

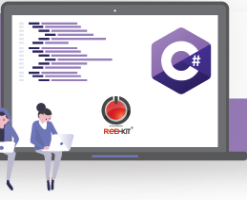
FORM EKRANI-2

FORM EKRANI

- ✓ *Solution Explorer Sekmesi*
- ✓ *Properties Sekmesi*
- ✓ *Toolbox Kontrolleri*

KAZANIMLAR

- ✓ *Visual Studio menülerini sıralar.*
- ✓ *Visual Studio menülerinin görevlerini açıklar.*
- ✓ *Windows form uygulamalarını kavrar.*
- ✓ *Program üzerinde Visual Studio menülerini kullanır.*



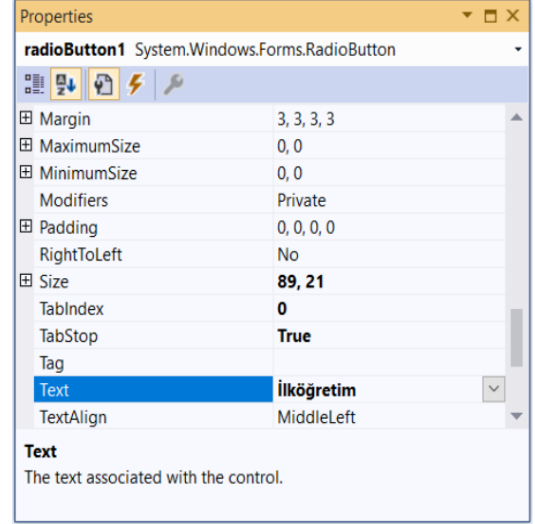
RadioButton

RadioButton kontrolü CheckBox'tan farklı olarak birkaç seçenekten sadece birini seçme imkanı veren bir onay kontrolüdür.

- ✓ Bu kontrolün tek başına kullanılması anlamsızdır. Bir kaç seçenekten birini seçme imkanı veren bir kontrol olduğu için en az iki tane birlikte verilmelidir.
- ✓ Formda düğmelerden biri seçildiğinde seçilmiş olan kendiliğinden kalkacaktır.
- ✓ GroupBox kontrolleri sayesinde farklı seçim butonları yapabilirsiniz

Bu kontrolün özellikleri CheckBox'a çok benzemektedir. CheckBox'ta olduğu gibi;

- ✓ Appearance özelliği ile komut düğmesi görünümüne sahip RadioButton düğmeleri oluşturulabilir.
- ✓ Text özelliği ile içindeki metin yazılabilir.
- ✓ TextAlign özelliği ile içindeki metnin yerleşmesi belirlenebilir.
- ✓ Image özellikleri ile içinde resim gösterilebilir.
- ✓ CheckAlign özelliği ile seçenek düğmesi sola, sağa alınabilir.



Örnek:

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

```
{  
    string adi, soyadi, cinsiyet = "";  
    adi = textBox1.Text;  
    soyadi = textBox2.Text;  
    if (radioButton1.Checked)  
    {  
        cinsiyet = "Erkek";  
    }  
    if (radioButton2.Checked)  
    {  
        cinsiyet = "Kadın";  
    }  
    MessageBox.Show("Adı Soyadı:" + adi + " " +  
        + soyadi + "\nCinsiyeti:" + cinsiyet);  
}
```





ComboBox

Açılır liste üzerinden sunulan seçenekler arasından seçim yapılmasına olanak sağlayan araçtır.

- ✓ Listelenen öğeler açılan bir kutuda görüntülenir ve listeden en fazla bir tane öğe seçilebilir.
- ✓ Combobox'ın Items özelliği kullanılarak açılır listeye eleman eklenebilir.

Sayfa içerisinde kodlar ile ComboBox a eleman eklenebilir.

```
private void Form1_Load(object sender, EventArgs e)
{
    string[] meslekler = { "İşçi", "Memur", "Mühendis", "Eğitimci", "Programcı", "Tekniker",
    "Veteriner" };
    ComboBox.Items.AddRange(meslekler);
}
```

Örnek: Örneğimizde combobox kontrolünde seçilen elemana göre listbox ' a eleman ekleyeceğiz. combobox1 içinde seçilen ilin ilçelerini listbox içinde göstereceğiz.

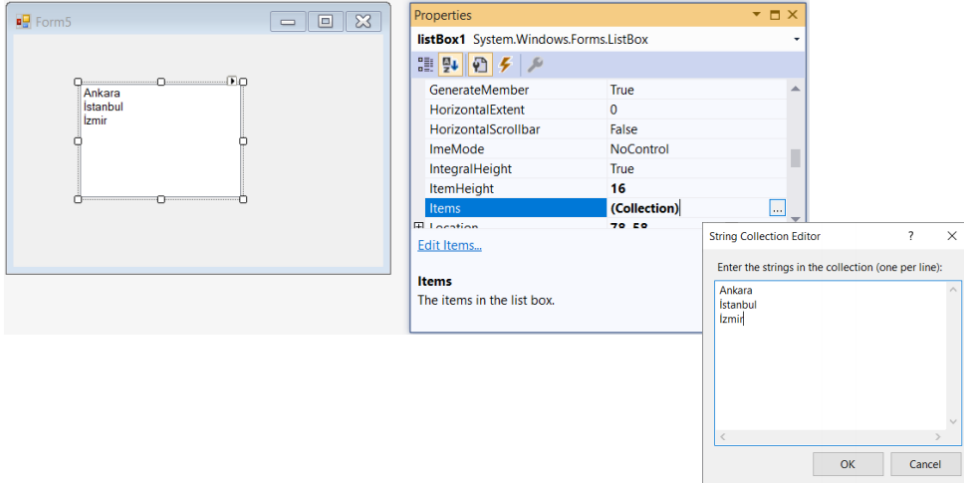
```
private void Form1_Load(object sender, EventArgs e)
{
    comboBox1.Items.Add("İstanbul");
    comboBox1.Items.Add("Ankara");
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    {
        listBox1.Items.Clear();
        if (comboBox1.SelectedIndex == 0)
        {
            listBox1.Items.Add("Avcılar");
            listBox1.Items.Add("Bahçelievler");
            listBox1.Items.Add("Beşiktaş");
            listBox1.Items.Add("Kadıköy");
        }
        else if (comboBox1.SelectedIndex == 1)
        {
            listBox1.Items.Add("Etimesgut");
            listBox1.Items.Add("Yenimahalle");
            listBox1.Items.Add("Çankaya");
            listBox1.Items.Add("Keçiören");
        }
    }
}
```



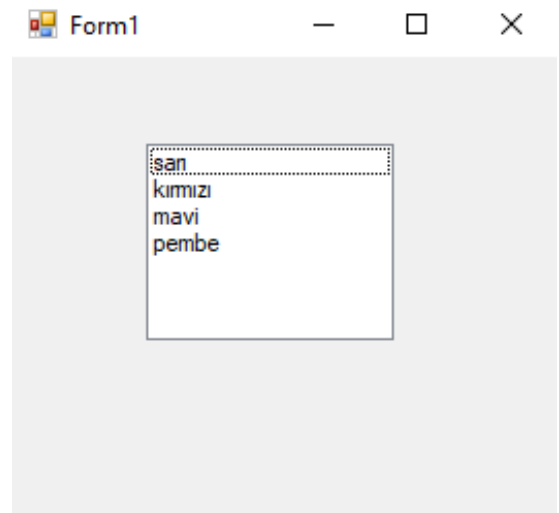
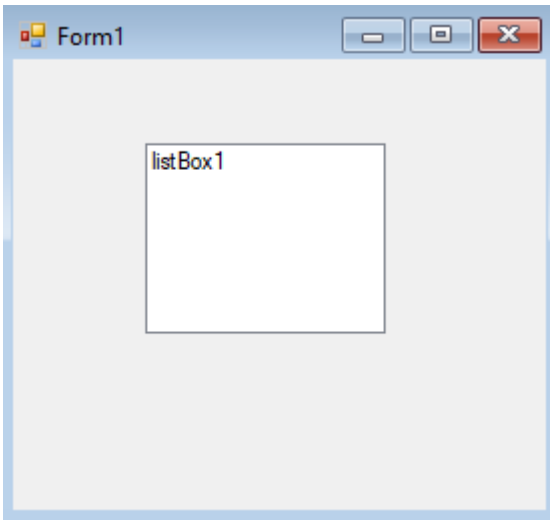
ListBox

Kullanıcıya sunulan seçeneklerin bir liste halinde görünmesini sağlar. Liste kutusundan istenen sayıda öğe seçilebilir.



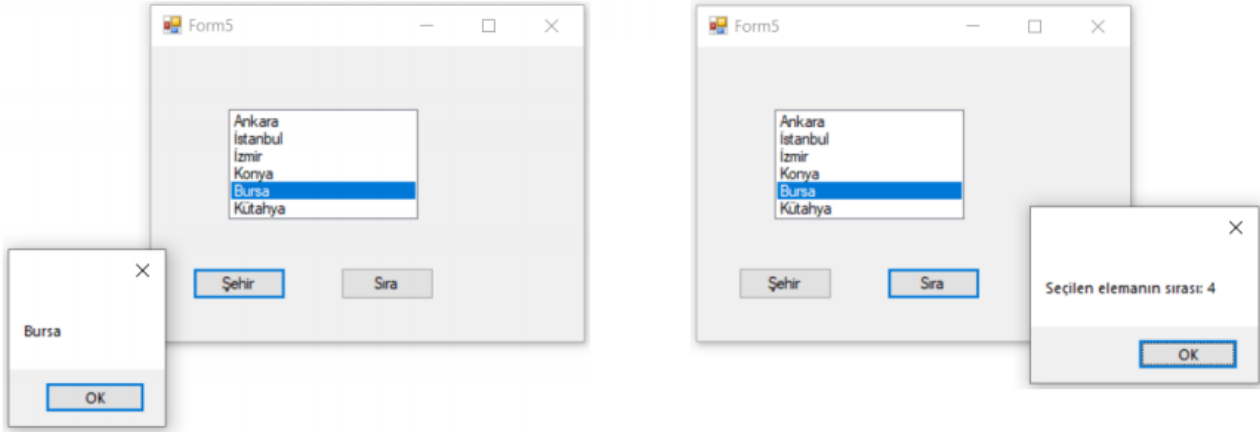
Örnek: Kod ortamında ListBox'a eleman ekleme

```
private void Form1_Load(object sender, EventArgs e)
{
    listBox1.Items.Add("sarı");
    listBox1.Items.Add("kırmızı");
    listBox1.Items.Add("mavi");
    listBox1.Items.Add("pembe");
}
```



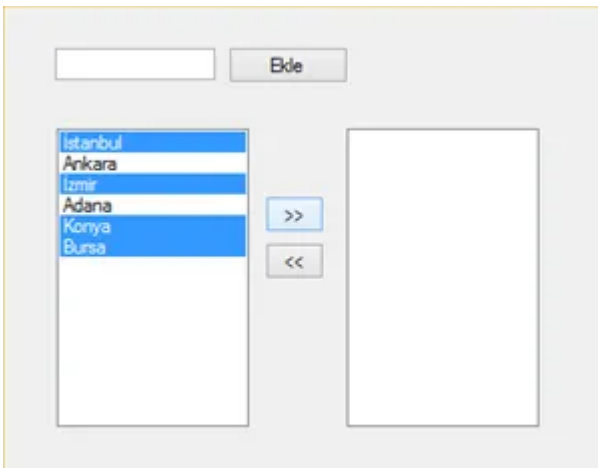


Örnek: Listboxdan seçilen şehiri birinci butona basıldığında şehir ismini ikinci butona basıldığında şehrin sırasını veren uygulamamız



```
private void button1_Click_1(object sender, EventArgs e)
{
    MessageBox.Show(listBox1.Text);
}
private void button2_Click(object sender, EventArgs e)
{
    MessageBox.Show("Seçilen elemanın sırası: " + listBox1.SelectedIndex.ToString());
}
```

Örnek: Ekle butonuna basıldığında textBox1 de bulunan metnin listBox1'e aktarılmasını sağlamak için Ekle butonuna çift tıklayarak; daha sonrada listboxlar arası geçiş yapalım.



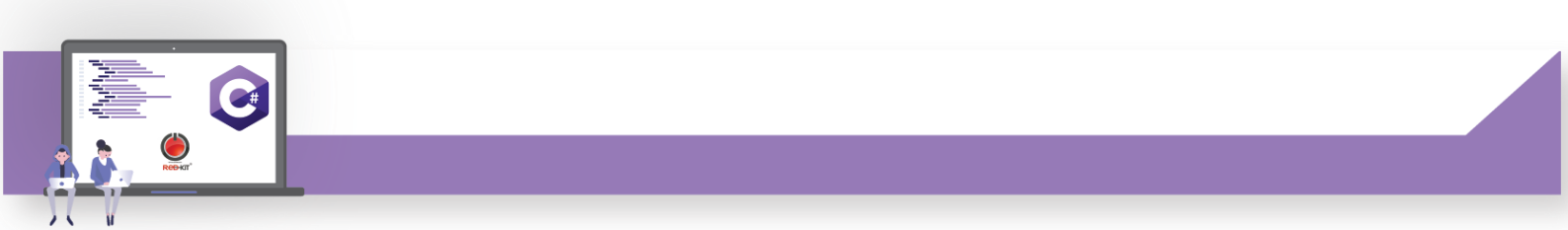


```
private void Form1_Load(object sender, EventArgs e)
{
    listBox1.SelectionMode = SelectionMode.MultiSimple; // Çoklu seçim
    listBox2.SelectionMode = SelectionMode.MultiSimple;
}
private void button1_Click(object sender, EventArgs e)
{
    Random rndm = new Random();

}
private void button1_Click_1(object sender, EventArgs e)
{
    listBox1.Items.Add(textBox1.Text);
}
private void button2_Click(object sender, EventArgs e)
{
    for (int i = listBox1.SelectedIndices.Count - 1; i >= 0; i--)
    {
        //Seçili elemanı ikinci listeye ekle
        listBox2.Items.Add(listBox1.SelectedItem);
        //seçili elemanı birinci listeden çıkar
        listBox1.Items.RemoveAt(listBox1.SelectedIndices[i]);
    }
}
private void button3_Click(object sender, EventArgs e)
{
    for (int j = listBox2.SelectedIndices.Count - 1; j >= 0; j--)
    {
        //Seçili elemanı birinci listeye ekle
        listBox1.Items.Add(listBox2.SelectedItem);
        //seçili elemanı ikinci listeden çıkar
        listBox2.Items.RemoveAt(listBox2.SelectedIndices[j]);
    }
}
```

Timer

C# Windows Form uygulamalarında Timer kontrolü kullanımı istenen komutların belirlenen zaman aralığında tekrarlanmasını sağlamak için kullanılmaktadır. Timer kontrolüne ait özellikleri ve Timer kontrolü kullanılarak yapılmış örnekleri yazının devamında bulabilirsiniz.



Özellikleri

Interval Özelliği: Bu özellik kodlarımızın ne kadar sürede tekrarlanacağını belirlediğimiz özelliktir. Milisaniye cinsinden değer verilir. Örnek olarak her 1 saniyede çalışmasını istediğimiz kodlar için Interval özelliği 1000, 1,5 saniye olarak ayarlamak için 1500 olarak ayarlanmalıdır.

Enabled Özelliği: Değeri true ise timer'ın çalışmasını, false ise durdurulmasını sağlar. Yani enabled değeri true ise timer'ın Tick olayına yazılan kodlar çalışır, false ise çalışmaz.

Start: Timer nesnesini başlatır.

Stop: Timer nesnesini durdurur

Olaylar

Tick Olayı : Tekrarlanmasını istediğimiz kodları Timer_Tick olayına yazarız.

ProgressBar

ProgressBar bir işlem sırasında işlemin ilerleme durumunu göstermek amacıyla kullanılır.

Maximum: ProgressBar nesnesinin alabileceği en fazla değeri belirtir.

Minimum: ProgressBar nesnesinin alabileceği en fazla değeri belirtir.

Step: ProgressBar'ın artış miktarını verir.

Örnek: Progressbar dolduğunda yeni bir sayfa açan uygulama yapalım..

```
private void Form1_Load(object sender, EventArgs e)
{
    timer1.Interval = 1000;
    progressBar1.Maximum = 2;
}

private void timer1_Tick(object sender, EventArgs e)
{
    if (progressBar1.Value != progressBar1.Maximum)
        progressBar1.Value++;
    else
    {
        timer1.Stop();
        Form2 frm2 = new Form2();
        frm2.Show();
        this.Hide();
    }
}
```

