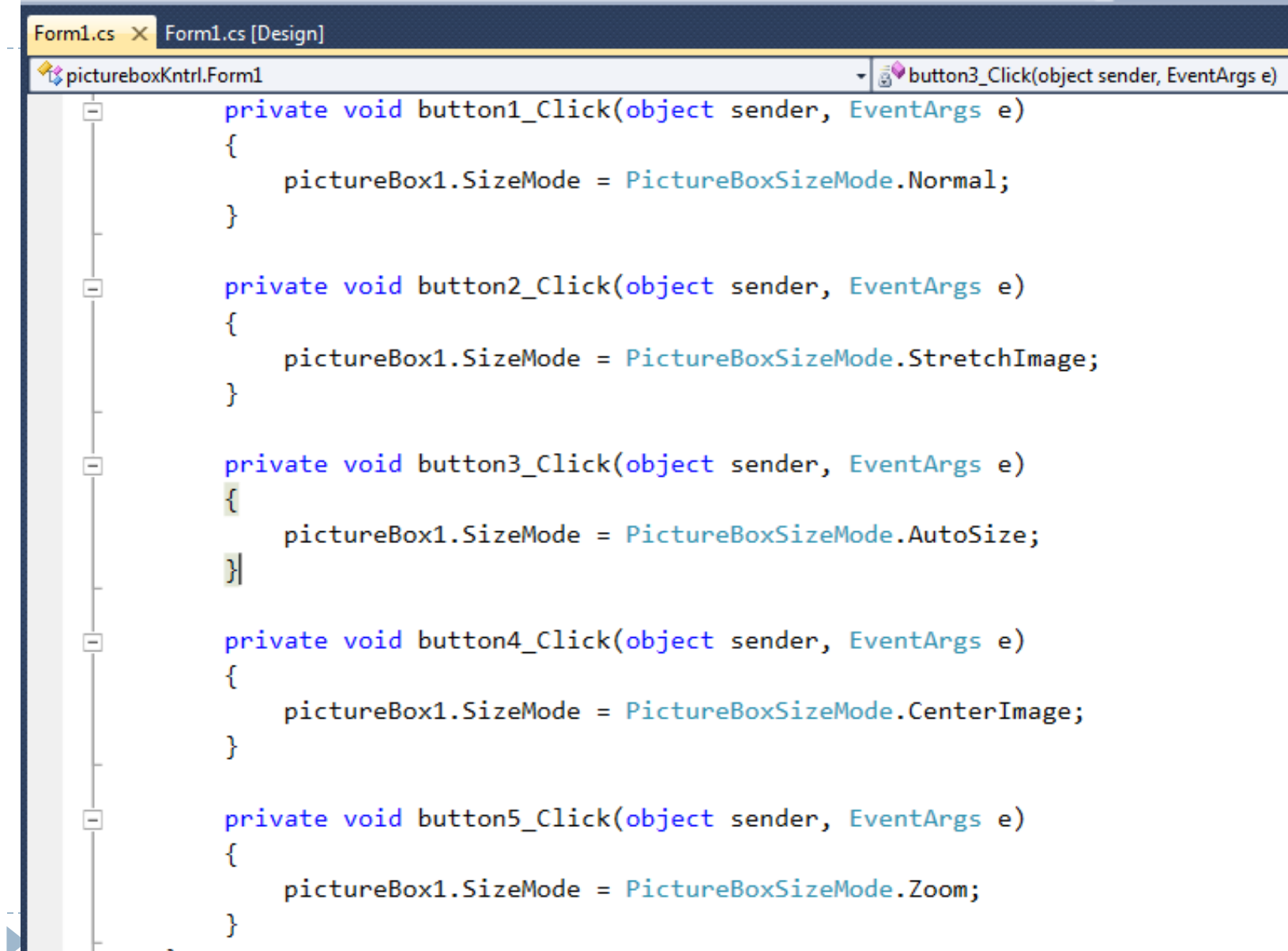


The image shows a Visual Studio IDE with a C# code file named `Form1.cs` in Design mode. The code defines a `numericUpDownKtrl` class that inherits from `Form1`. It includes an `InitializeComponent()` method and a `numericUpDown1_ValueChanged` event handler. The event handler sets the `Maximum` property to 100, the `Minimum` property to 0, converts the `Value` to a `decimal` named `sayi`, and adds it to `listBox1.Items`.

```
Form1.cs* x Form1.cs [Design]*
numericUpDownKtrl:Form1
numericUpDown1_ValueChanged(object sender, EventArgs e)
InitializeComponent();
}
private void numericUpDown1_ValueChanged(object sender,
EventArgs e)
{
    numericUpDown1.Maximum = 100;
    numericUpDown1.Minimum = 0;
    decimal sayi = numericUpDown1.Value;
    listBox1.Items.Add(sayi);
}
```

To the right of the code editor is a preview window titled `Form1` showing the visual representation of the `numericUpDown1` control. It consists of a text box displaying the value `6`, with up and down arrow buttons on the right side. Below the text box is a list box containing the numbers `1, 2, 3, 2, 1, 2, 3`.

PictureBox Kontrolü



The screenshot shows the Visual Studio IDE with a C# file named Form1.cs in Design view. The code defines five click events for buttons, each setting the SizeMode of pictureBox1 to a different value. The code is as follows:

```
Form1.cs x Form1.cs [Design]
pictureBoxKntnl.Form1 button3_Click(object sender, EventArgs e)

private void button1_Click(object sender, EventArgs e)
{
    pictureBox1.SizeMode = PictureBoxSizeMode.Normal;
}

private void button2_Click(object sender, EventArgs e)
{
    pictureBox1.SizeMode = PictureBoxSizeMode.StretchImage;
}

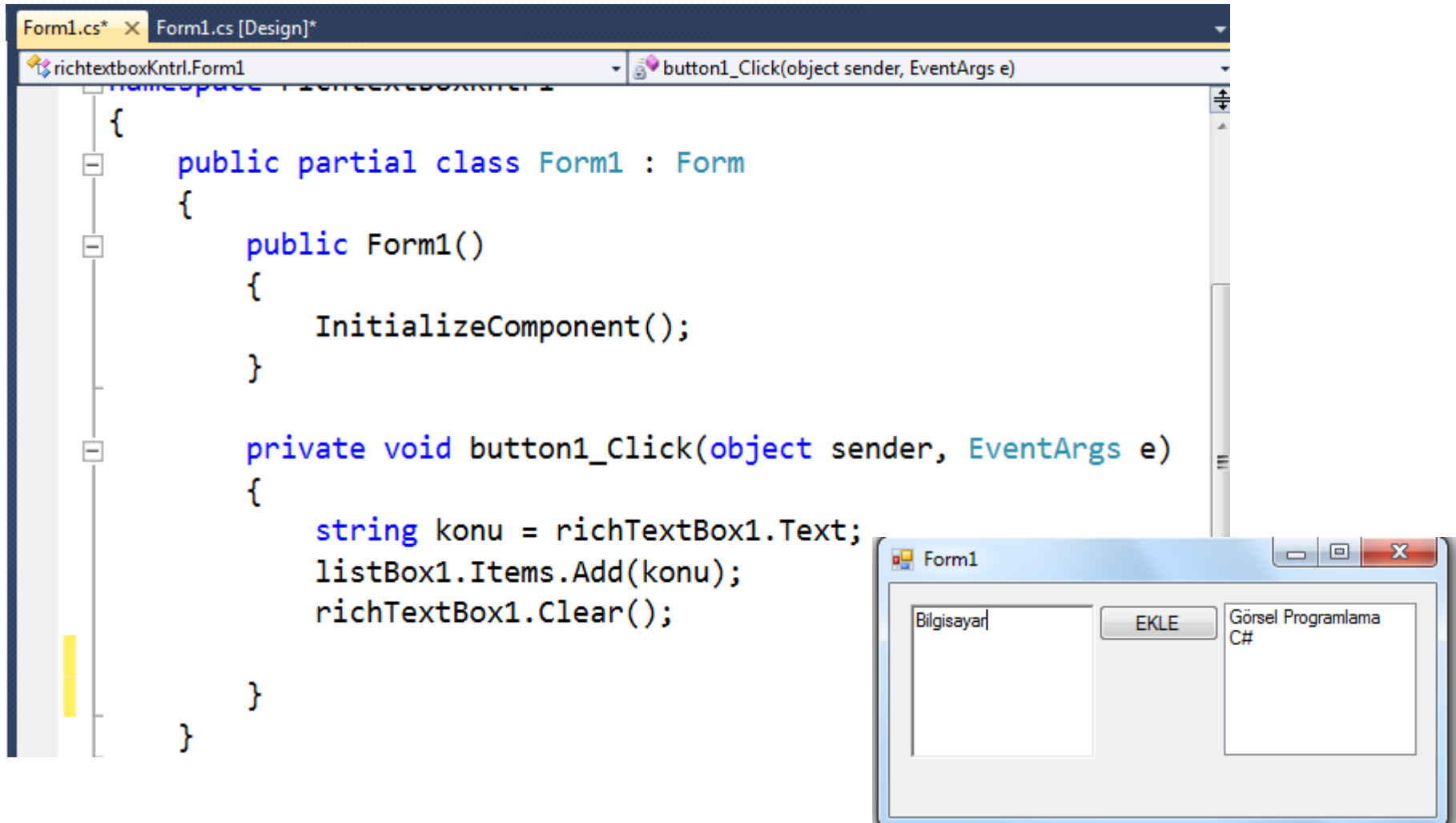
private void button3_Click(object sender, EventArgs e)
{
    pictureBox1.SizeMode = PictureBoxSizeMode.AutoSize;
}

private void button4_Click(object sender, EventArgs e)
{
    pictureBox1.SizeMode = PictureBoxSizeMode.CenterImage;
}

private void button5_Click(object sender, EventArgs e)
{
    pictureBox1.SizeMode = PictureBoxSizeMode.Zoom;
}
```



RichTextBox Kontrolü: Textbox ile aynı özelliklere sahiptir. Bir nevi MultiLine özelliği açık olan Textbox demektir.



The image displays a Visual Studio development environment. The main window shows the code for `Form1` in `Form1.cs`. 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 also defined, which takes the text from `richTextBox1`, adds it to `listBox1`, and clears `richTextBox1`.

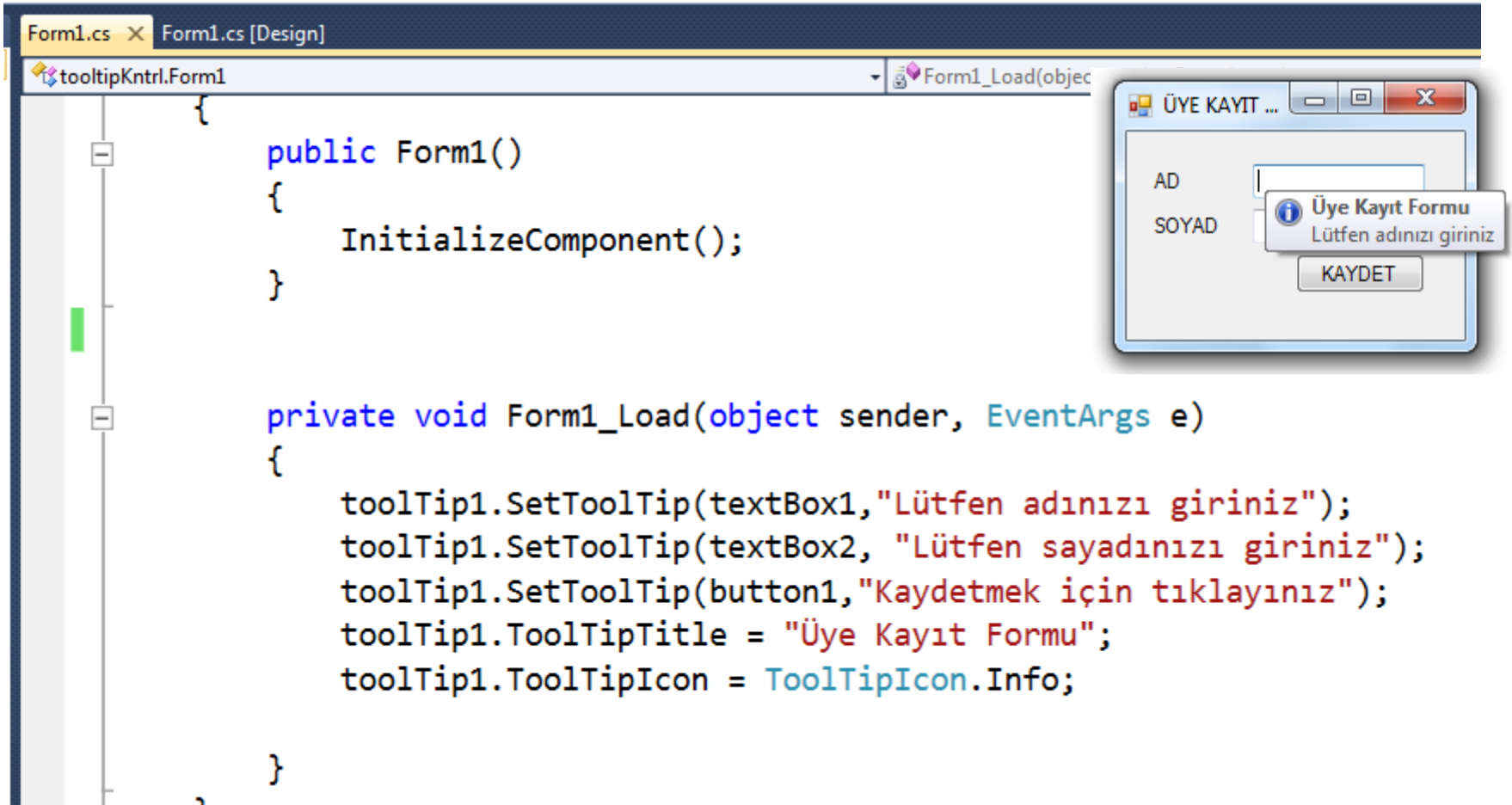
```
Form1.cs* x Form1.cs [Design]*
namespace RichTextBoxKntrl
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            string konu = richTextBox1.Text;
            listBox1.Items.Add(konu);
            richTextBox1.Clear();
        }
    }
}
```

In the bottom right corner, a preview of the `Form1` window is shown. It features a text box containing the text "Bilgisayar", a button labeled "EKLE", and a list box containing the text "Görsel Programlama C#".

ToolTip Kontrolü:

Kontrollere ipucu eklemek için kullanılır. Genellikle mouse imleci kontrol üzerine geldiğinde çalışır



The image shows a Visual Studio IDE window with two panes. The left pane displays the C# code for a form named `Form1`. The right pane shows a preview of the application window titled "ÜYE KAYIT ...".

Code Snippet:

```
Form1.cs x Form1.cs [Design]
tooltipKntrl.Form1
Form1_Load(objec

{

    public Form1()
    {
        InitializeComponent();
    }

    private void Form1_Load(object sender, EventArgs e)
    {
        tooltip1.SetToolTip(textBox1, "Lütfen adınızı giriniz");
        tooltip1.SetToolTip(textBox2, "Lütfen sayadınızı giriniz");
        tooltip1.SetToolTip(button1, "Kaydetmek için tıklayınız");
        tooltip1.ToolTipTitle = "Üye Kayıt Formu";
        tooltip1.ToolTipIcon = ToolTipIcon.Info;
    }
}
```

Application Window Preview:

The application window is titled "ÜYE KAYIT ...". It contains two text boxes labeled "AD" and "SOYAD", and a button labeled "KAYDET". A tooltip is displayed over the "AD" text box, showing the text "Üye Kayıt Formu" and "Lütfen adınızı giriniz".

Web Browser Kontrolü: Günlük hayatta kullandığımız tarayıcılarla aynı işi yapar.

- ▶ Document: Mevcut html kodlarını çalıştırır.
- ▶ Url: Sayfa isminin alınıp döndürülmesini sağlar
- ▶ ScrollBarsEnabled: Kaydırma çubuklarının aktifliğini ayarlar
- ▶ GoHome: Ana sayfaya geri gidilir.
- ▶ Stop: Navigate işlemini sonlandırır.



Form1

<<

>>

www.google.com

Git

Görseller Play Haberler Gmail Drive Takvim Çeviri Daha fazlası ▾

Oturum açın



Web'de daha hızlı ge

Google Chrome

Google
Türkiye

Google'da Ara

Kendimi Şanslı Hissediyorum

```
private void btnGeri_Click(object sender, EventArgs e)
{
    webBrowser1.GoBack();
    //Bir önceki sayfaya gidilmesini sağlar
}

private void btnIleri_Click(object sender, EventArgs e)
{
    webBrowser1.GoForward();
    //Bir sonraki sayfaya gidilmesini sağlar
}

private void btnGit_Click(object sender, EventArgs e)
{
    webBrowser1.Navigate(textBox1.Text);
    //Belirtilen url'ye ulaşılmasını ve görüntülenmesini sağlar
}
```

ContextMenuStrip Kontrolü: Kontrollere sağ tuş özelliği eklemek için kullanılır.

- ▶ **Item:** Menü öğelerine erişimi sağlar
- ▶ **Checked:** Menü öğelerinin yanında check işaretinin olup olmasını ayarlar.
- ▶ **DefaultItem:** Varsayılan menü öğesini belirler.
- ▶ **ShortcutKeys:** Kullanılacak kısayol tuşlarını belirler.
- ▶ **ShowShortcutKeys:** Kısayol tuşlarının kullanılıp kullanılmayacağını belirler.



MenuStrip Kontrolü: Menü dizaynı yapmak için kullanılır. Bu kontrolün özellikleri ContextMenuStrip ile aynıdır.

The image shows a Visual Studio IDE window with a C# code file named `Form1.cs [Design]`. The code defines three event handlers for a `MenuStrip` control, all operating on `richTextBox1`:

```
private void kesToolStripMenuItem_Click(object sender, EventArgs e)
{
    richTextBox1.Cut();
}

private void kopyalaToolStripMenuItem_Click(object sender, EventArgs e)
{
    richTextBox1.Copy();
}

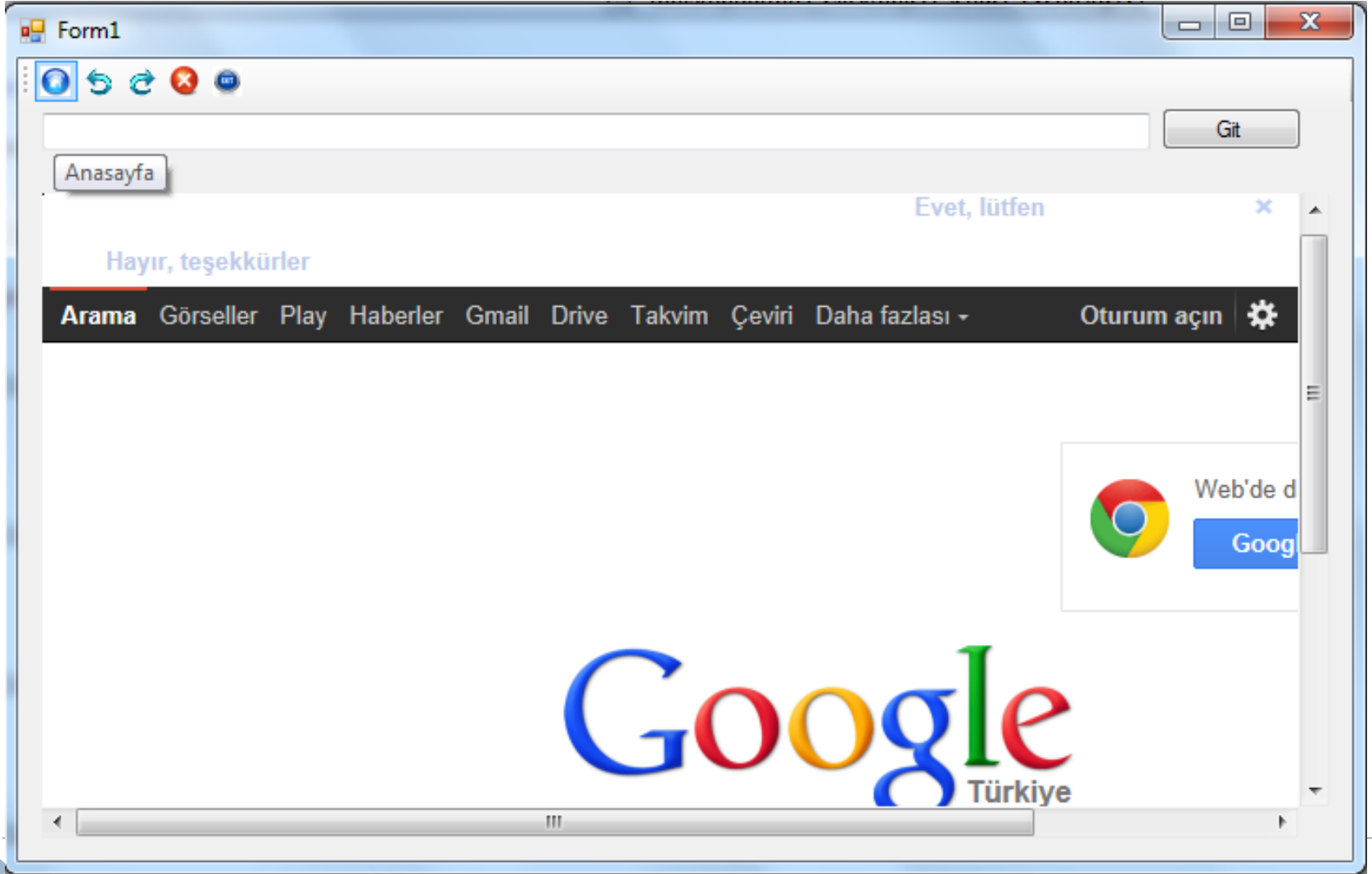
private void yapıştırToolStripMenuItem_Click(object sender, EventArgs e)
{
    richTextBox1.Paste();
}
```

On the right side, there is a preview of the `Form1` window. It features a menu bar with a single menu item labeled `Düzen`. A context menu is displayed over the form, containing the following items:

- Kes Ctrl+X
- Kopyala
- Yapıştır

A dashed blue line connects the `yapıştırToolStripMenuItem_Click` method in the code to the `Yapıştır` item in the menu preview.

ToolStrip Kontrolü: Form ekranının üst kısmında araç çubuğu oluşturmak için kullanılır.



}

```
private void button1_Click(object sender, EventArgs e)
```

{

```
    webBrowser1.Navigate(textBox1.Text);
```

}

```
private void toolStripButton1_Click(object sender, EventArgs e)
```

{

```
    webBrowser1.Navigate("www.google.com");
```

}

```
private void toolStripButton2_Click(object sender, EventArgs e)
```

{

```
    webBrowser1.GoBack();
```

}

```
private void toolStripButton3_Click(object sender, EventArgs e)
```

{

```
    webBrowser1.GoForward();
```

}

```
private void toolStripButton4_Click(object sender, EventArgs e)
```

{

```
    webBrowser1.Stop();
```

}

```
private void toolStripButton5_Click(object sender, EventArgs e)
```

{

```
    Application.Exit();
```

}

Timer Kontrolü:

- ▶ İstedğimiz komutların belirlediğimiz zaman aralığında tekrarlamasını istiyorsak timer kullanabiliriz.
- ▶ **Interval Özelliği:** Timer'ın çalışacağı zaman bölümüdür. Zaman mili saniye cinsindedir. Mesela her 1 saniyede kodları çalıştırmak istersek interval 1000, her 2 saniyede kodları çalıştırmak için interval'i 2000 yapmamız gerekir.
- ▶ **Enabled Özelliği:** Değeri true ise timer'ın çalışmasını, false ise durdurulmasını sağlar.
- ▶ **Tick Olayı :** Bu olaya yazılan kodlar interval ile belirlenen zamanda tekrar edilir.

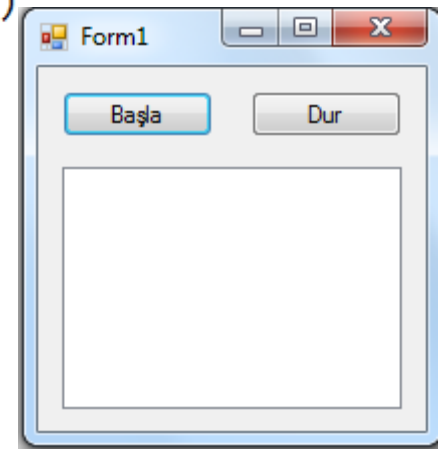


```
private void Form1_Load(object sender, EventArgs e)
{
    timer1.Enabled = false;
    timer1.Interval = 1000;
}

private void btnBasla_Click(object sender, EventArgs e)
{
    timer1.Enabled = true;
}

private void btnDur_Click(object sender, EventArgs e)
{
    timer1.Enabled = false;
}

private void timer1_Tick(object sender, EventArgs e)
{
    listBox1.Items.Add("Görsel Programlama");
}
}
```



ProgressBar Kontrolü:

- ▶ **Maximum Özelliği:** Progressbar'ın maksimum alacağı değeri belirler. Varsayılan değeri 100'dür.
- ▶ **Minimum Özelliği:** Progressbar'ın minimum alacağı değeri belirler. Varsayılan değeri sıfırdır.
- ▶ **Value Olayı :** Progressbar'ın gözüken değeridir.
- ▶ **Step:** Artış değeridir.



Form1.cs* × Form1.cs [Design]*

progressbarKntrl2.Form1

[-]

[-]

[-]

```
public Form1()
{
    InitializeComponent();
}

int i=0;

private void Form1_Load(object sender, EventArgs e)
{
    timer1.Enabled = true;
}

private void timer1_Tick(object sender, EventArgs e)
{
    if (i < 100)
    {
        i++;
        progressBar1.Value = i;
        label1.Text = "% " + i;
    }
}
```

150 %

Form1

label1