

ENM5223 ENDÜSTRİ MÜHENDİSLİĞİNDE İLERİ PROGRAMLAMA UYGULAMALARI – C# FORM ARAÇLARI

Doç. Dr. İbrahim KÜÇÜKKOÇ

Web: <http://ikucukkoc.baun.edu.tr>

Email: ikucukkoc@balikesir.edu.tr

2

Yararlanılan Kaynaklar:

C# Eğitim Kitabı 2. Baskı, Murat Yücedağ, Dikeyksen Yayınları, 2019.

C# Ders Notları, YMT218 Nesne Tabanlı Programlama, Fırat Üniversitesi (Erişim Tarihi 29 Ekim 2019).

C# Programlama Ders Notları, Dr. Serkan Dişlitaş, Hitit Üniversitesi (Erişim Tarihi 29 Ekim 2019).

Güncelleme: 6 Ekim 2021

Visual Studio IDE kullanarak, C# Windows Forms Uygulaması geliřtirmek iin ařağıdaki adımlar takip edilmelidir.

Visual Studio 2019

Open recent

As you use Visual Studio, any projects, folders, or files that you open will show up here for quick access.

You can pin anything that you open frequently so that it's always at the top of the list.

Get started



Clone or check out code

Get code from an online repository like GitHub or Azure DevOps



Open a project or solution

Open a local Visual Studio project or .sln file



Open a local folder

Navigate and edit code within any folder



Create a new project

Choose a project template with code scaffolding to get started

[Continue without code](#) →

Varolan projeyi aç

Yeni bir proje oluřtur



Create a new project

Recent project templates

A list of your recently accessed templates will be displayed here.

All Languages

All Platforms

All Project Types

 Console App (.NET Framework)
A project for creating a command-line application
C# Windows Console

 Windows Forms App (.NET Framework)
A project for creating an application with a Windows Forms (WinForms) user interface
C# Windows Desktop

 Class Library (.NET Framework)
A project for creating a C# class library (.dll)
C# Windows Library

 Class Library (.NET Core)
A project for creating a class library that targets .NET Core.
C# Windows Linux macOS Library

 SQL Server Database Project
A project for creating a SQL Server database.
Query Language Windows Web

 Unit Test Project (.NET Framework)
A project that contains MSTest unit tests.

Back

Next

C# için “.NET Framework”
seçildiğinden emin
olunmalıdır.



Configure your new project

Windows Forms App (.NET Framework)

C#

Windows

Desktop

Project name

Proje1

Location

C:\Users\ibrah\source\repos

Solution name 

Proje1

☐ Place solution and project in the same directory

Framework

.NET Framework 4.7.2

Proje ismi verildikten sonra CREATE butonuna tıklanabilir.

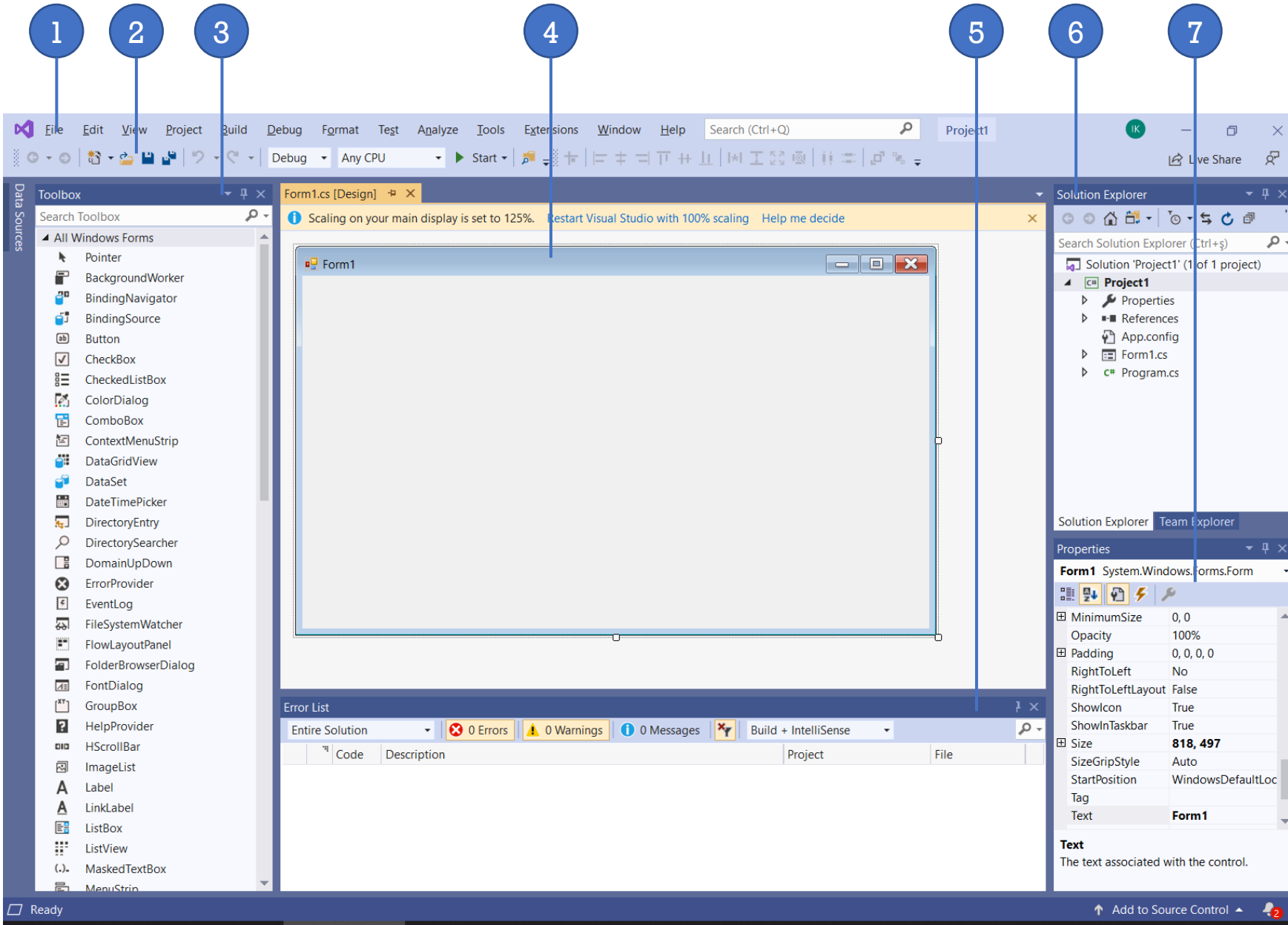
**NOT: Aynı işlemler
FILE -> NEW -> PROJECT
yolunu izleyerek de
gerçekleştirilebilir.**

Yapılan bu işlemler sonucunda Visual Studio IDE, C# Windows Forms Uygulaması geliştirmek için hazır hale getirilmiş olur.

Back

Create



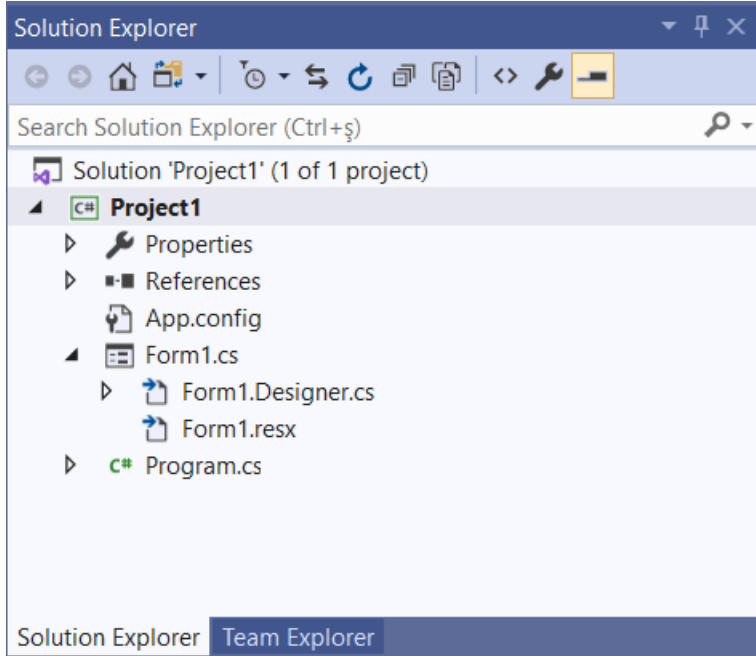


1. Menü Çubuğu (Menu Bar)
2. Standart Araç Çubuğu (Standard Toolbar)
3. Araç Kutusu (ToolBox)
4. Form Tasarımcısı (Forms Designer)
5. Hata Listesi (Output Window-Error List),
6. Çözüm Gezgini (Solution Explorer)
7. Özellikler ve Olaylar (Properties / Events)



Çözüm Gezgini (Solution Explorer)

- Çözüm Gezgini (Solution Explorer), çözüm ve çözüme bağlı projelerdeki kontrol, sınıf, metot vb. bileşenleri görüntülemek amacıyla kullanılan Visual Studio aracıdır.
- Solution, projeleri gruplandırarak bir çatı altında toplamak amacıyla kullanılır. Bir Solution içerisinde, bir veya daha fazla proje olabilir. Solution ile ilgili kodlar *.sln uzantılı dosyalarda tutulmaktadır.

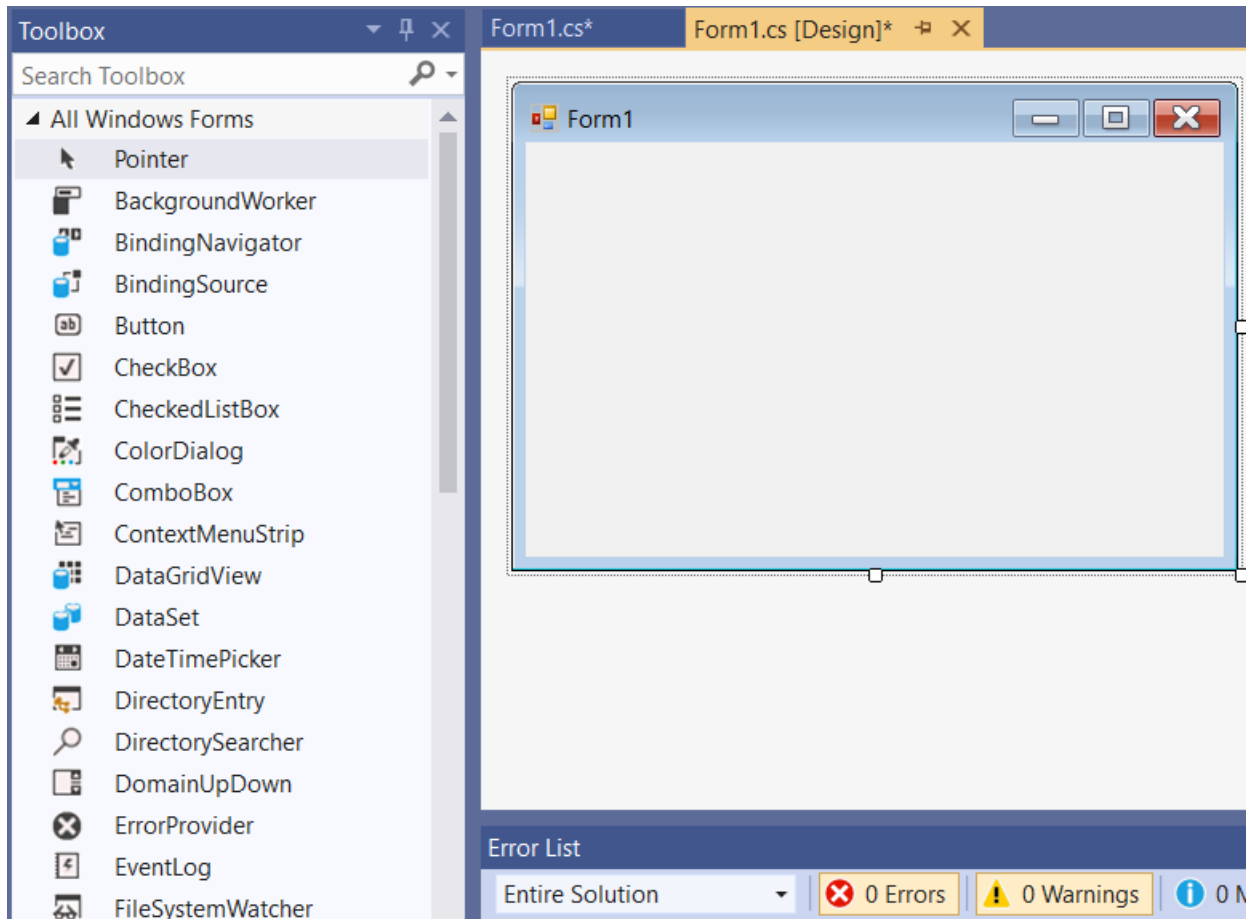


Dosya Adı	Açıklama
App.config	Uygulama ile ilgili genel tanımlama ve bilgileri içerir.
Form1.cs	Windows Form Uygulaması Kod sayfasıdır. (Her form için ayrı bir kod sayfası mevcut olabilir.)
Form1.Designer.resx	Form tasarımında kullanılan nesnelerin görsel özellikleri ile ilgili kodların yer aldığı dosyadır. C# tarafından otomatik olarak oluşturulur.
Form1.resx	Form tasarımında kullanılan yazı, resim vb. medya öğelerini içeren “resource-kaynak” dosyadır.
Program.cs	Proje ortam ayarlarını içeren dosyadır. (Örneğin birden çok forma sahip Projelerde, başlangıç formu bu dosya içerisinde belirlenir.)



Toolbox

- Windows tabanlı uygulamalar geliştirirken sıkça kullanacağımız bir grup kontrol vardır.
- Form kontrolü hariç diğer bütün kontroller **Toolbox** panelinden seçilir.
- Bu kontroller sürüklenip Form üzerine istenilen pozisyona bırakılır.

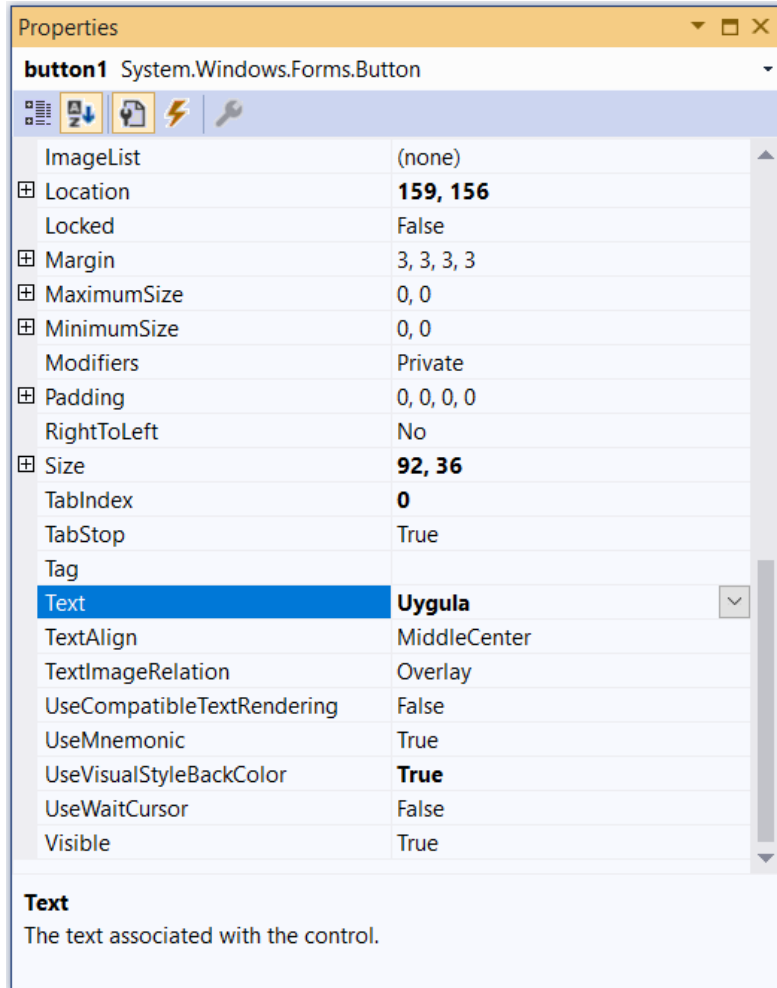


Toolbox kapalı ise **Ctrl + Alt + X** tuşlarına birlikte basarak aktif hale getirilebilir.
Veya

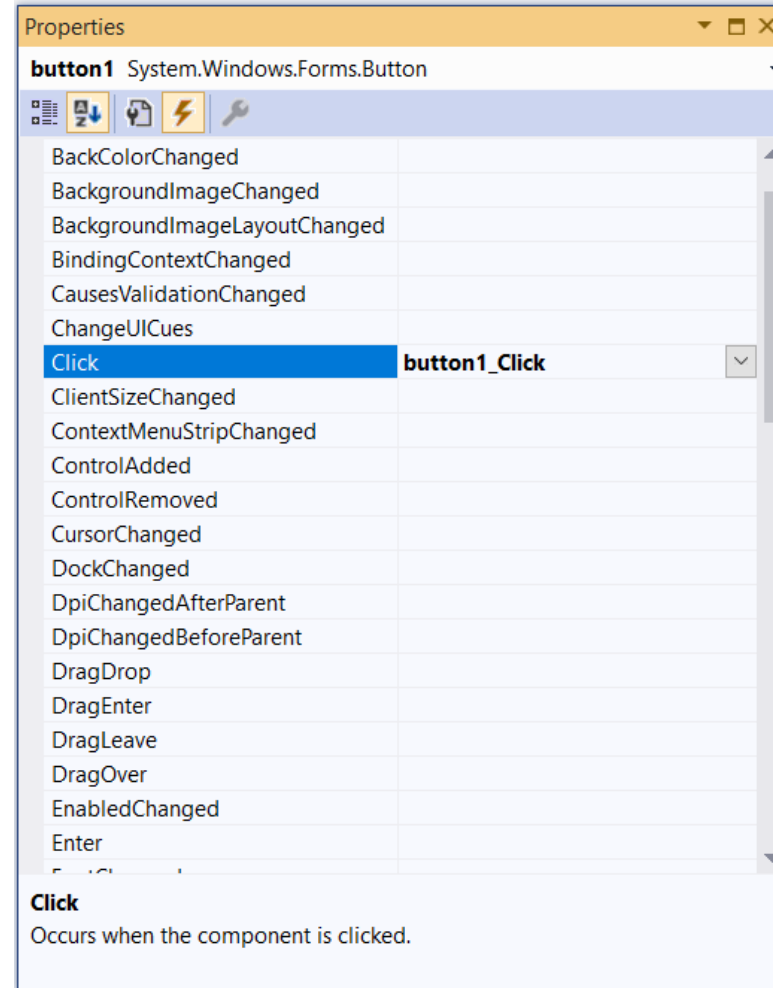


Özellikler Penceresi (Properties & Events)

- Seçili olan nesnenin özelliklerini ve olaylarını görüntüleyip düzenlemek amacıyla kullanılan Visual Studio aracıdır.



Properties



Events



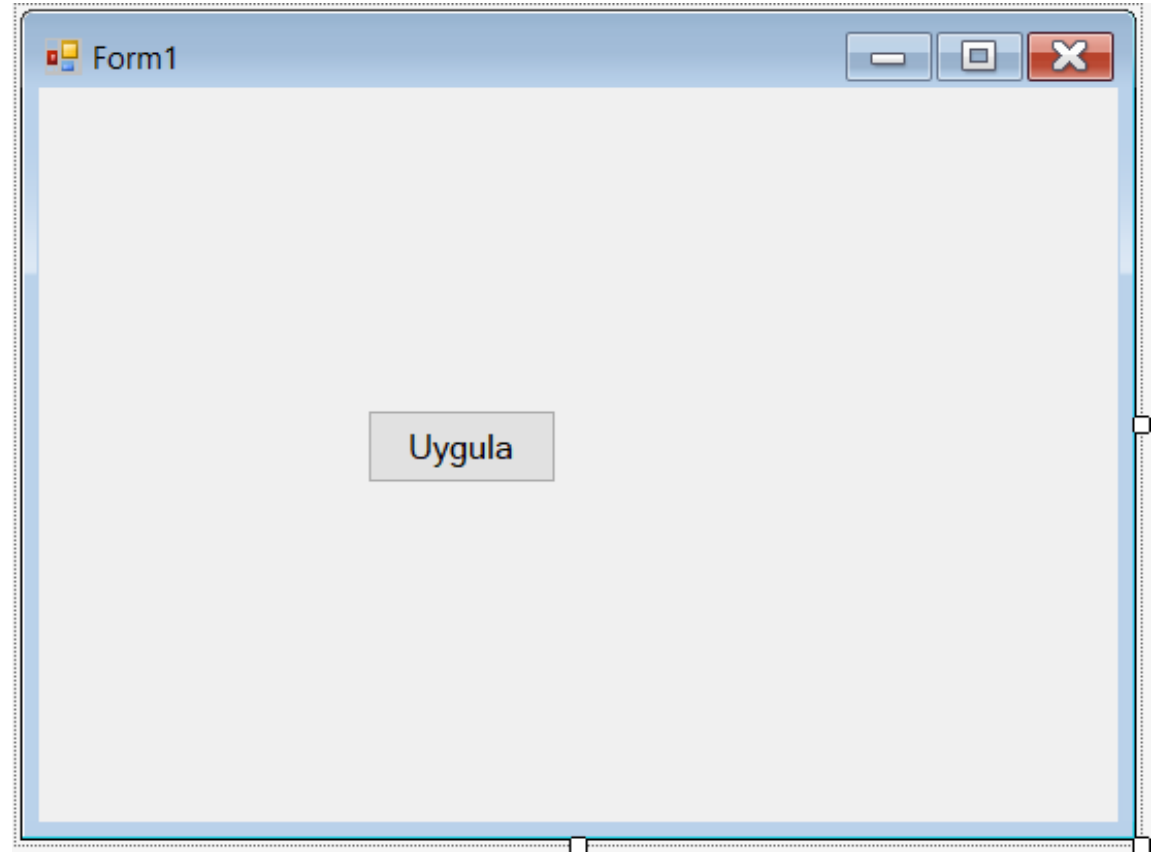
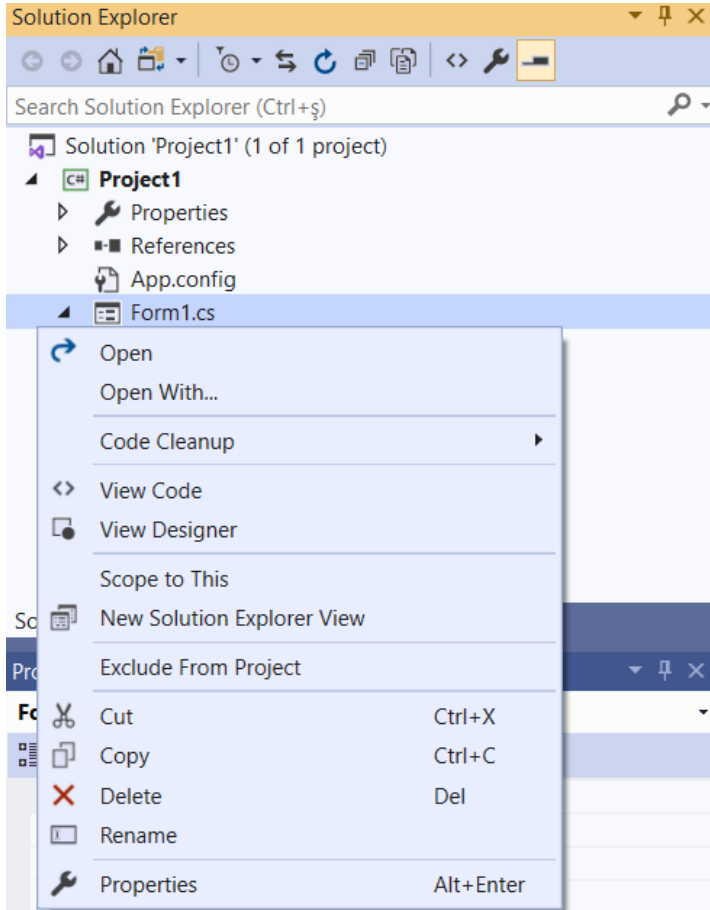
Özellikler Penceresi (Properties & Events)

Simge	Simge Adı	Açıklama
	Özellikler (Properties)	Seçili nesneye ait özellikleri listeler.
	Olaylar (Events)	Seçili nesneye ait olayları listeler.
	Kategorik Sıralama	Özellik ve olayların kategorilere ayrılarak listelenmesini sağlar
	Alfabetik Sıralama	Özellik ve olayların alfabetik olarak sıralı listelenmesini sağlar



Kodu Görüntüleme

- Çözüm Gezgini penceresinde **Form1** üzerinde sağ klik yapılarak elde edilen menüden proje formu ve kaynak kodları görüntülenebilmektedir.
- Diğer bir yol ise, Designer alanında Form1'in üzerine çift tıklamaktır.



Kodu Görüntüleme

```
Form1.cs*  Form1.cs [Design]*
C# Project1 Project1.Form1 Form1_Load(object sender, EventArgs e)

7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 namespace Project1
12 {
13     public partial class Form1 : Form
14     {
15         public Form1()
16         {
17             InitializeComponent();
18         }
19
20         private void button1_Click(object sender, EventArgs e)
21         {
22
23         }
24
25         private void Form1_Load(object sender, EventArgs e)
26         {
27
28         }
29     }
30 }
31
```



Start (Başlat)

- C# program geliştirme işlemlerinde yapılan değişiklikler için **Kaydetme** işlemi yapılmalıdır.

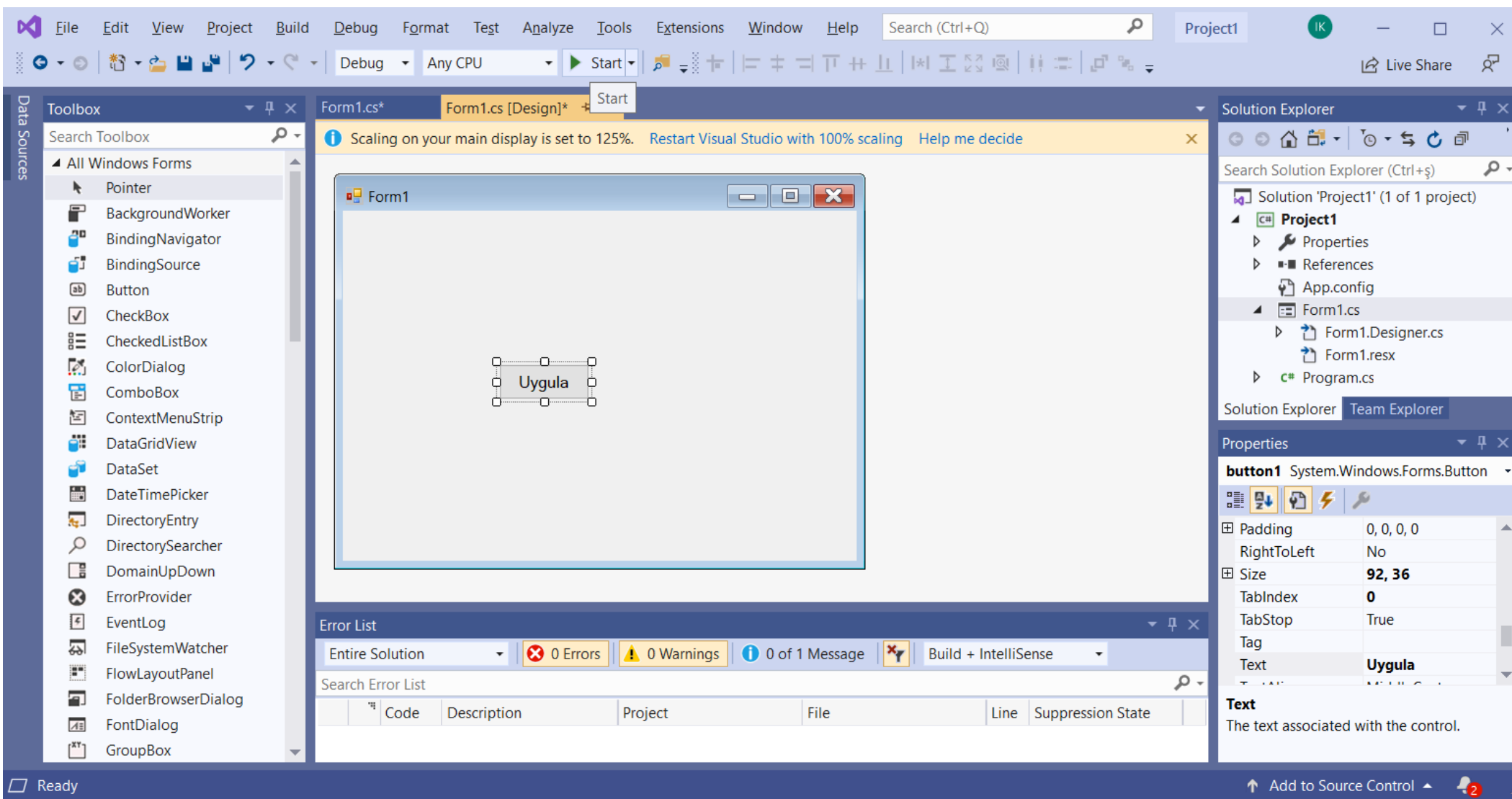
Program Kaydetme Seçenekleri		
Simge	Kısayol Adı	Açıklama
	Ctrl + S	Aktif Form dosyaları kaydedilir.
	Ctrl + Shift + S	Tüm Dosyalar Kaydedilir.

PROGRAMIN DERLENMESİ VE BAŞLATILMASI:

C# dilinde yazılmış bir Windows Forms Uygulamasını çalıştırmak için aşağıdaki yollardan birisi izlenebilir:

- Araç çubuğundan “**Başlat (START)**” butonuna tıklamak
- Menüden “**Hata Ayıkla (DEBUG) ->Hata Ayıklamayı Başlat (START DEBUGGING)**” a tıklamak
- **F5** fonksiyon tuşunu kullanmak





Designer ekranından butonun üzerine çift tıklanarak Click olayına komut yazılabilir. Oluşturulan butonun **Click** olayına yazılan komut aşağıdaki gibidir.



Visual Studio interface showing a C# project named "Project1". The main editor displays the code for "Form1.cs" in Design view. The code defines a partial class Form1 with a constructor, an InitializeComponent method, and two event handlers: button1_Click and Form1_Load. The button1_Click method shows a message box with the text "Merhaba Dünya".

```
13 public partial class Form1 : Form
14 {
15     public Form1()
16     {
17         InitializeComponent();
18     }
19
20     private void button1_Click(object sender, EventArgs e)
21     {
22         MessageBox.Show("Merhaba Dünya");
23     }
24
25     private void Form1_Load(object sender, EventArgs e)
26     {
27     }
28 }
29
30
31
```

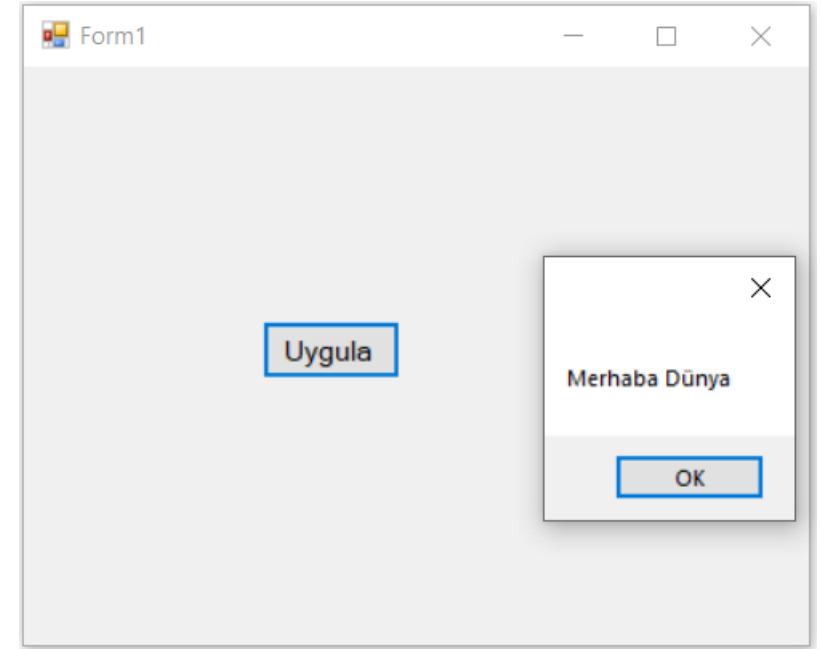
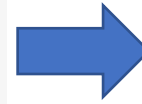
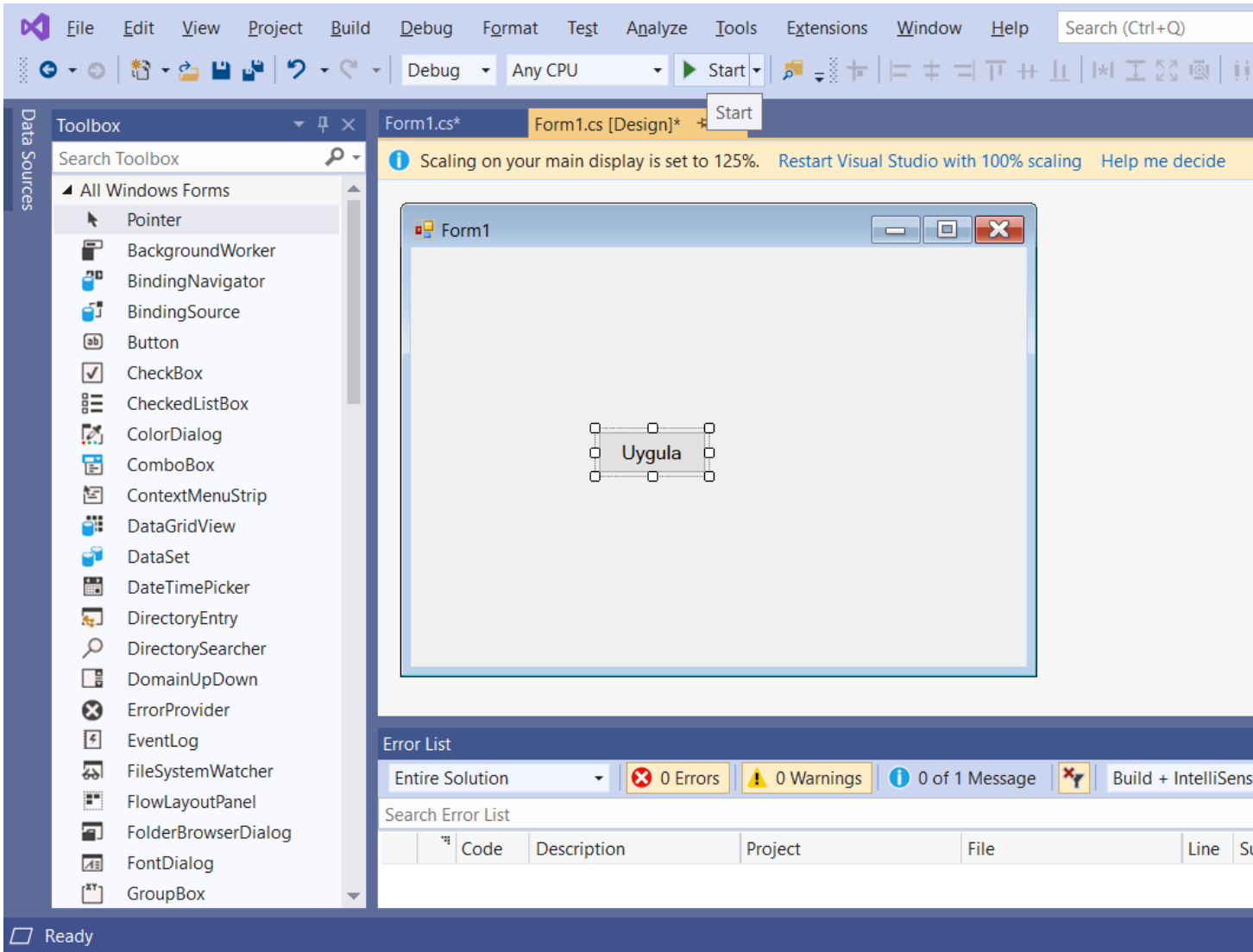
The Solution Explorer on the right shows the project structure:

- Solution 'Project1' (1 of 1 project)
 - Project1
 - Properties
 - References
 - App.config
 - Form1.cs
 - Form1.Designer.cs
 - Form1.resx
 - Program.cs

The Error List at the bottom shows 0 Errors, 0 Warnings, and 0 of 1 Message.

Code	Description	Project	File	Line	Suppression State
------	-------------	---------	------	------	-------------------





Derleme Hatası!

- Yazılan bir programın built edilmesi durumunda eğer hatalar varsa aşağıdaki gibi program çalıştırılmaz ve hata listesi verilir. Hatalar giderilmedikçe program çalıştırılmaz.

The screenshot shows the Visual Studio IDE with a C# file named Form1.cs. The code is as follows:

```
12 {  
13     public partial class Form1 : Form  
14     {  
15         public Form1()  
16         {  
17             InitializeComponent();  
18         }  
19  
20         private void button1_Click(object sender, EventArgs e)  
21         {  
22             MessageBox.Show("Merhaba Dünya")---  
23         }  
24  
25         private void Form1_Load(object sender, EventArgs e)  
26         {  
27  
28         }  
29     }  
}
```

The code has three errors highlighted in red squiggly lines under the closing brace of the button1_Click method on line 22. A dialog box from Microsoft Visual Studio is open, asking: "There were build errors. Would you like to continue and run the last successful build?" with "Yes" and "No" buttons. Below the dialog, the Error List window is visible, showing the following errors:

Code	Description	Project	File	Line	Suppression State
CS1525	Invalid expression term '}'	Project1	Form1.cs	22	Active
CS1002	; expected	Project1	Form1.cs	22	Active
CS1059	The operand of an increment or decrement operator must be a variable, property or indexer	Project1	Form1.cs	22	Active

Dikkat: Programın başarılı bir şekilde Built edilen en son halini çalıştırmak için 'Evet' denir.



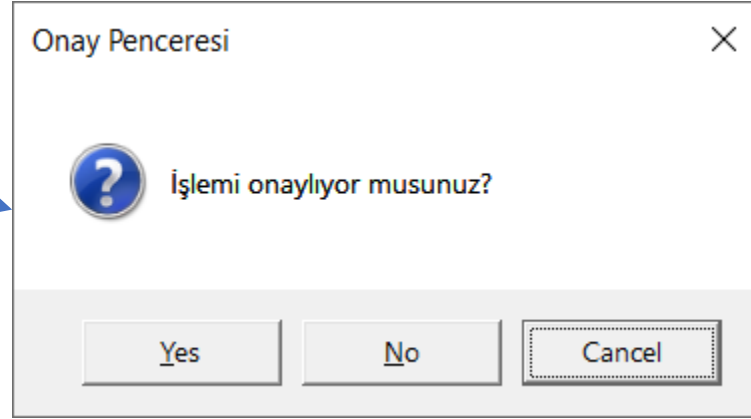
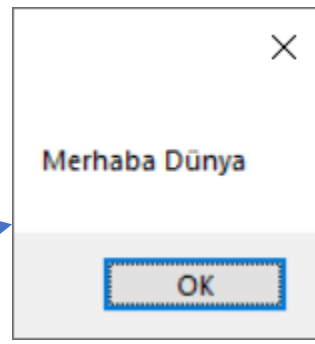
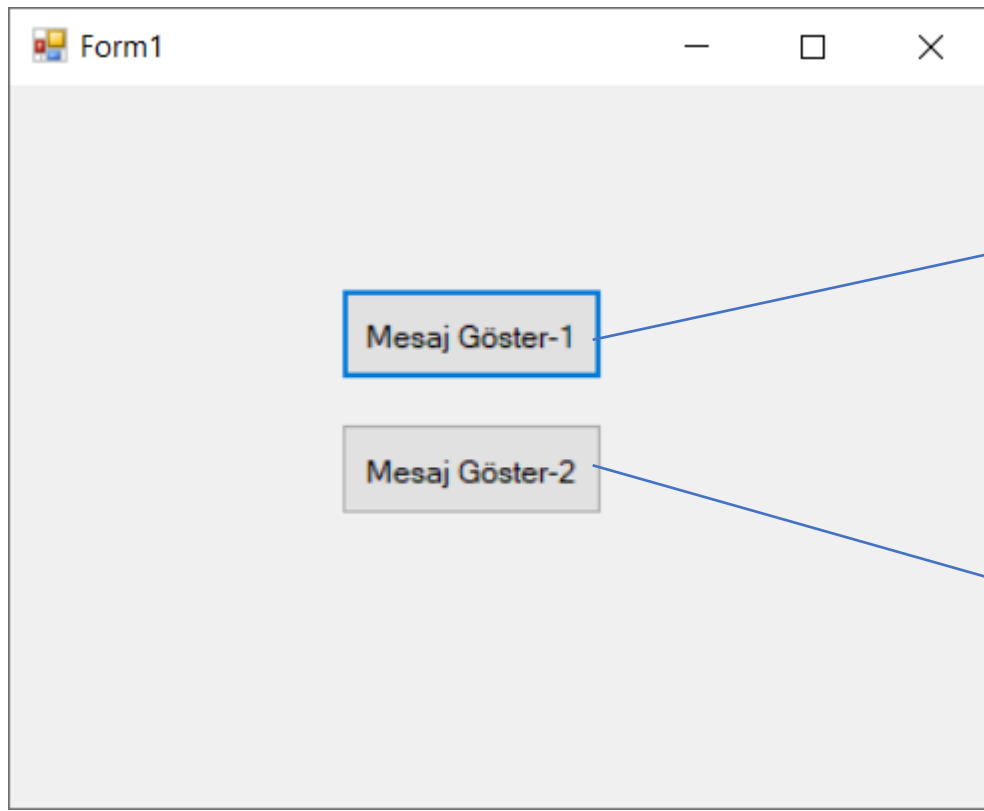
MessageBox

Kullanıcıya diyalog penceresi içerisinde mesaj verme işlemi için **MessageBox** sınıfı **Show** metodu kullanılır.

```
Form1.cs*  Form1.cs [Design]*
C# Project1  Project1.Form1  button2_Click(o

19
20     private void button1_Click(object sender, EventArgs e)
21     {
22         MessageBox.Show("Merhaba Dünya");
23     }
24
25     private void button2_Click(object sender, EventArgs e)
26     {
27         MessageBox.Show("İşlemi onaylıyor musunuz?", //Mesaj içeriği
28             "Onay Penceresi", //Mesaj Başlığı
29             MessageBoxButtons.YesNoCancel, //Mesaj Butonları
30             MessageBoxIcon.Question, //Mesaj Iconu
31             MessageBoxDefaultButton.Button3, //Default Seçili Buton
32             MessageBoxOptions.DefaultDesktopOnly //PencereYerleşimSeçeneği
33     );
34     }
35 }
36 }
```





1. **MessageBox.Show** metodunun alabileceği parametreler şunlardır:

- Mesaj içeriği
- Mesaj Başlığı,
- Butonlar,
- Icon
- Default Seçili Buton
- Mesaj Penceresi Yerleşim Seçenekleri

2. **MessageBox.Show** metodunun alabileceği bu parametrelerin tanımlama sıralamalarına dikkat edilmelidir.

3. Ayrıca sonraki parametrenin kullanılabilmesi için daha önceki parametrelerin kullanılması gerektiği de unutulmamalıdır.

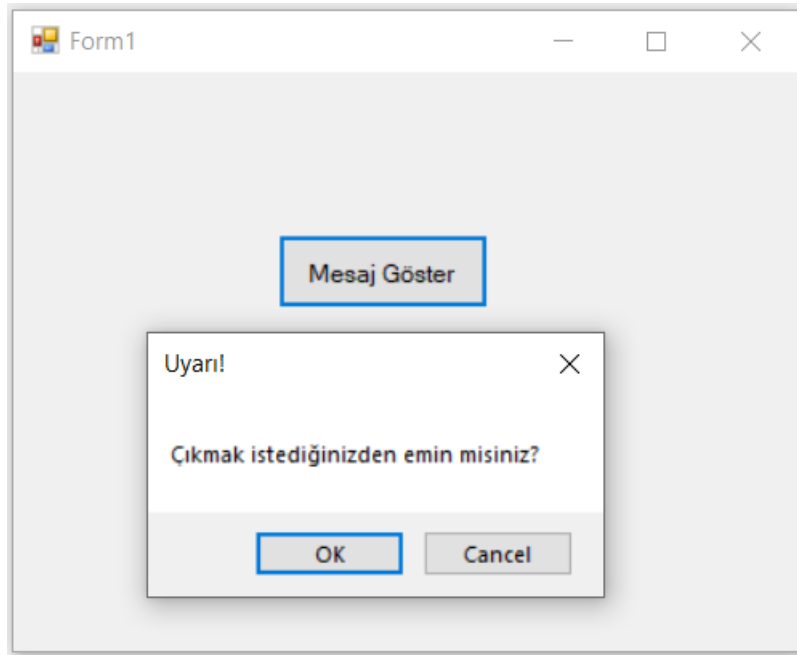


Icon Adı	Icon Görünümü
MessageBoxIcon. Information MessageBoxIcon. Asteriks	
MessageBoxButtons. Question	
MessageBoxButtons. Error MessageBoxButtons. Hand MessageBoxButtons. Stop	
MessageBoxButtons. Exclamation MessageBoxButtons. Warning	
MessageBoxButtons. None veya Bu parametre hiç kullanılmazsa	Icon Görünmez

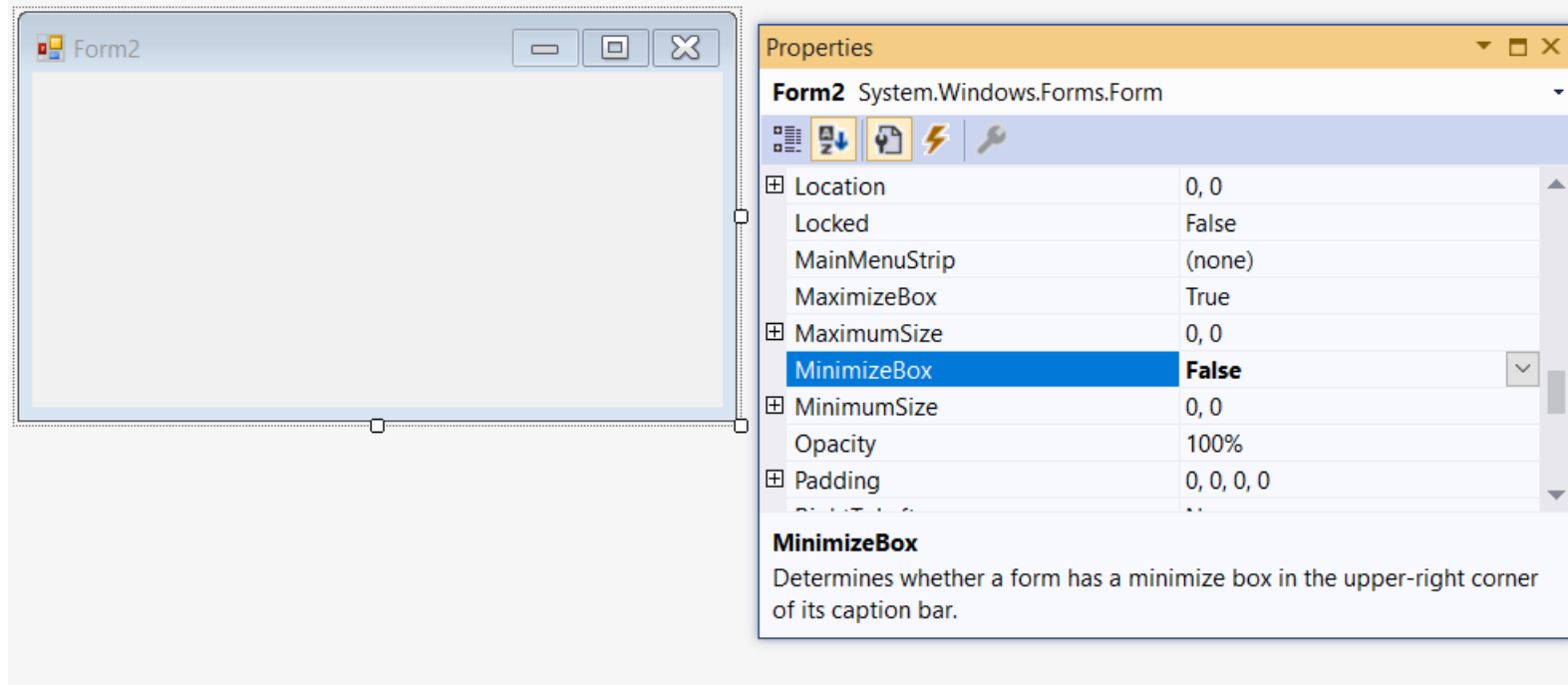


Diyalog Result: Basılan tuşa göre işlem yapma

```
private void button1_Click(object sender, EventArgs e)
{
    DialogResult sonuc;
    sonuc = MessageBox.Show("Çıkmak istediğinizden emin misiniz?", "Uyarı!", MessageBoxButtons.OKCancel);
    if (sonuc == DialogResult.OK)
    {
        Form1.ActiveForm.Close();
    }
}
```



Form Özellikleri – MinimizeBox, MaximizeBox



MinimizeBox Özelliği: Simge Durumuna Küçültme

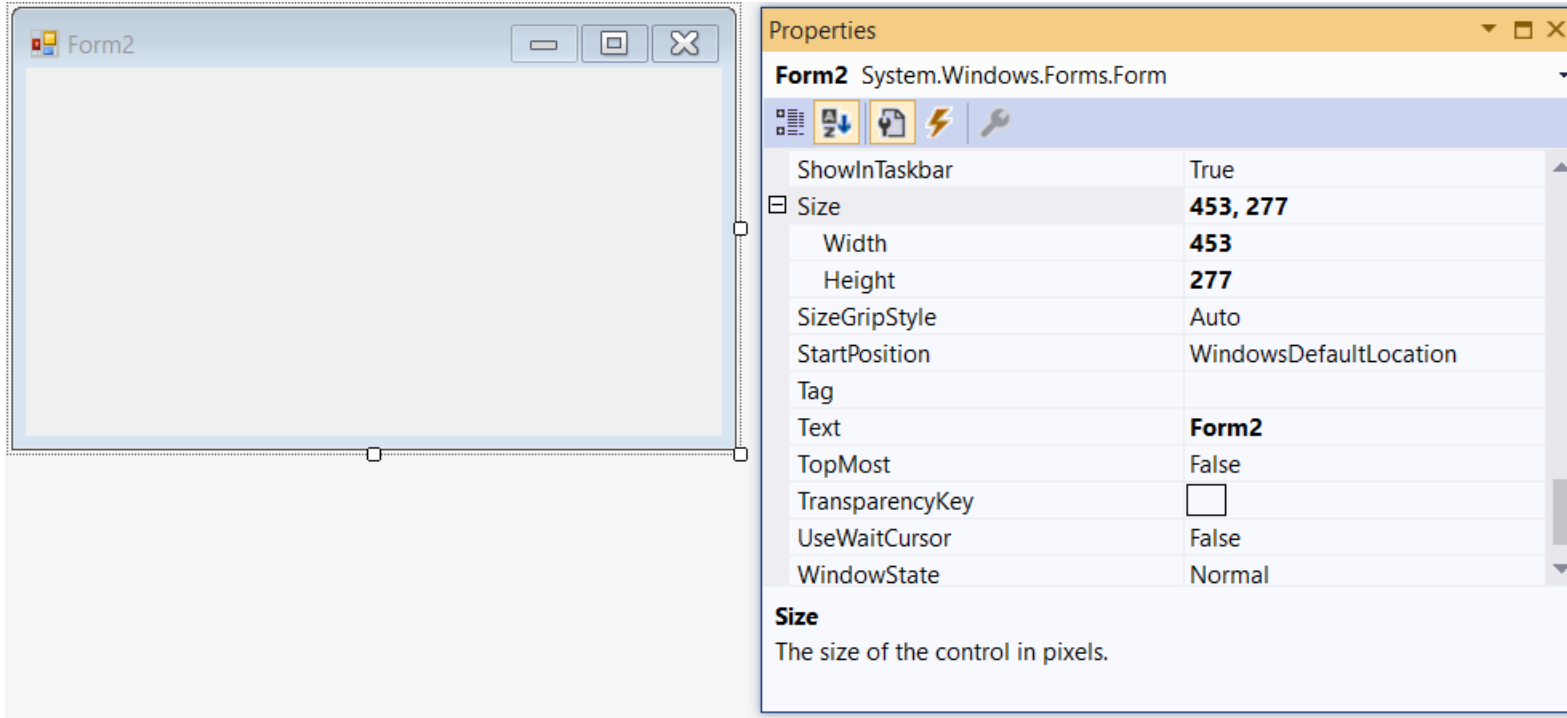
MaximizeBox Özelliği: Ekranı Kaplama

MinimizeBox =false, MaximizeBox=true



Form Özellikleri

- **Formun boyutları:** Width (yatay boyutu) , Height (yüksekliği)



- **Formun Ekrandaki Konumu (Location):**

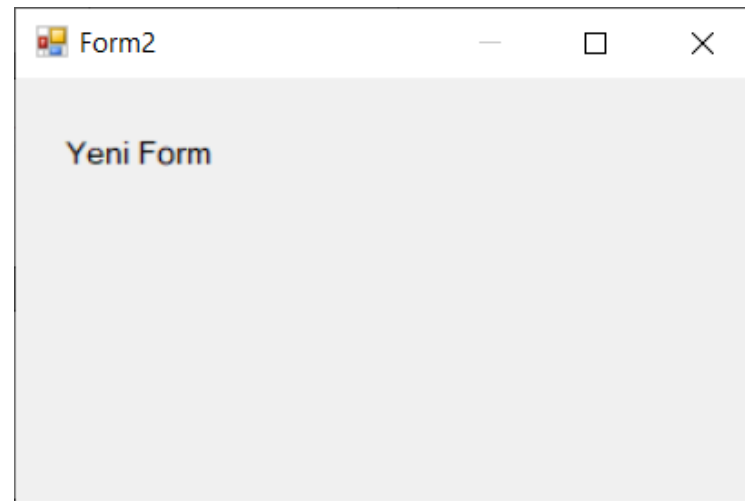
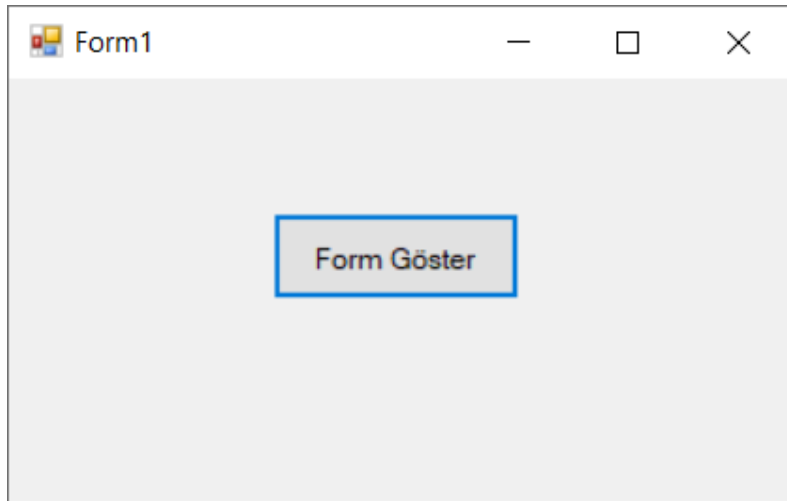
```
private void Form1_Load(object sender, EventArgs e)
{
    this.Location=new Point (150, 250);
}
```



Form Özellikleri

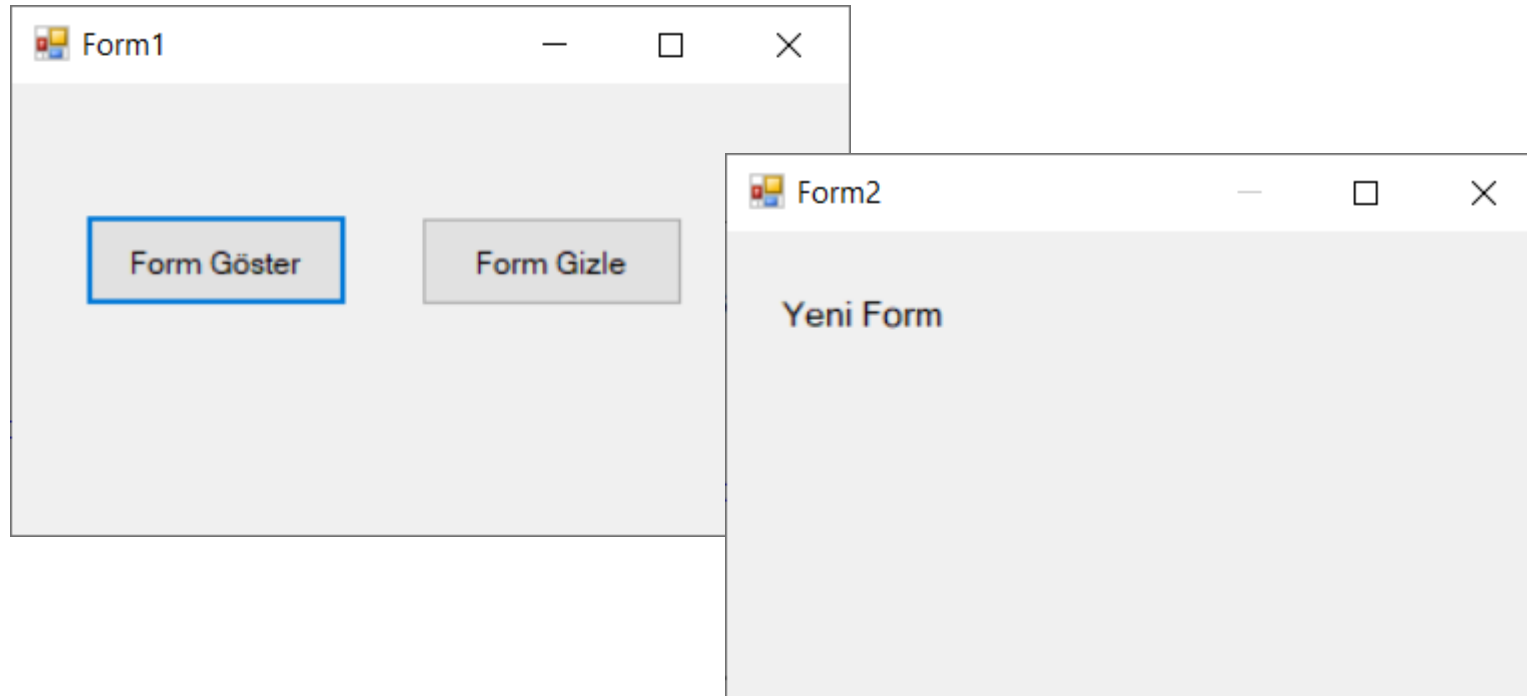
- **BackgroundImage:** Form arka planına resim ekleme
- **Opacity:** Formun şeffaflığını ayarlama
- **Show** metodu birden fazla formu aynı anda ekrana getirmek için, **ShowDialog** ise sadece çalışmak istediğimiz formu ekrana getirmek için kullanılır. Bu form dışındakilere erişilemez.

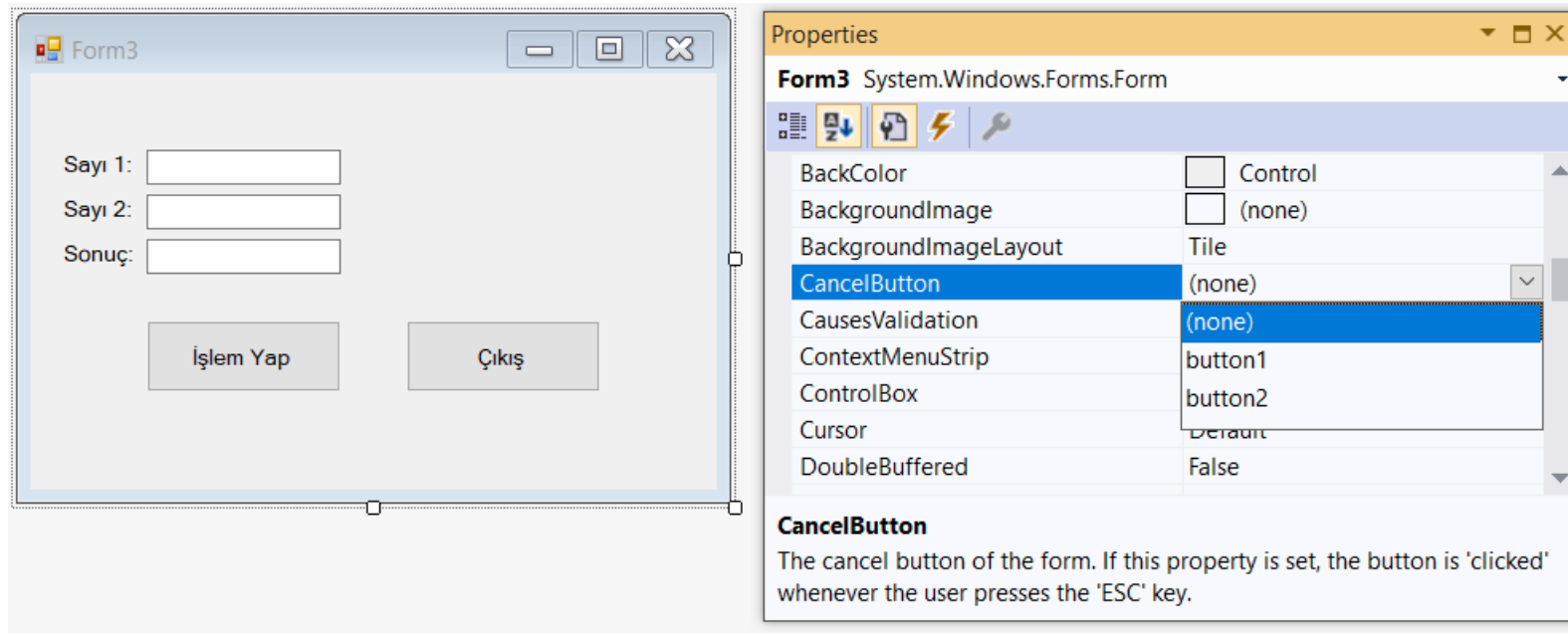
```
private void button1_Click(object sender, EventArgs e)
{
    Form2 yeniForm = new Form2();
    this.Hide();
    yeniForm.Show();
}
```



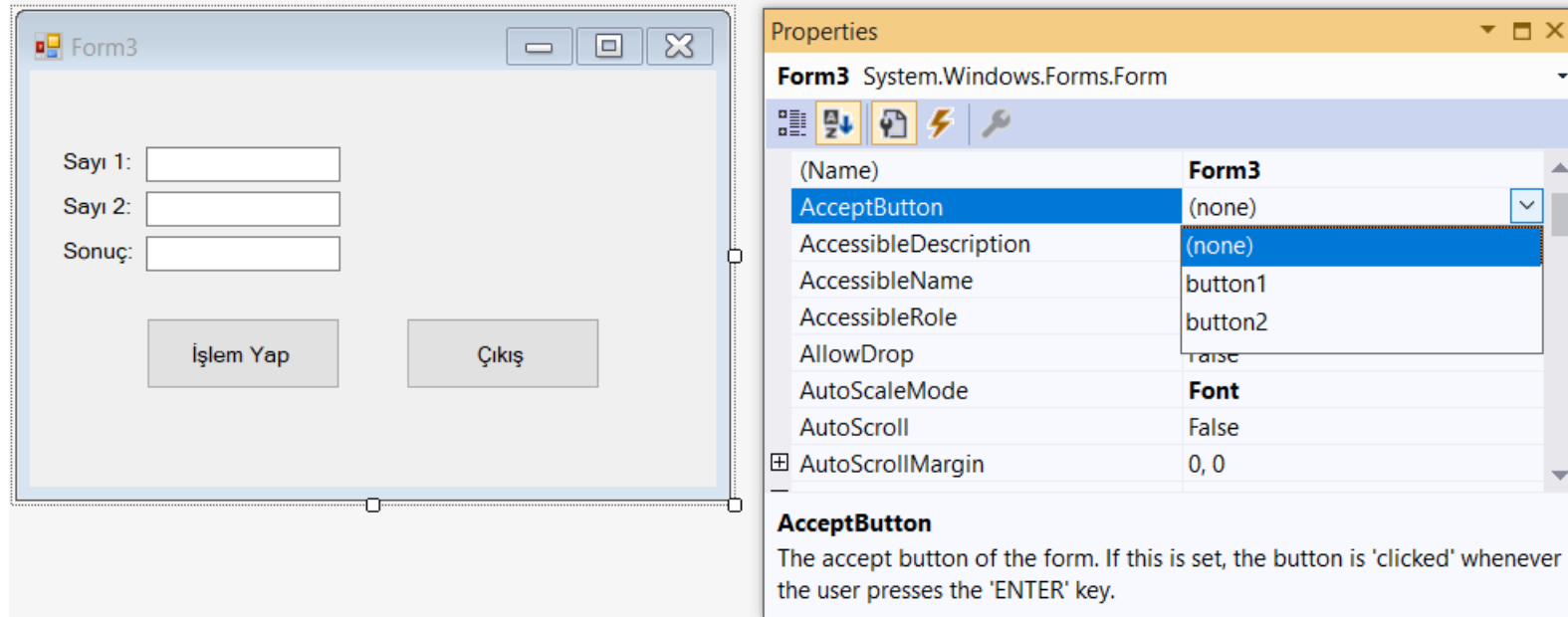
Form Özellikleri

```
Form2 yeniForm = new Form2();  
private void button1_Click(object sender, EventArgs e)  
{  
    yeniForm.Visible = true;  
}  
private void button2_Click_1(object sender, EventArgs e)  
{  
    yeniForm.Visible = false;  
}
```





- **İptal (CancelButton) düğmesi**
Esc ile aynı işleve sahiptir. Forma yerleştirmiş olduğunuz düğmeye benzer işlevi vermek istiyorsanız formun özelliğine söz konusu düğmenin adını aktarmalısınız.



- **Tamam (AcceptButton) düğmesi**
Enter ile tuşu aynı işleve sahiptir. Herhangi bir anda Enter tuşuna basıldığında formdaki düğmelerden birisi tıklanmış gibi işlem yapılmasını istiyorsanız o düğmenin adını formun AcceptButton özelliğine aktarmalısınız.



Form Özellikleri

Özellik	Açıklama
AcceptButton	Form üzerinde Enter tuşuna basıldığı zaman “tıklanacak” Button kontrolü
CancelButton	Form üzerinde Esc tuşuna basıldığı zaman “tıklanacak” Button kontrolü
Opacity	Formun şeffaflık oranı (0 -1 arası)
MaximizeBox	Ekranı Kapla düğmesinin görünürlüğü
MinimizeBox	Simge Durumunda Küçült düğmesinin görünürlüğü
ControlBox	Close, Maximize ve Minimize düğmelerinin tümünün görünürlüğü
StartPosition	Form açıldığı zaman, ekran üzerindeki konumu
TopMost	Formun tüm pencerelerin üzerinde gözükmesi
FormBorderStyle	Formun kenar stili
MaximumSize	Formun alabileceği maksimum büyüklük
MinimumSize	Formun alabileceği maksimum büyüklük



Form Olayları

Olay	Açıklama
Click	Form üzerine tıklandığı zaman gerçekleşir
Closing	Form kapanmadan hemen önce gerçekleşir
Closed	Form kapandıktan sonra gerçekleşir
Load	Form yüklenirken gerçekleşir
KeyDown	Form üzerindeyken bir tuşun basılması ile gerçekleşir
KeyUp	Basılan tuşun kaldırılması ile gerçekleşir

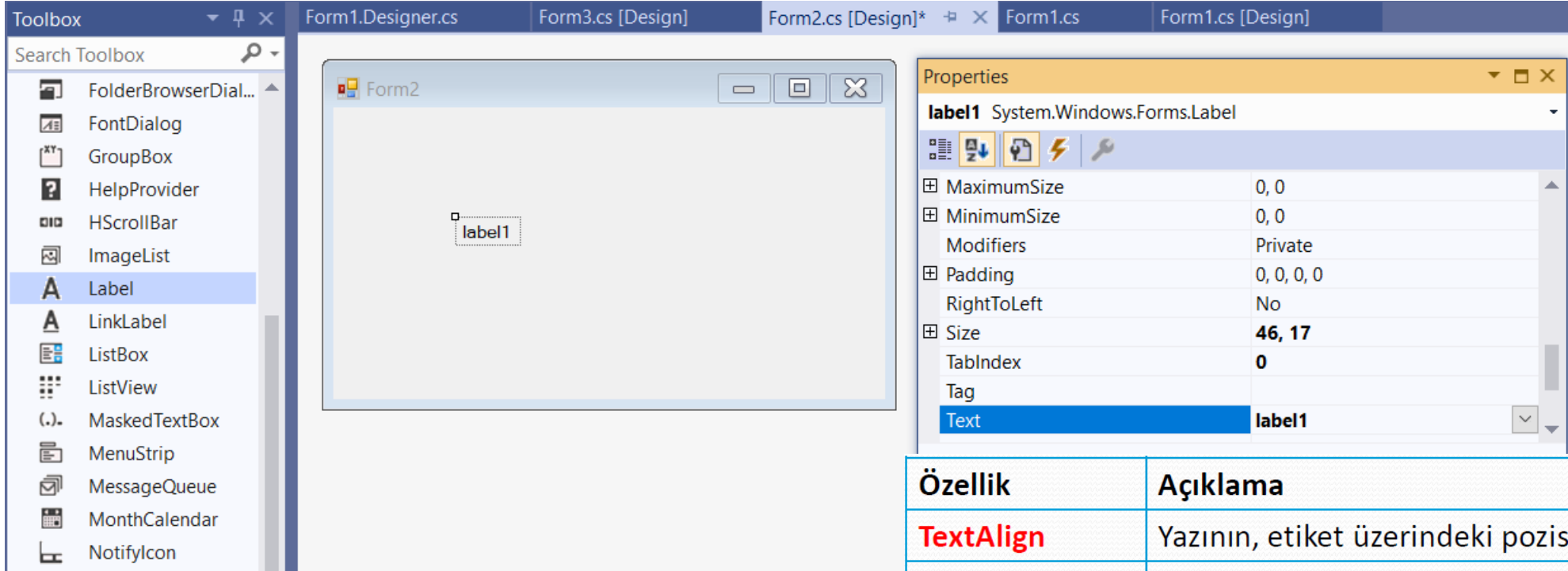
Form Metodları

Metot	Açıklama
Hide	Formu Visible özelliğini False yaparak, gizler
Close	Formu kapatır. Eğer form başlangıç formuysa uygulama sonlanır
Show	Formu gösterir. Hide ile gizlenmişse, Visible özelliği True yapılır.
ShowDialog	Formu diyalog kutusu olarak gösterir.



Label

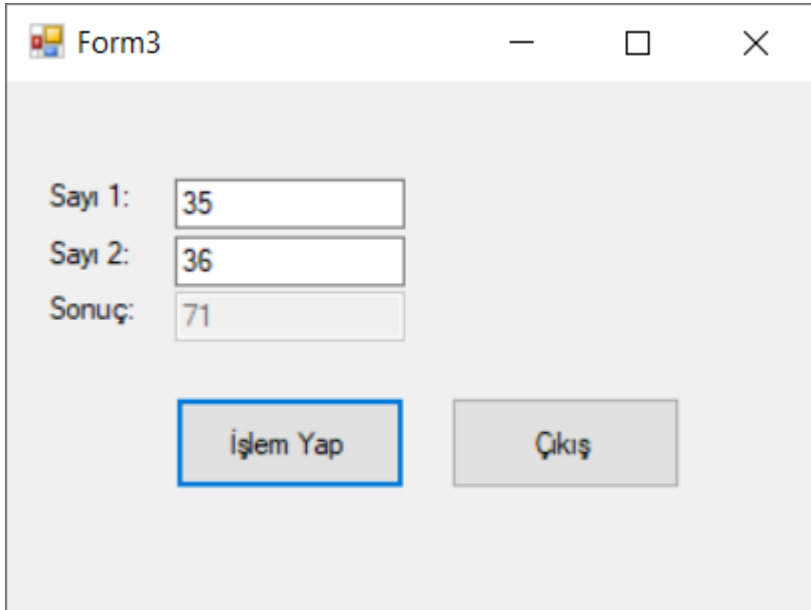
- Label kontrolü Form üzerinde kullanıcıya bilgi vermek amaçlı kullanılan etikettir.



Özellik	Açıklama
TextAlign	Yazının, etiket üzerindeki pozisyonu belirler.
BorderStyle	Kontrolün kenar stilidir. FixedSingle değeri, kontrolün kenar çizgilerini gösterir. Fixed3D değeri, kenarların üç boyutlu olmasını sağlar
Image	Etiket üzerinde görüntülenmek istenen resmi tutar
ImageAlign	Etiket üzerindeki resmin nerede duracağını belirler
RightToLeft	Etiket üzerindeki yazının yönünü belirler. Eğer Yes değerini alırsa, yazılar sağdan sola gösterilir

TextBox

- Metin kutuları, kullanıcıdan bilgi almak için kullanılır.



The screenshot shows a Windows application window titled "Form3". Inside the window, there are three text boxes arranged vertically. The first is labeled "Sayı 1:" and contains the number "35". The second is labeled "Sayı 2:" and contains the number "36". The third is labeled "Sonuç:" and contains the number "71". Below these text boxes, there are two buttons. The first button is labeled "İşlem Yap" and has a blue border, indicating it is the active or default button. The second button is labeled "Çıkış".

```
private void button1_Click(object sender, EventArgs e)
{
    int sayi1 = Convert.ToInt32(textBox1.Text);
    int sayi2 = Convert.ToInt32(textBox2.Text);
    textBox3.Text = Convert.ToString(sayi1+sayi2);
}
```

```
private void button2_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```



TextBox

- **KeyDown:** Bir tuşa basıldığı durumda meydana gelen olay
- **KeyUp:** Tuştan elin çekildiği durumda meydana gelen olay
- **KeyPress:** Bir tuşa basılma ve çekme anı arasında meydana gelen olaydır.
- Formun üzerine yerleştirilen nesnelerin **KeyDown** metodu yerine **formun KeyDown metodunun** işletilmesini istiyorsanız formun **KeyPreview** özelliğini **true** yapmalısınız. Bu özellik **true** iken öncelik formun **KeyDown** olayını temsil eden metoda verilir ve aktif nesnenin **KeyDown** metodu, formun **KeyDown** metodunun çalışması sona erdikten sonra işletilir. Bu durum **KeyUp** ve **KeyPress** için de geçerlidir.

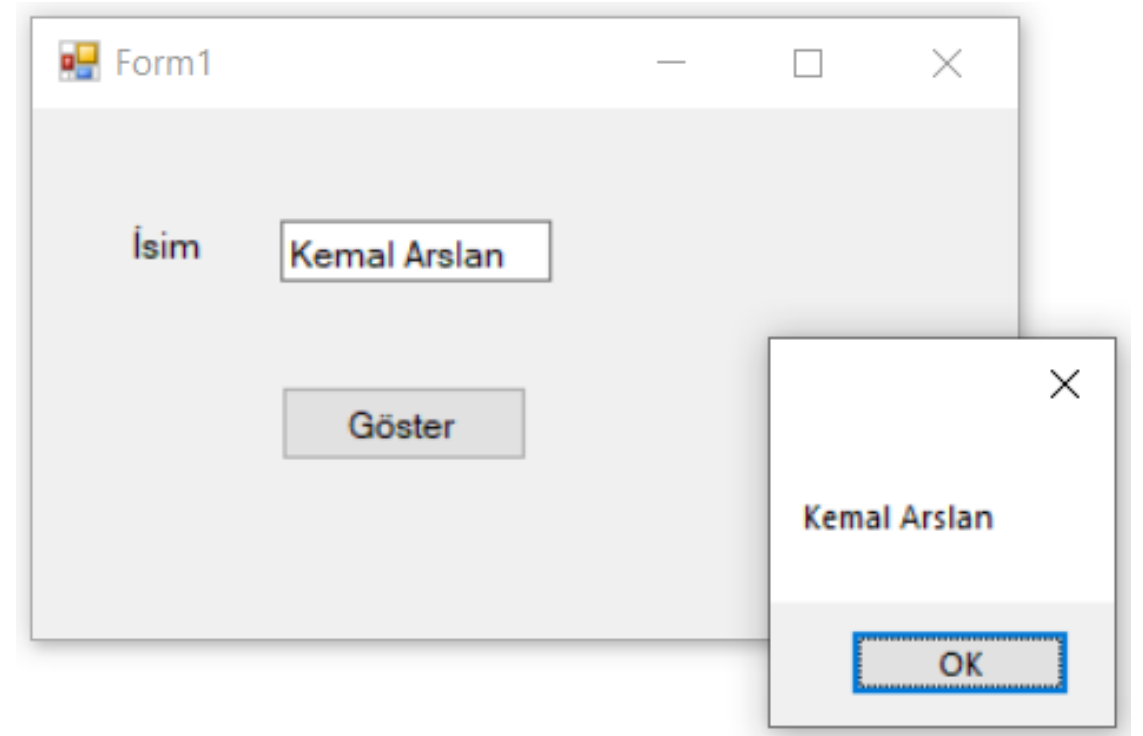


TextBox

```
private void textBox2_KeyDown(object sender, KeyEventArgs e)
{
    if (e.KeyCode == Keys.Enter) // Enter Tuşu
        MessageBox.Show(textBox2.Text);
}
```

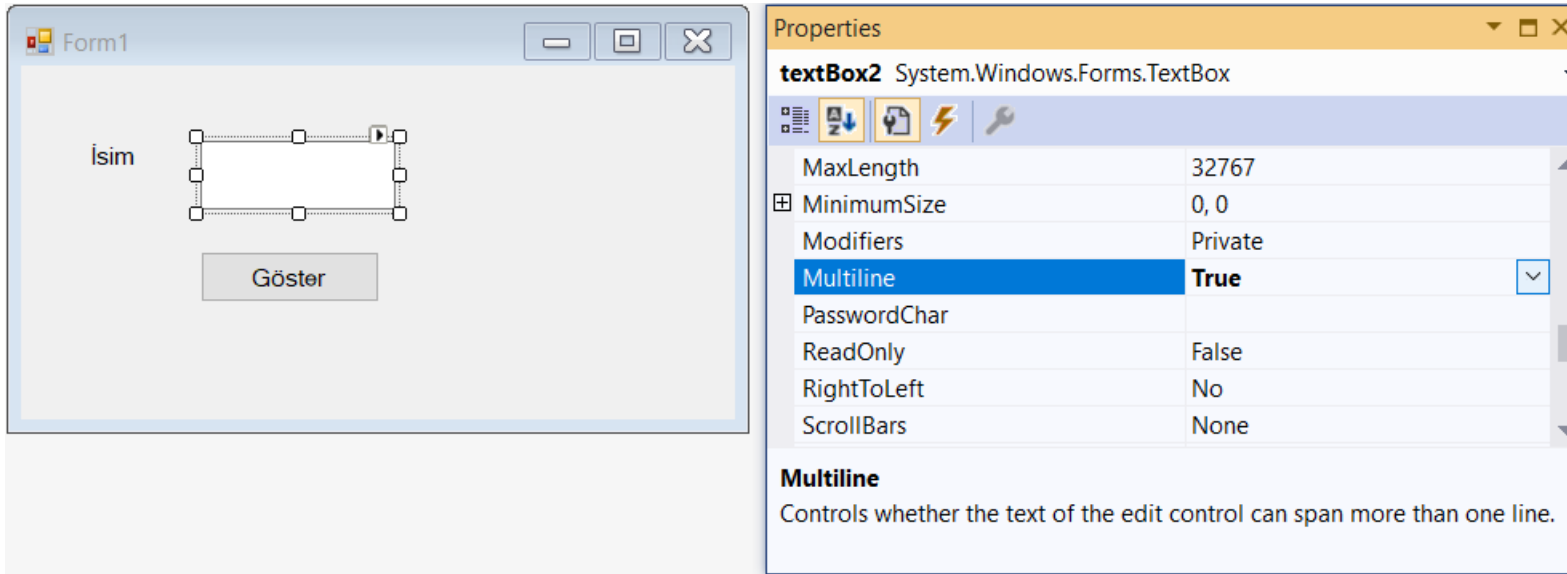
Yukarıdaki komut TextBox'ın KeyDown olayına yazılmıştır.

ENTER tuşuna basıldığında, TextBox'a yazılmış olan metin MessageBox vasıtasıyla gösterilmektedir.



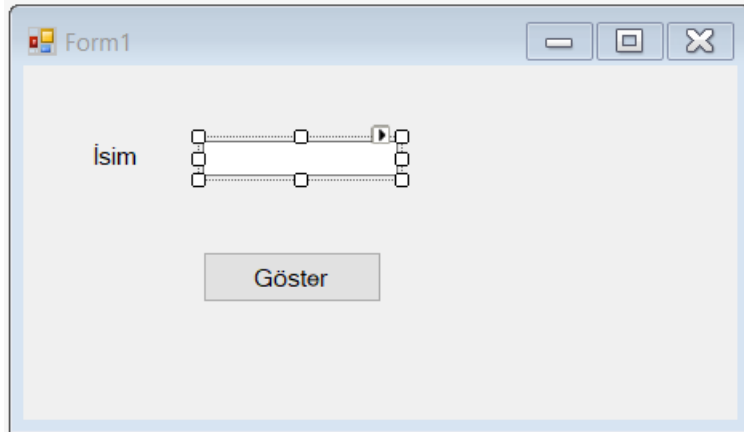
TextBox

- **MultiLine Özelliği:** Metin kutusuna bilginin çoklu satır olarak girilmesini sağlar.
- **ScrollBars Özelliği:** Eğer satır sayısı metin kutusunun boyutundan fazla ise kaydırma çubuklarına gerek duyulabilir. Kaydırma çubukları eklemek için ScrollBars özelliği kullanılır, bu özellik dört değişik değer alır.

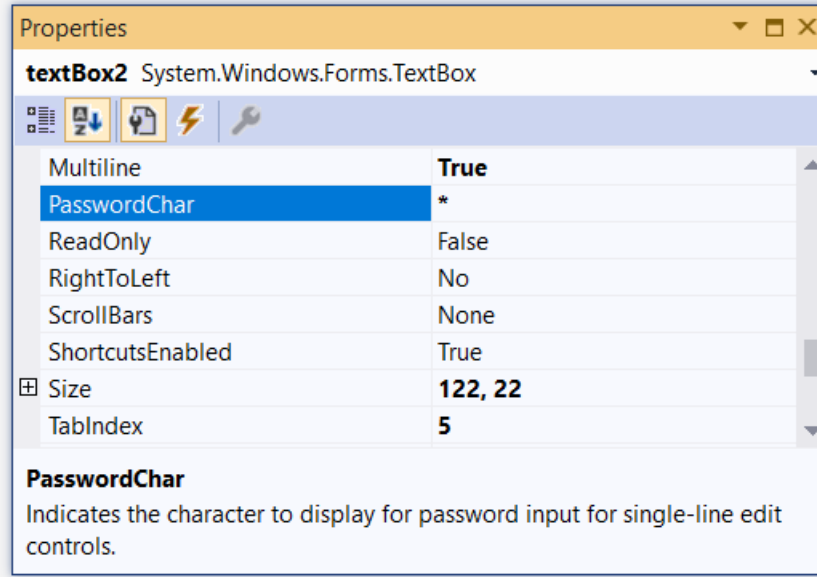


TextBox

- **PasswordChar Özelliği:** Metin kutusuna girilen bilginin belirlenen karakter ile gizlenmesini sağlar.

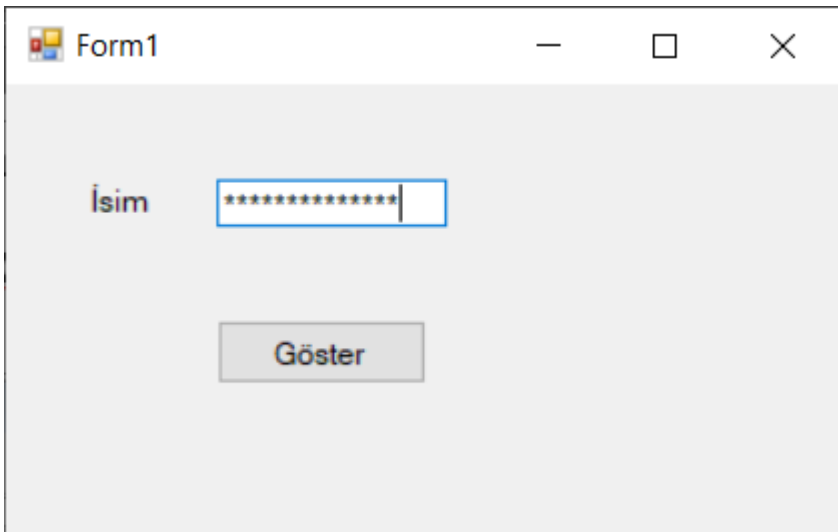


The screenshot shows a Windows Form titled 'Form1'. Inside the form, there is a text box with a password character mask (asterisks) and a button labeled 'Göster'.



The screenshot shows the Visual Studio Properties window for 'textBox2' (System.Windows.Forms.TextBox). The 'PasswordChar' property is highlighted and set to '*'. Below the properties, the 'PasswordChar' property is described: 'Indicates the character to display for password input for single-line edit controls.'

Property	Value
Multiline	True
PasswordChar	*
ReadOnly	False
RightToLeft	No
ScrollBars	None
ShortcutsEnabled	True
Size	122, 22
TabIndex	5



The screenshot shows the same Windows Form titled 'Form1' as above, but with the text box now displaying a password character mask (asterisks) and the 'Göster' button.



TextBox Özellikleri

Özellik	Açıklama
MultiLine	Metin kutusuna birden fazla satırda değer girilebilmesini sağlar. False durumunda ise, metin kutusunun yüksekliği değiştirilemez
ScrollBars	Metin kutusunda kaydırma çubuklarının görünmesi. Varsayılan olarak kaydırma çubuğu görüntülenmez, ancak Horizontal, Vertical kaydırma çubukları ya da ikisi birden gösterilebilir.
PasswordChar	Metin kutusuna parola girilecekse, girilen karakterlerin hangi karakter olarak görüneceğini belirler
WordWrap	Metin kutusuna girilen değerlerin, satır sonlandığında bir alt satıra geçilmesini belirtir. Eğer MultiLine özelliği False ise, alt satırlar tanımlı olmayacağı için bu özelliğin bir etkisi görülmez.
MaxLength	Metin kutusunun alabileceği maksimum karakter sayısını belirtir.
ReadOnly	Metin kutusunun yazmaya karşı korumalı olduğunu belirtir.
CharacterCasing	Metin kutusuna karakterler girilirken büyük veya küçük harfe çevrilmesini sağlar. Upper değeri büyük, Lower değeri küçük harfe çevrimi sağlar.



TextBox Olayları

Olay	Açıklama
TextChanged	Metin kutusundaki yazı değiştiği zaman gerçekleşir.
KeyPress, KeyPreview, KeyDown	Bir tuşa basılma durumunda gerçekleşir

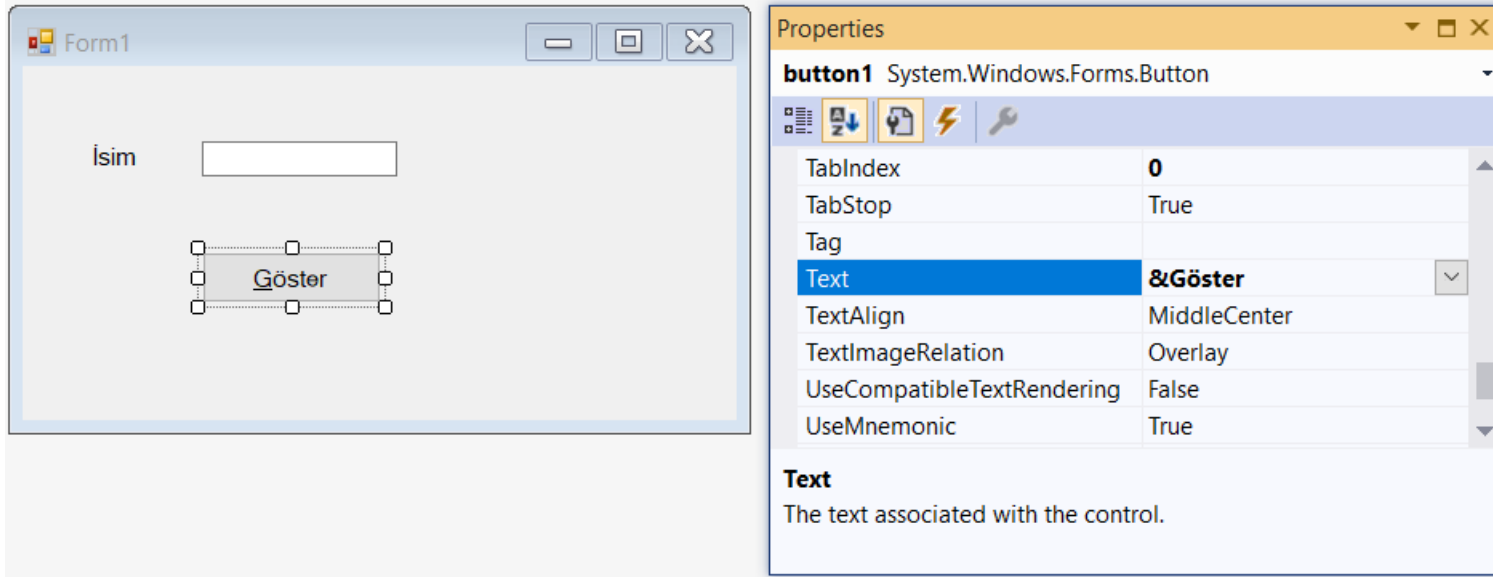
TextBox Metodları

Metot	Açıklama
Cut	Seçilen karakterleri siler ancak hafızada tutar.
Copy	Seçilen karakterleri kopyalar
Paste	Hafızaya alınan karakterleri metin kutusuna yapıştırır
Clear	Metin kutundaki yazıları temizler
SelectAll	Metin kutusundaki tüm yazıyı seçer



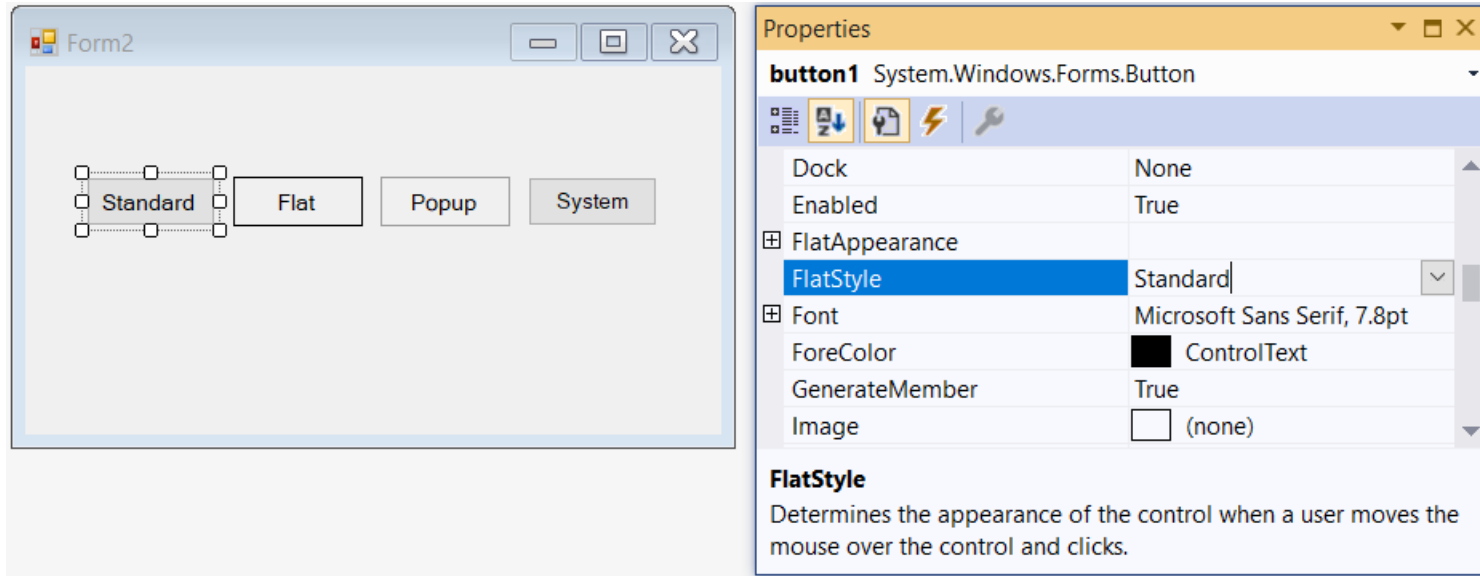
Button

- Windows uygulamalarında, form üzerinde komut düğmeleri olarak kullanılır.
- Kısayol tuşları ile buton kontrolüne erişmek için & işareti kullanılır. Örneğimizde **Alt+ G** tuşuna basarak butona tıklanma sağlanır.



Button

- **Enabled ve Visible Özellikleri:** Butonun aktif veya görünür olmasını kontrol eder.
- **TabIndex ve TabStop Özellikleri**
TabStop false olursa o nesne atlanır.
- **FlatStyle Özelliği**
Dört farklı değer alabilir. Bunlar Standart, popup, flat, system' dir. Görünüm şekilleri aşağıdaki gibidir.

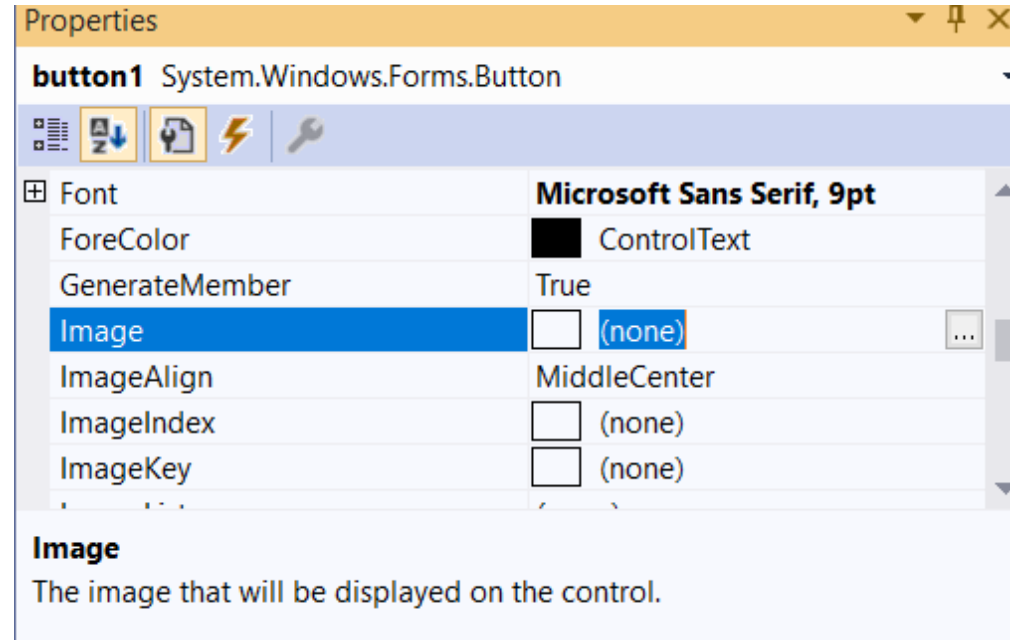


Button

- **PerformClick() metodu:**
- Butonların çalışması için üzerine tıklamak gerekir. Ancak **PerformClick()** metodu ile buton tıklanmış gibi bir etki sağlanır.

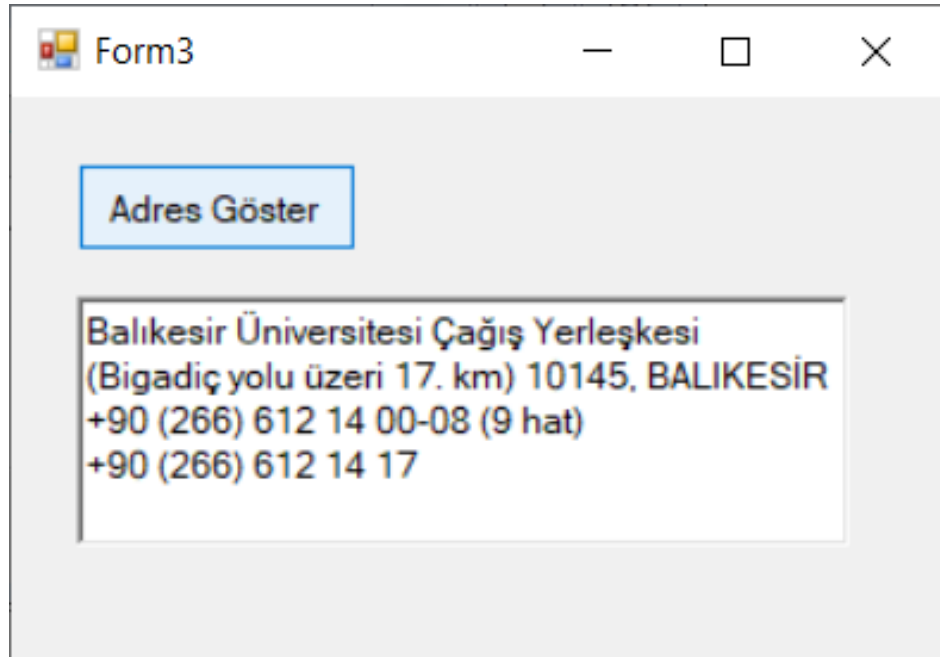
```
private void Form1_Load(object sender, EventArgs e)
{
    button1.PerformClick();
}
```

- **Image Özelliği:** Butonların üzerine resim yerleştirmek için kullanılır.



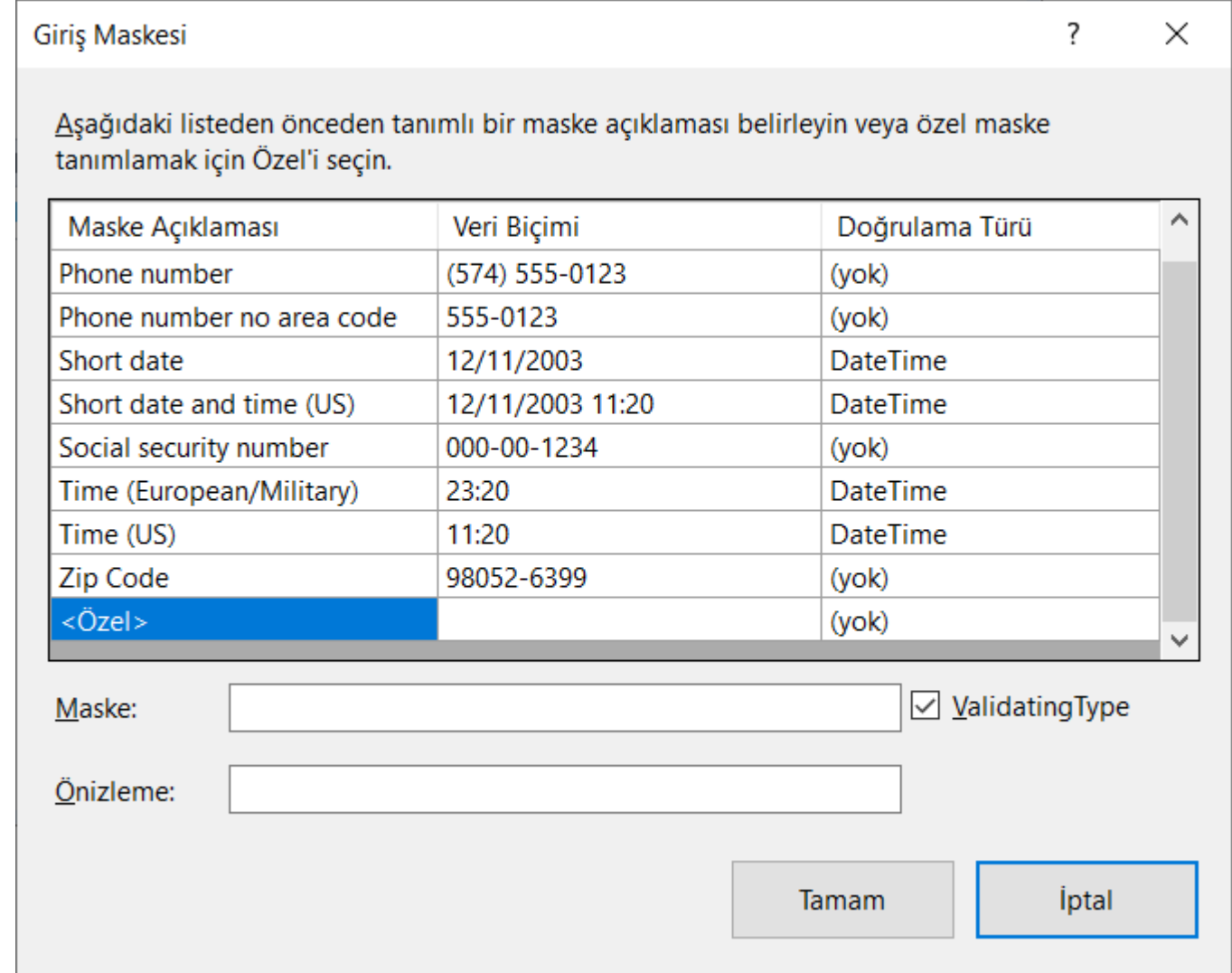
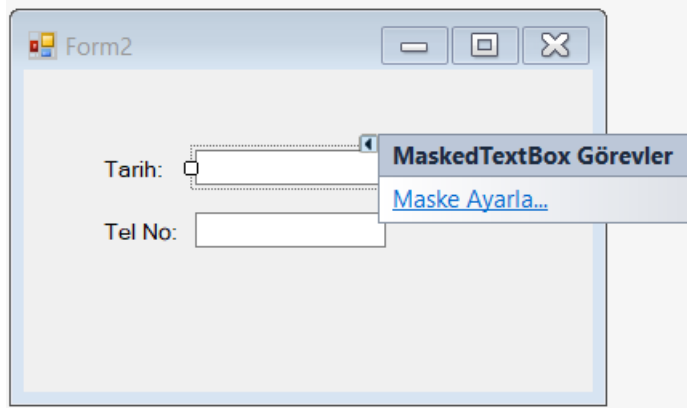
RichTextBox

- Zengin metin kutusu anlamına gelmektedir. Normal metin kutusundan farklı olarak alt satıra da yazmaya imkan vermektedir.
- Kod yapısı olarak Lable veya TextBox ile aynı yapıda kullanılmaktadır.

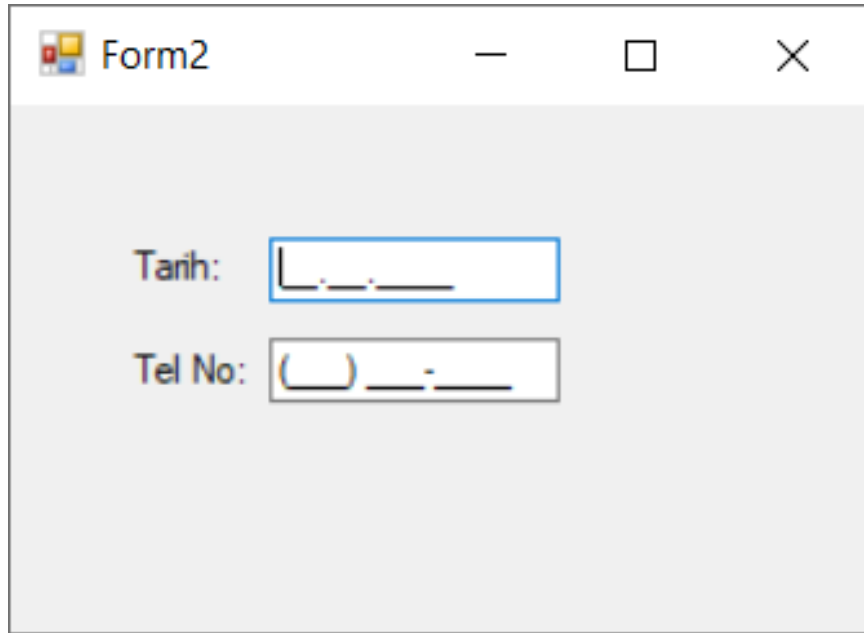


MaskedTextBox

Maskeli metin kutusu anlamına gelir. Güçlü tasarımlar konusunda çok sıklıkla kullanılır. Belirli bir formatta alınması gereken veri girişi için kullanılır. Alabileceği değerler aşağıdaki görselde görünmektedir.



MaskedTextBox

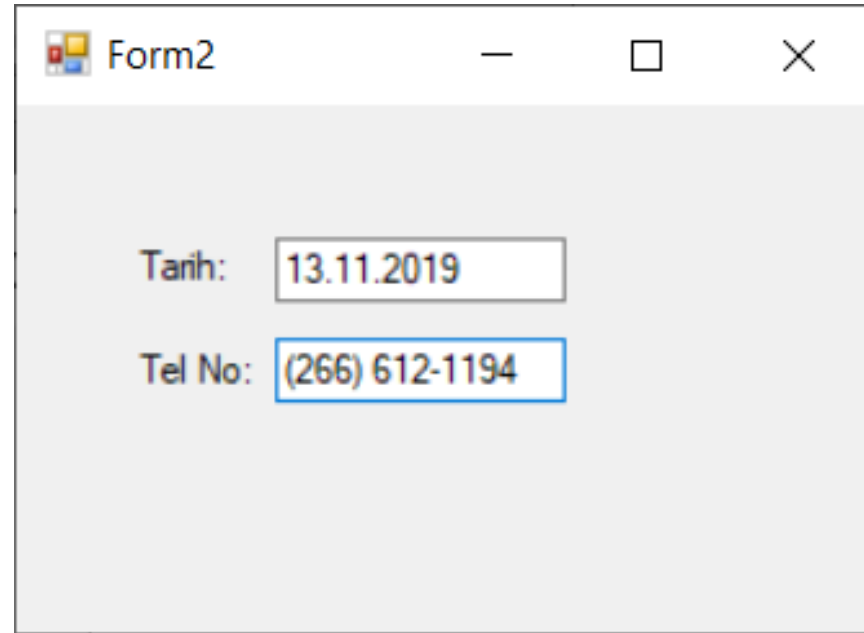


Form2

Tarih:

Tel No:

The image shows a Windows form titled 'Form2' with a light gray background. It contains two labels, 'Tarih:' and 'Tel No:', each followed by a MaskedTextBox control. The date mask is 'dd.MM.yyyy' and the phone number mask is '(000) 000-0000'. The controls are currently empty, showing only the mask characters.



Form2

Tarih:

Tel No:

The image shows the same 'Form2' window, but now the MaskedTextBox controls contain data. The 'Tarih' control displays '13.11.2019' and the 'Tel No' control displays '(266) 612-1194'. The controls have a blue border, indicating they are selected or have focus.



CheckBox

- Kontrol Kutusu, kullanıcıya birden çok seçeneği seçme imkanı sağlar. Checked özelliği kontrol kutusunun seçilip seçilmediğini kontrol eder. Seçili ise , true değilse false değerini alır.

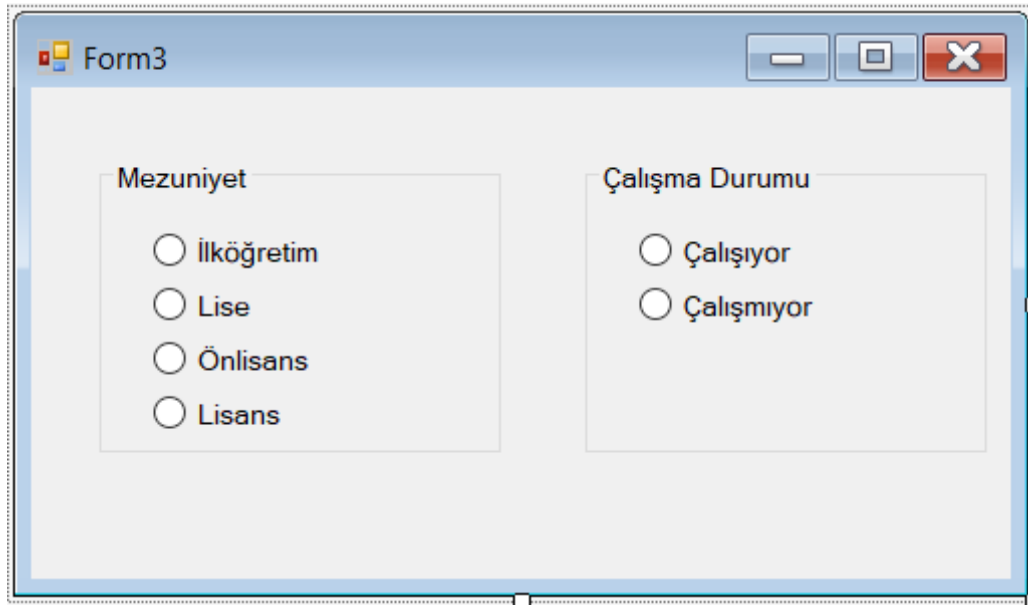
Özellik	Açıklama
checked	Kontrolün seçili olup olmadığını belirler
checkAlign	Seçme kutusunun ve üzerinde yazan metnin birbirlerine göre konumlarını belirler
Appearance	Kontrolün seçme kutusu ya da düğme şeklinde olmasını belirler
ThreeState	Seçili olup olmaması dışında, Intermediate durum da eklenir. Eğer kontrol Intermediate durumundaysa checked özelliği True olur.
AutoChecked	Kontrole basıldığı zaman seçili duruma geçileceğini belirtir. Eğer bu özellik False ise, kontrolün durumunu değiştirmek için, click olayında, checked özelliğini güncellemek gerekir

Olay	Açıklama
checkChanged	Seçme kutusunun durumu değiştiği zaman gerçekleşir.



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.

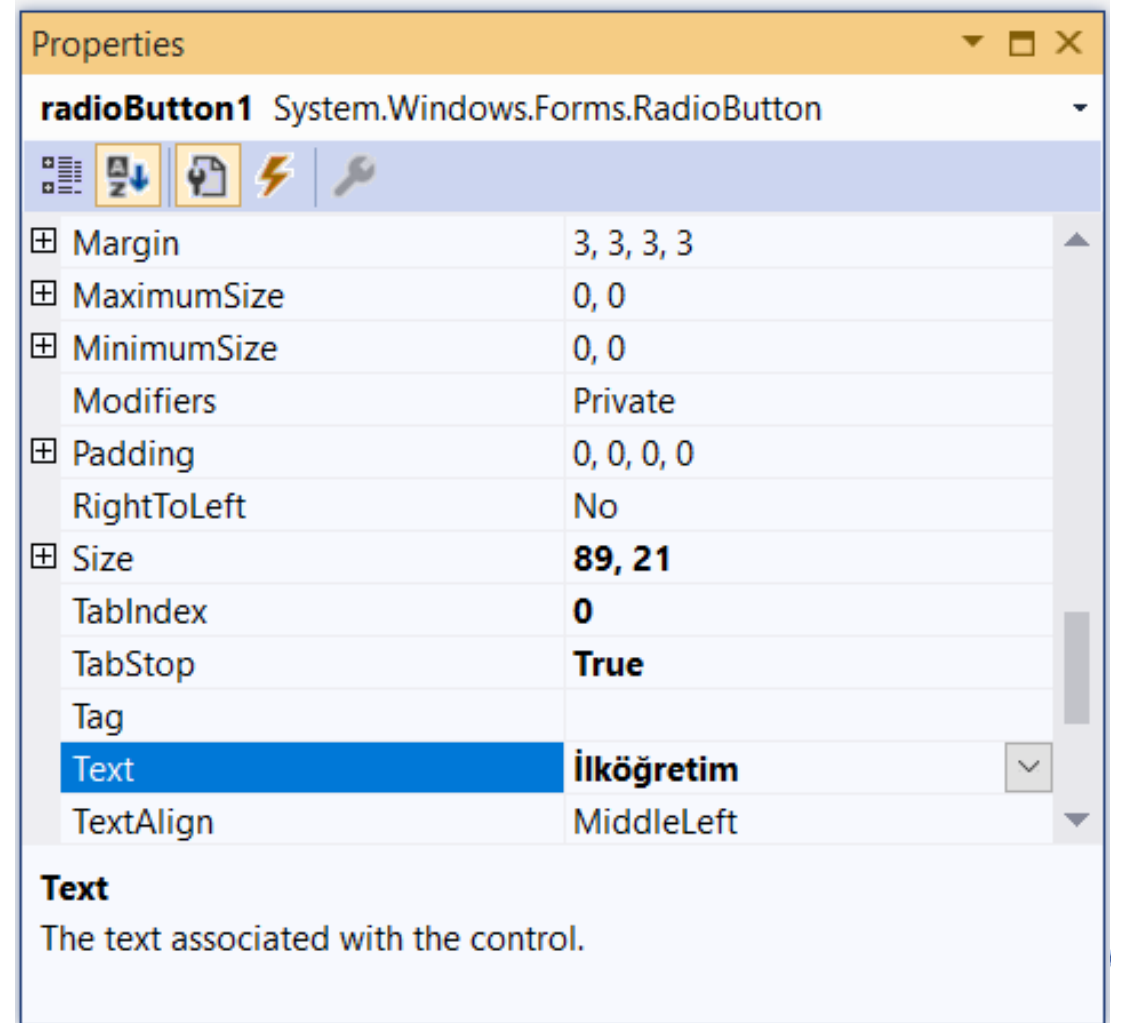


The screenshot shows a Windows application window titled "Form3". Inside the window, there are two GroupBox controls. The first GroupBox, titled "Mezuniyet", contains four radio buttons with the following labels: "İlköğretim", "Lise", "Önlisans", and "Lisans". The second GroupBox, titled "Çalışma Durumu", contains two radio buttons with the following labels: "Çalışıyor" and "Çalışmıyor". The window has standard Windows window controls (minimize, maximize, close) in the top right corner.

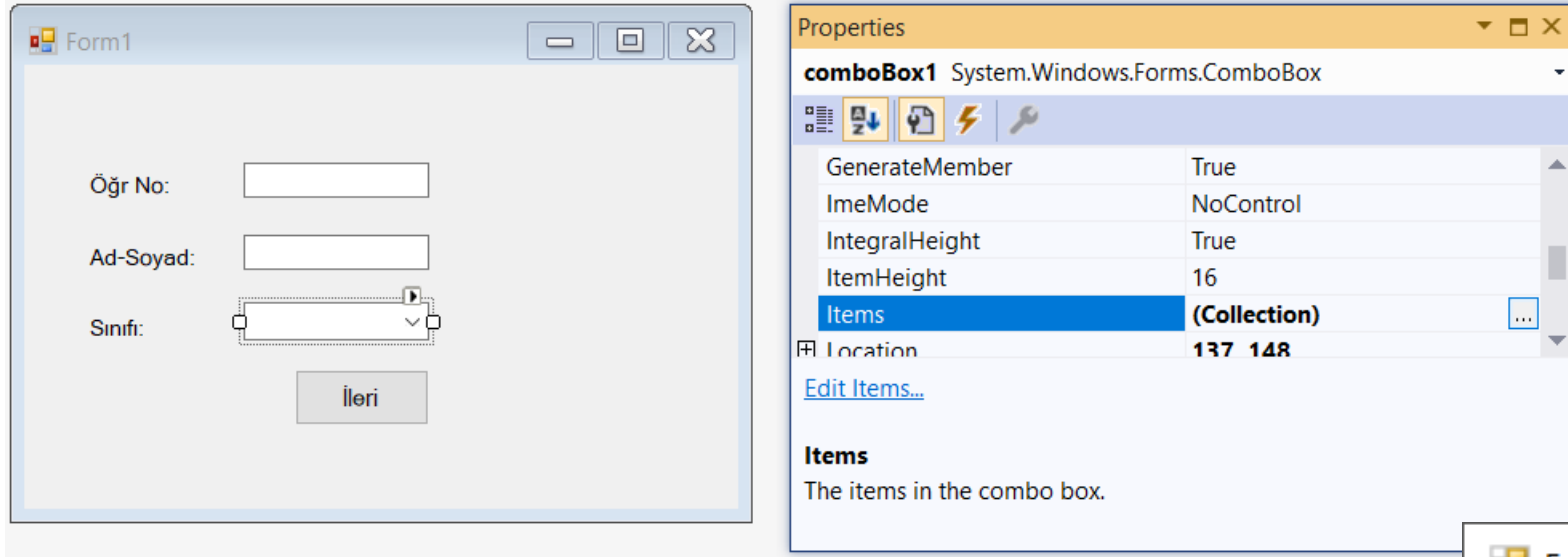


RadioButton

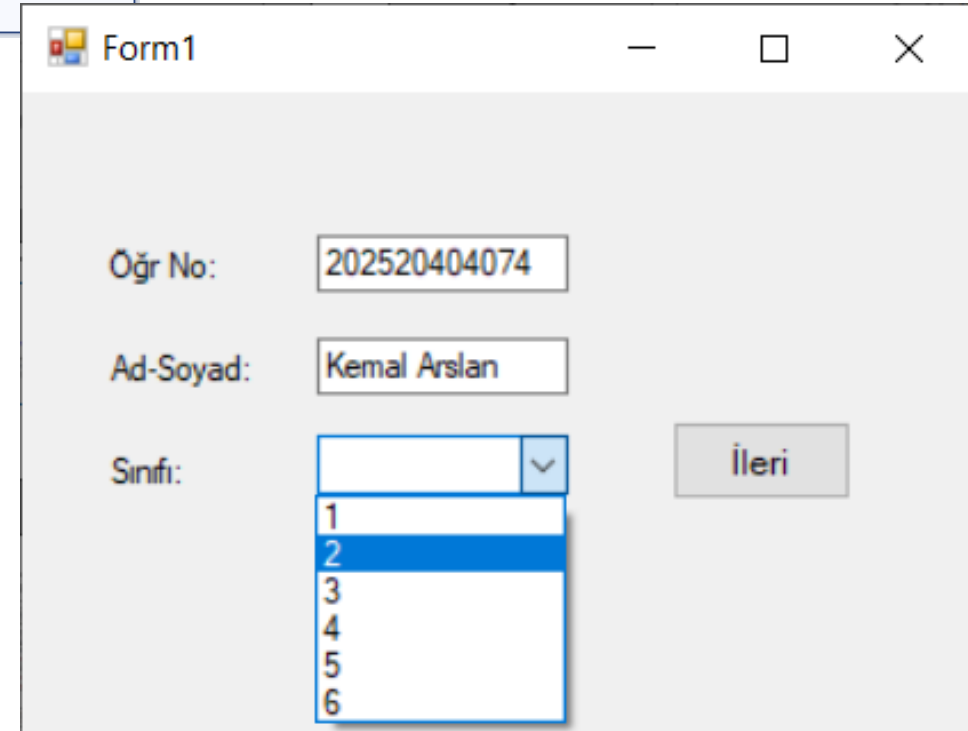
- Bu kontrolün özellikleri CheckBox'a çok benzemektedir. CheckBox'ta olduğu gibi;
- **Appearance** özelliği ile komut düğmesi görünümüne sahip RadioButton düğmeleri oluşturulabilir.
- **Text** özelliği ile içindeki metin yazılabilir.
- **TextAlign** özelliği ile içindeki metnin yerleşmesi belirlenebilir.
- **Image** özellikleri ile içinde resim gösterilebilir.
- **CheckAlign** özelliği ile seçenek düğmesi sola, sağa alınabilir.



ComboBox



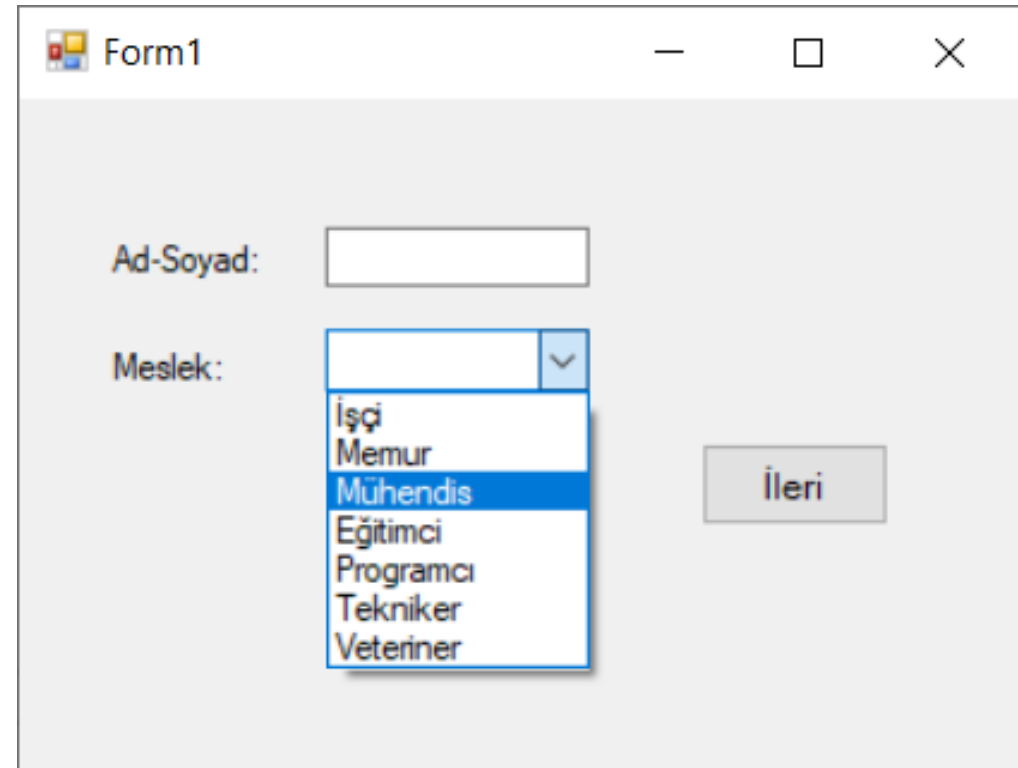
- Açılır liste üzerinden sunulan seçenekler arasından seçim yapılmasına olanak sağlayan araçtır.
- Listelenen öğeler açılan bir kutuda görüntülenir ve listeden en fazla bir tane öğe seçilebilir.
- **Combobox**'ın **Items** özelliği kullanılarak açılır listeye eleman eklenebilir.



ComboBox

- 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" };
    comboBox1.Items.AddRange(meslekler);
}
```



The screenshot shows a Windows application window titled "Form1". Inside the window, there are two labels: "Ad-Soyad:" and "Meslek:". The "Ad-Soyad:" label is followed by an empty text box. The "Meslek:" label is followed by a ComboBox. The ComboBox is open, displaying a list of professions: "İşçi", "Memur", "Mühendis", "Eğitimci", "Programcı", "Tekniker", and "Veteriner". The "Mühendis" item is currently selected and highlighted in blue. To the right of the ComboBox, there is a button labeled "İleri".



ComboBox

Özellik	Açıklama
DropDownStyle	Kontrolün listeleme stilini belirler. Simple stil, listedeki sadece bir öğeyi görüntüler. DropDown stili, listenin tüm elemanlarını görüntüleyerek seçilmelerini ve kullanıcının değer girmesini sağlar. DropDownList kullanıcının değer girmesini engeller.
DropDownWidth	ComboBox kontrolünün açılan listesinin genişliğini belirler.
MaxDropDownItems	Kontrole eklenebilecek maksimum öğe sayısını belirler.
MaxLength	Kullanıcının girebileceği maksimum karakter sayısını belirler.
SelectedText	Seçilen öğenin görüntülenen yazısını belirler.

The screenshot displays a Windows Forms application window titled "Form1". Inside the form, there are two labels: "Ad-Soyad:" followed by a text box, and "Meslek:" followed by a ComboBox control. The ComboBox has a dropdown arrow and the text "Seçim Yapınız" is visible. Below these controls is a button labeled "İleri". To the right of the form, the "Properties" window is open, showing the properties for the selected "comboBox1" control. The "Text" property is highlighted, and its value is "Seçim Yapınız".

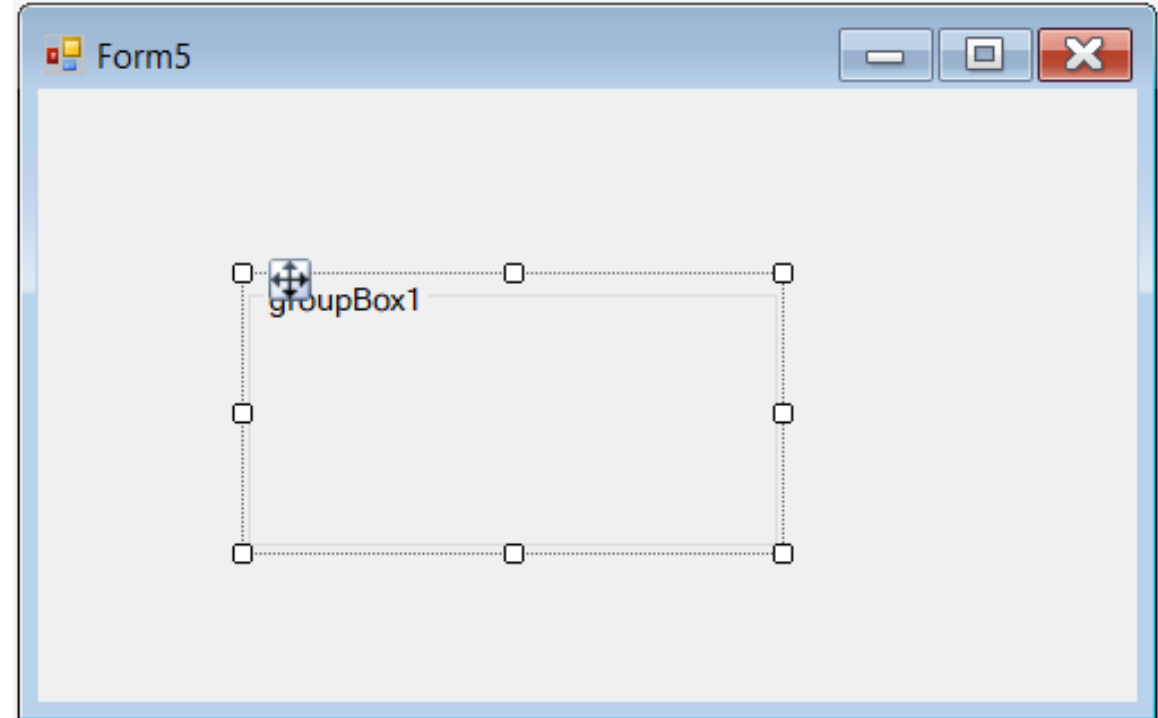
Properties	
comboBox1 System.Windows.Forms.ComboBox	
TabIndex	6
TabStop	True
Tag	
Text	Seçim Yapınız
UseWaitCursor	False
ValueMember	
Visible	True

Text
The text associated with the control.



GroupBox

- Bu kontrol tek başına değil, diğer kontrolleri gruplamak için kullanılır. Kontrolleri bu GroupBox ile gruplamanın birçok avantajı vardır.
- GroupBox içine yerleştirilen kontroller, GroupBox'a bağımlıdırlar ve konumları bu çerçeve dışına taşamaz. Özellikle birkaç kontrolü birden görünür ya da görünmez yapmak için hepsinin Visible özelliğini tek tek değiştirmek yerine çerçevenin Visible özelliğini değiştirilerek çerçeve içindeki tüm kontroller aynı anda değiştirilebilir.
- Her bir kontrol tek tek taşınmak yerine çerçeve taşınır. Çerçevelerin buna benzer pek çok faydaları vardır. Ayrıca RadioButton' ların gruplanmasında çerçeve kullanmak kaçınılmaz olabilir.



Örnek

- Aşağıdaki form dizaynında aile bilgileri bölümünün medeni hali bekar olanlar için gösterilmemesi, mezun olduğu fakülte bölümünün ise sadece öğrenim durumu üniversite olması durumunda gösterilmesi gerekir. Aile bilgileri ve mezun olduğu fakülte içindeki bütün kontrolleri tek seferde gizleyip, göstermek için sadece onun içinde bulunduğu GroupBox kontrolünü gizleyip göstermek yeterlidir.

The screenshot shows a Windows application window titled "Form3". Inside the window, there is a form with three main sections, each enclosed in a light gray border:

- Kişisel Bilgiler (Personal Information):** Contains four input fields: "Kimlik No:" (text), "Ad-Soyad:" (text), "Şehir" (dropdown menu), and "Adres:" (text area).
- Medeni Hali (Marital Status):** Contains three radio buttons: "Bekar" (unselected), "Evli" (selected), and "Dul" (unselected).
- Aile Bilgileri (Family Information):** Contains two input fields: "Çocuk Sayısı" (text) and "Eşi Çalışıyor" (checkbox, unselected).

Below the "Medeni Hali" section, there is another section titled "Öğrenim Durumu" (Education Status) with four radio buttons: "İlköğretim" (unselected), "Lise" (unselected), "Önlisans" (unselected), and "Lisans" (selected).

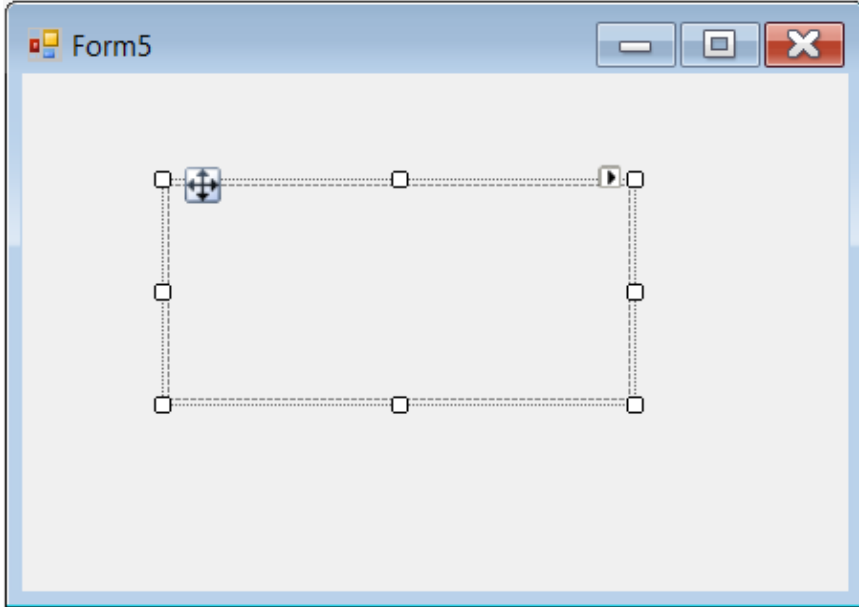
Below the "Aile Bilgileri" section, there is another section titled "Mezun Olduğu Fakülte" (Faculty Graduated From) with four radio buttons: "Mühendislik" (unselected), "İİBF" (unselected), "Fen-Edebiyat" (unselected), and "Diğer" (unselected).

At the bottom right of the form, there is a button labeled "Kaydet" (Save).



Panel

- Kullanılacak grupları kontrol etmek, üzerinde daha rahat işlem yapabilmek için bir arada tutan bir toolbox aracıdır. Panel üzerine istediğimiz toolbox aracını rahatça bırakabilir üzerlerinde işlem yapabiliriz. GroupBox gibi ayarlama yapan bir kontroldür. GroupBox'a göre en büyük artısı kaydırma çubuklarını (AutoScroll) desteklemesidir.



Properties

panel1 System.Windows.Forms.Panel

AutoScroll	True
AutoScrollMargin	0, 0
AutoScrollMinSize	0, 0
AutoSize	False
AutoSizeMode	GrowOnly
BackColor	Control
BackgroundImage	(none)
BackgroundImageLayout	Tile
BorderStyle	None

AutoScroll

Indicates whether scroll bars automatically appear when the control contents are larger than its visible area.



ListBox

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

The image displays a Windows Forms application with a form titled "Form5". On the form, a **ListBox** control is visible, containing the items "Ankara", "İstanbul", and "İzmir".

The **Properties** window for the **listBox1** control is shown, displaying various properties. The **Items** property is highlighted, showing it is a **(Collection)**.

The **String Collection Editor** dialog is open, allowing the user to enter strings for the collection. The dialog contains the text "Enter the strings in the collection (one per line):" and a text area with the following items:

- Ankara
- İstanbul
- İzmir

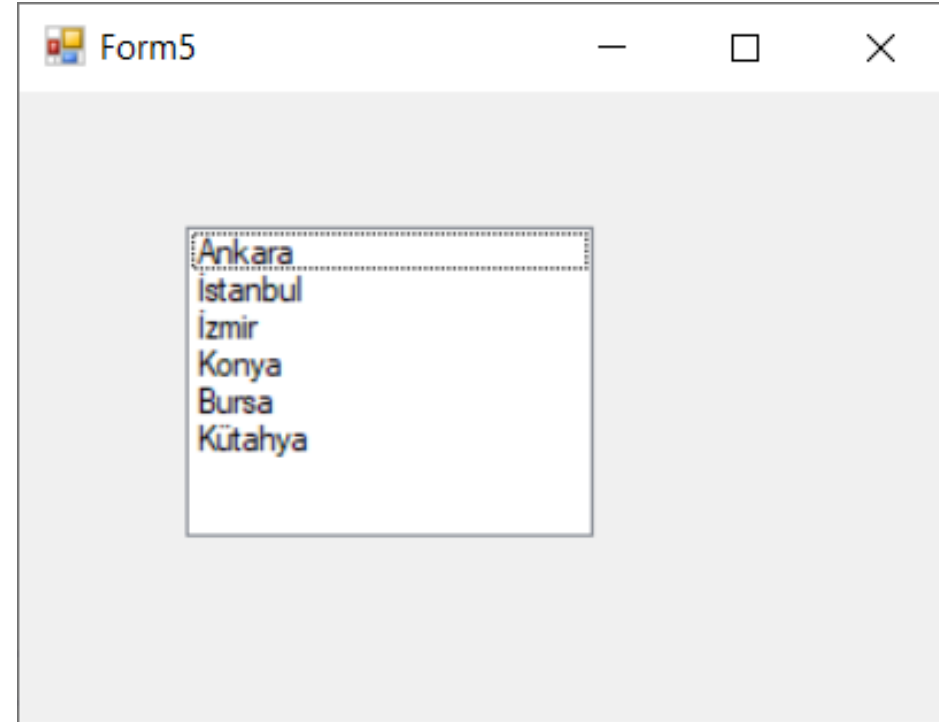
The dialog has **OK** and **Cancel** buttons at the bottom.



ListBox

- Kod ortamında ListBox'a eleman ekleme

```
private void Form5_Load(object sender, EventArgs e)
{
    listBox1.Items.Add("Konya");
    listBox1.Items.Add("Bursa");
    listBox1.Items.Add("Kütahya");
}
```



ListBox

- **AddRange() metodu:** Birden fazla eleman bir seferde listBox nesnesine ekler. String bir dizinin elemanlarını da ekleyebilirsiniz.
`listBox2.Items.AddRange(listBox1.Items);`
`listBox1.Items.AddRange(string dizi_adı);`
- **CopyTo() metodu:** listBox'ın içeriğini bir dizi değişkene aktarır.
`string[] sehir; sehir=new string[3];`
`listBox1.Items.CopyTo(sehir,0);`
- **Insert(sıra_no, eleman) metodu:** İsteddiğimiz bir sıraya eleman yerleştirmek için kullanılır.
`listBox1.Items.Insert(1, "eleman");`
- **Remove() metodu:** Listbox'tan bir eleman silmek için kullanılır.
`listBox1.Items.Remove("eleman");`



ListBox

- **Clear() metodu:** Liste kutusunu tamamen boşaltmak
- **Contains(string) metodu:** Liste içerisinde bir elemanın olup olmadığını kontrol eder varsa true yoksa false değeri geri döner.

```
listBox1.Items.Contains("a");
```

- **FindStringExact(string) metodu:** Contains ile aynı işleve sahip fakat elemanın ilk geçtiği index numarasını geri döndürür.

```
listBox1.FindStringExact("a");
```

- **Count Özelliği:** Liste kutusundaki eleman sayısını verir.

```
private void button1_Click(object sender, EventArgs e)
{
    for (int i = 0; i < listBox1.Items.Count; i++)
    {
        MessageBox.Show("" + listBox1.Items[i]);
    }
}
```

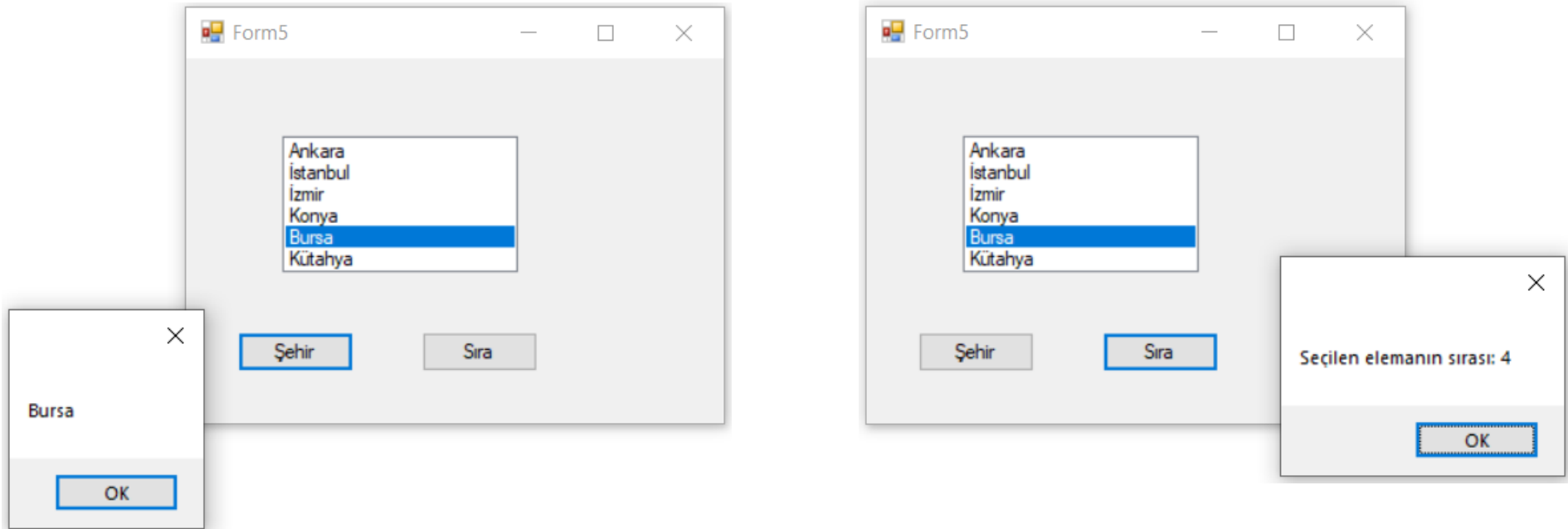


ListBox

- **Text Özelliği:** Liste kutusundaki seçilen elamanın değerini döndürür.
- **SelectedIndex Özelliği:** Liste kutusundaki elamanının indis numarasını verir.

```
private void button1_Click(object sender, EventArgs e)
{
    MessageBox.Show(listBox1.Text);
}
```

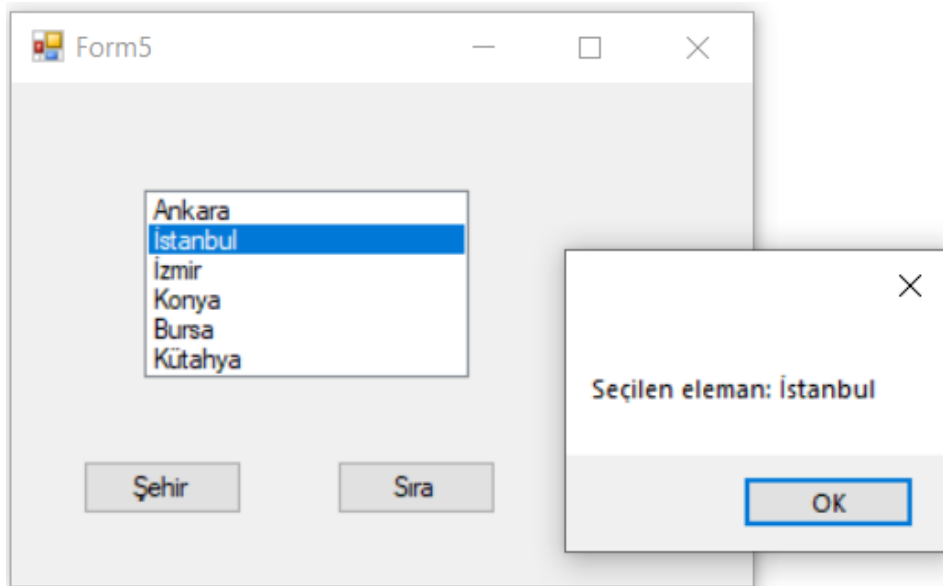
```
private void button2_Click(object sender, EventArgs e)
{
    MessageBox.Show("Seçilen elemanın sırası: " + listBox1.SelectedIndex.ToString());
}
```



ListBox

- **SelectedIndexChanged() metodu:** Liste kutusundan herhangi bir eleman seçildiğinde çalışan olaydır.

```
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    MessageBox.Show("Seçilen eleman: " + listBox1.Text);
}
```



ListBox

Özellik	Açıklama
Items	Liste kutusuna eklenen öğelerin tutulduğu koleksiyon nesnesidir.
SelectedItem	Liste kutusundan seçilen öğeyi alır.
SelectedItems	Liste kutusundan seçilen öğeleri alır. Seçilen öğeler dinamik bir dizide tutulur.
SelectedIndex	Liste kutusundan seçilen öğenin indisini alır.
SelectedIndices	Liste kutusundan seçilen öğelerin indislerini bir koleksiyon nesnesinde tutar.
DataSource	Listenin öğelerinin tutulduğu veri kaynağıdır. Veri kaynağı boş geçilirse Items koleksiyonuna eklenen öğeler görüntülenir.
DisplayMember	Veri kaynağından gelen öğelerin, kullanıcıya gösterilecek özelliğidir.
ValueMember	Veri kaynağından gelen öğelerin, dönüş değerini belirleyen özelliğidir.
SelectedValue	Seçilen öğenin, liste kutusunun ValueMember ile belirtilen özelliğidir.
SelectionMode	Liste kutusundan kaç tane öğe seçilebileceğini belirtir. None değeri 0, One değeri 1, MultiSimple ve MultiExtended değerleri birden fazla öğenin seçilebileceğini belirtir.
MultiColumn	Liste kutusundaki öğelerin birden fazla kolonda görüntülenmesini belirler.



ListBox

Olay	Açıklama
SelectedIndexChanged	Liste kutusunda bir öge seçildiği zaman gerçekleşir.

Metot	Açıklama
GetItemText	Parametre olarak verilen nesnenin liste kutusunda gösterilen yazısını döndürür.
GetSelected	Parametre olarak verilen indisteki öğenin seçili olup olmadığını döndürür.
FindString	Parametredeki String ifadesini liste kutusunda arayarak, bulduğu ilk öğenin indisini döndürür



Örnek

- Tedarikçiden alınacak ve stokta bulunan ürünleri listelemek ve alım satım işlemi yapmak için ListBox kontrolleri kullanma.

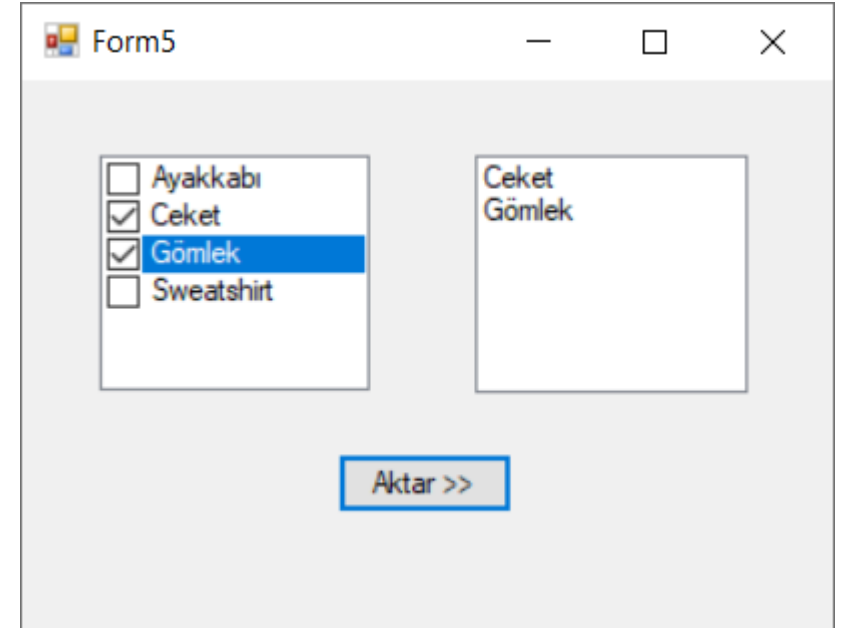
Tedarikçideki Ürünler		Stoktaki Ürünler
Defter - 4,99 TL Cetvel - 1,99 TL Pergel - 2,99 TL	Ekle >>>	Kalem - 1,49 TL Silgi - 0,39 TL Not Defteri - 3,79 TL
	<<< Çıkar	
	Hesapla	
	Seçilen Ürün	1,49 TL
	Toplam	5,67



CheckedListBox

- Liste kutusunun tüm özellik, metod ve olaylarını alır ve listedeki öğelerin işaret kutusu ile gösterilmesini sağlar.

```
private void button1_Click(object sender, EventArgs e)
{
    int sayi = checkedListBox1.Items.Count;
    for (int i = 0; i < sayi; i++)
    {
        if (checkedListBox1.GetItemChecked(i) == true)
            listBox1.Items.Add(checkedListBox1.Items[i]);
    }
}
```



CheckedListBox

Özellik	Açıklama
CheckedItems	Liste kutusunda işaretlenmiş öğeleri tutar
CheckedIndices	Liste kutusunda işaretlenmiş öğelerin indislerini tutar
CheckOnClick	Liste kutusunda öğeye tıklandığı zaman işaretlenmesini belirler. False ise ilk tıklamada öğe seçilir, ikinci tıklamada seçme kutusu işaretlenir.

Metot	Açıklama
GetItemSelected	Parametre olarak verilen indisteki öğenin seçili olup olmadığını döndürür
SetItemSelected	İlk parametrede verilen indisteki elemanın seçili olup olmadığını, ikinci parametrede verilen Boolean değeri ile belirler



Örnek

- Kategori başına stoktaki ürün toplamının gösterildiği bir uygulamada listelenen kategorileri seçmek için CheckedListBox kullanımı

The image displays two side-by-side screenshots of a Windows application interface, illustrating the use of a CheckedListBox.

Left Screenshot:

- At the top, there is a label "Kategori Adı:" followed by a text input field and an "Ekle" button.
- Below the input field, there is a list box containing four items: "Film", "Müzik", "Bilgisayar", and "Kitap". Each item has an unchecked checkbox to its left.
- To the right of the list box is a label "label2" followed by a large empty rectangular area.

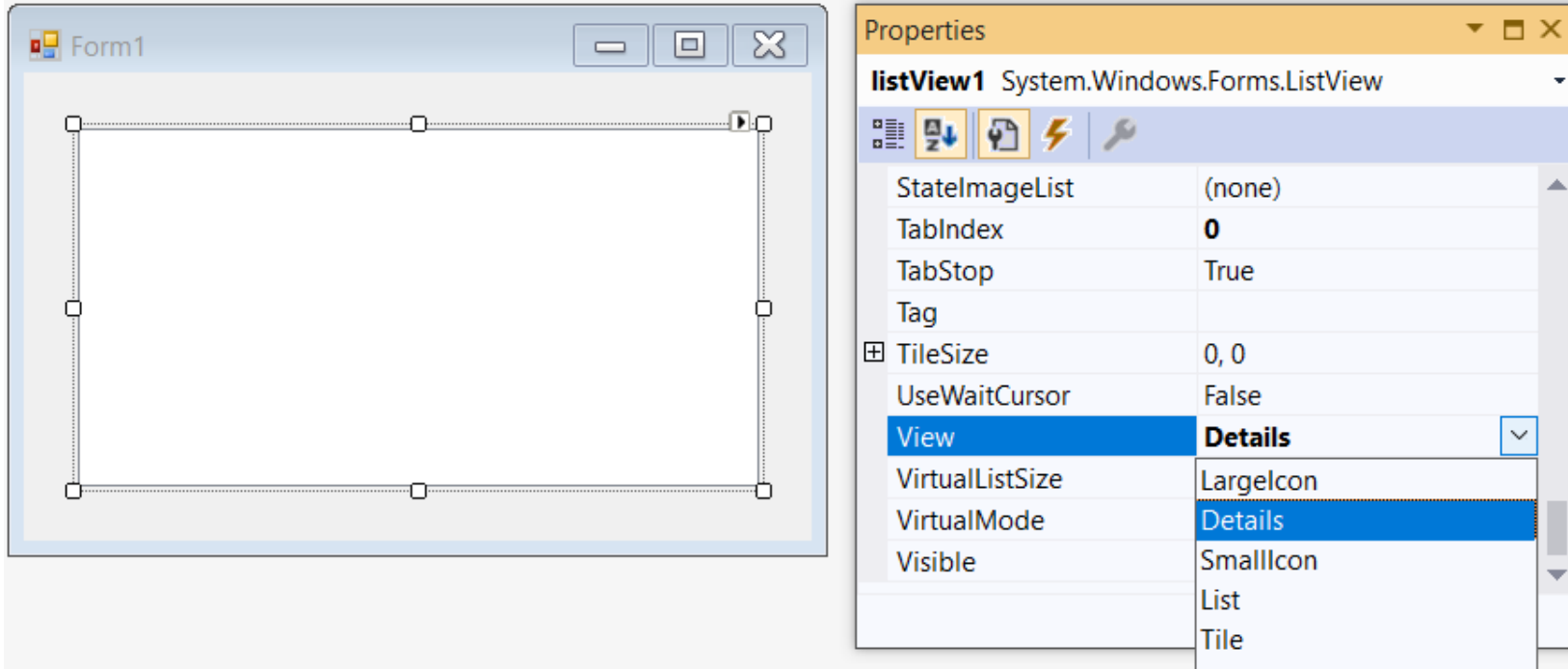
Right Screenshot:

- The layout is identical to the left screenshot, but the list box now shows the first two items, "Film" and "Müzik", with their checkboxes checked. "Müzik" is currently selected (highlighted).
- The large rectangular area to the right now displays the text: "Seilen kategorilerdeki toplam ürün: 1110".



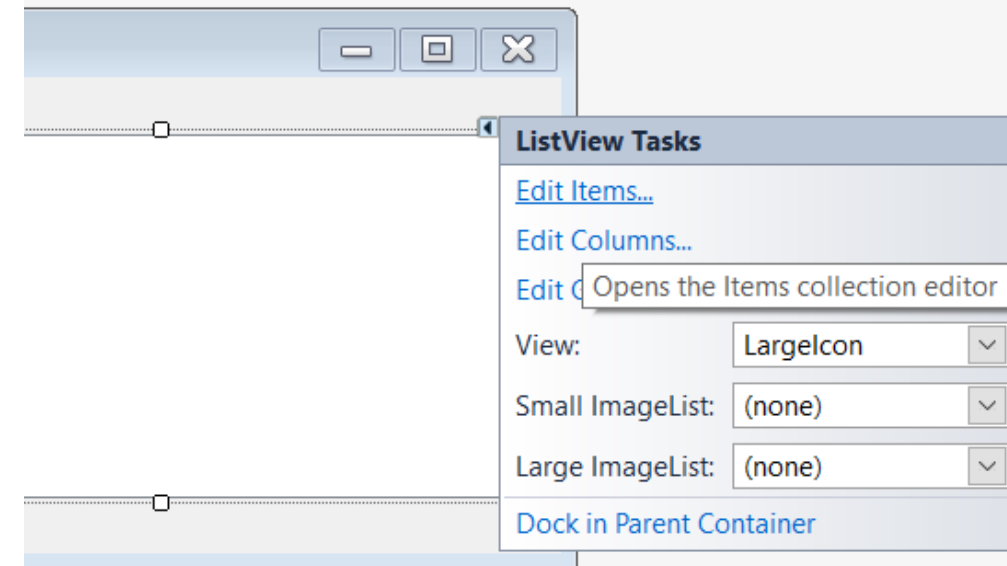
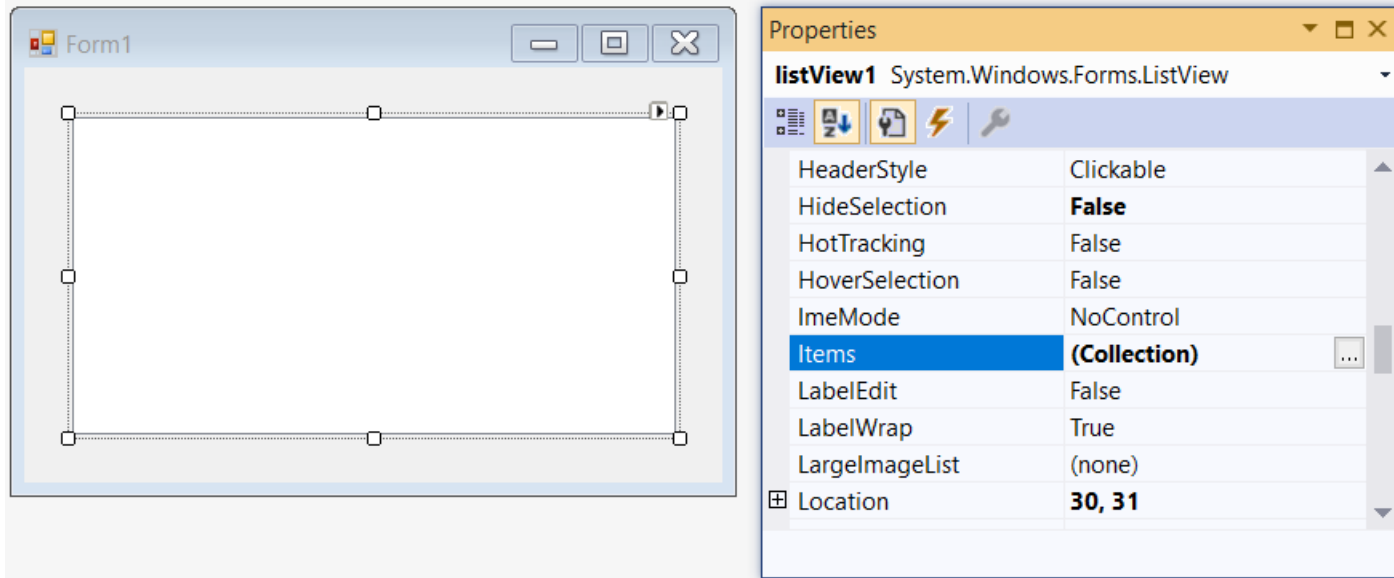
ListView

- ListView gelişmiş bir listeleme kontrolüdür. Listbox'ta olduğu gibi içine elemanlar eklenebilir, her elemana bir resim verilebilir ve listedeki elemanlar farklı biçimlerde listelenebilir.
- Form uygulamasına ListView özelliği eklendikten sonra *Properties* -> *View* kısmından *Details* seçeneğini seçerek oluşturulacak sütunlar görüntülenir.



ListView

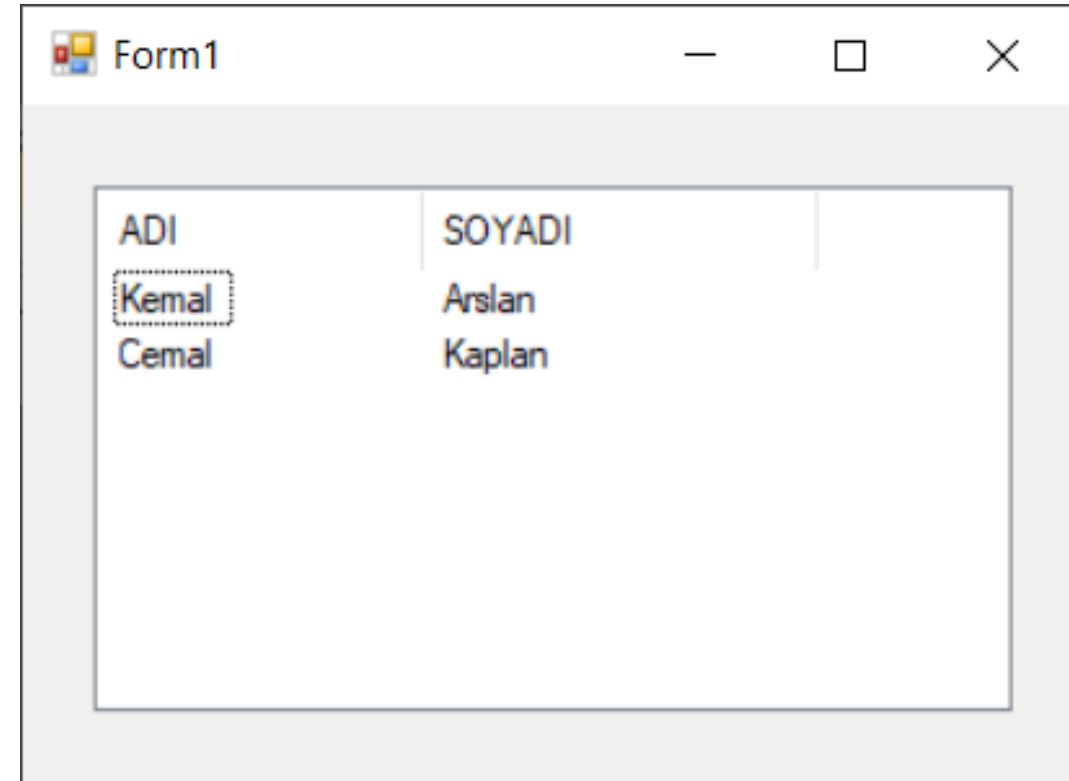
- ListView içerisindeki verileri sütunlar halinde gruplamak için **Properties/Columns (Collection)** seçeneği seçilir ve sütun ifadeleri girilir.
- Sütunlar içerisinde görülecek ifadeler için de **Items (Collection)** seçeneği seçilip ifadeler girilir. Burada dikkat edilmesi gereken şey birinci sütuna gelecek ifadeyi ekledikten sonra bu pencerede iken diğer sütunlara gelecek ifadeler için **SubItems (Collection)** seçeneğini kullanmaktır.



ListView

- Listview'a kodlama yoluyla da ekleme yapılabilir.

```
listView1.Activation = ItemActivation.OneClick;  
// tek tıklamada aktif hale getir.  
listView1.Columns.Add("ADI", -2, HorizontalAlignment.Left);  
// Adı sütun başlığı, -2, Girilen ifadenin tam olarak sığması için gereken ifade, sola hizalama  
listView1.Columns.Add("SOYADI", -2, HorizontalAlignment.Left);  
listView1.Items.Add("Kemal");  
listView1.Items[0].SubItems.Add("Arslan");  
listView1.Items.Add("Cemal");  
listView1.Items[1].SubItems.Add("Kaplan");  
listView1.View = View.Details;
```



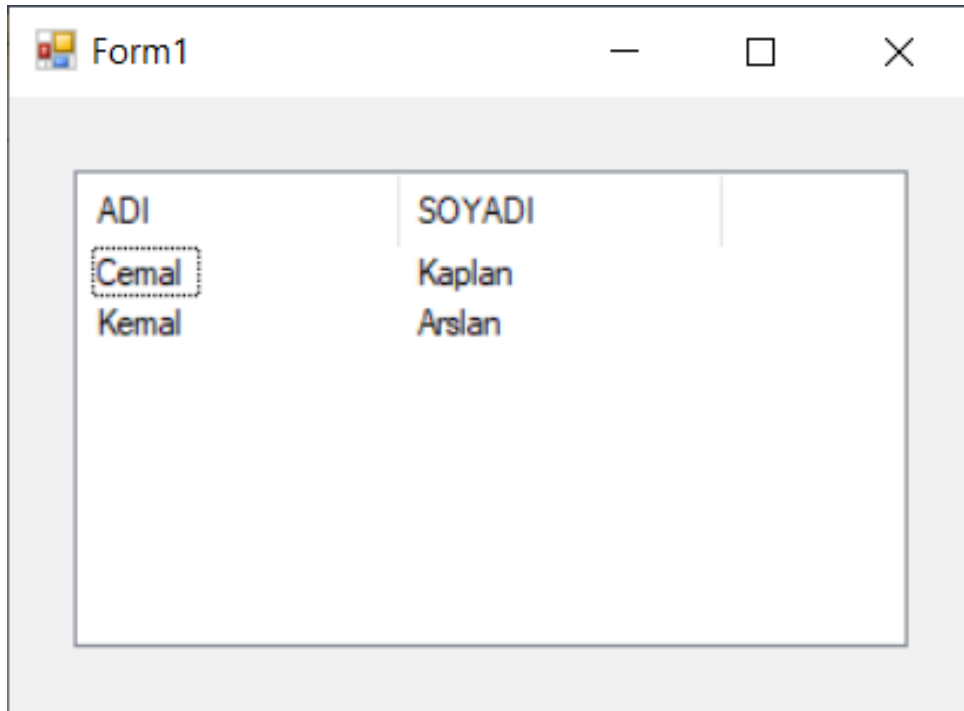
The screenshot shows a Windows application window titled "Form1". Inside the window, there is a ListView control. The ListView has two columns: "ADI" and "SOYADI". It contains two items. The first item has "Kemal" in the "ADI" column and "Arslan" in the "SOYADI" column. The second item has "Cemal" in the "ADI" column and "Kaplan" in the "SOYADI" column. The "Kemal" text is highlighted with a dashed border.

ADI	SOYADI
Kemal	Arslan
Cemal	Kaplan

ListView

- Şayet aşağıdaki kod eklenirse ListView'daki sıralama alfabetik olarak yapılır.

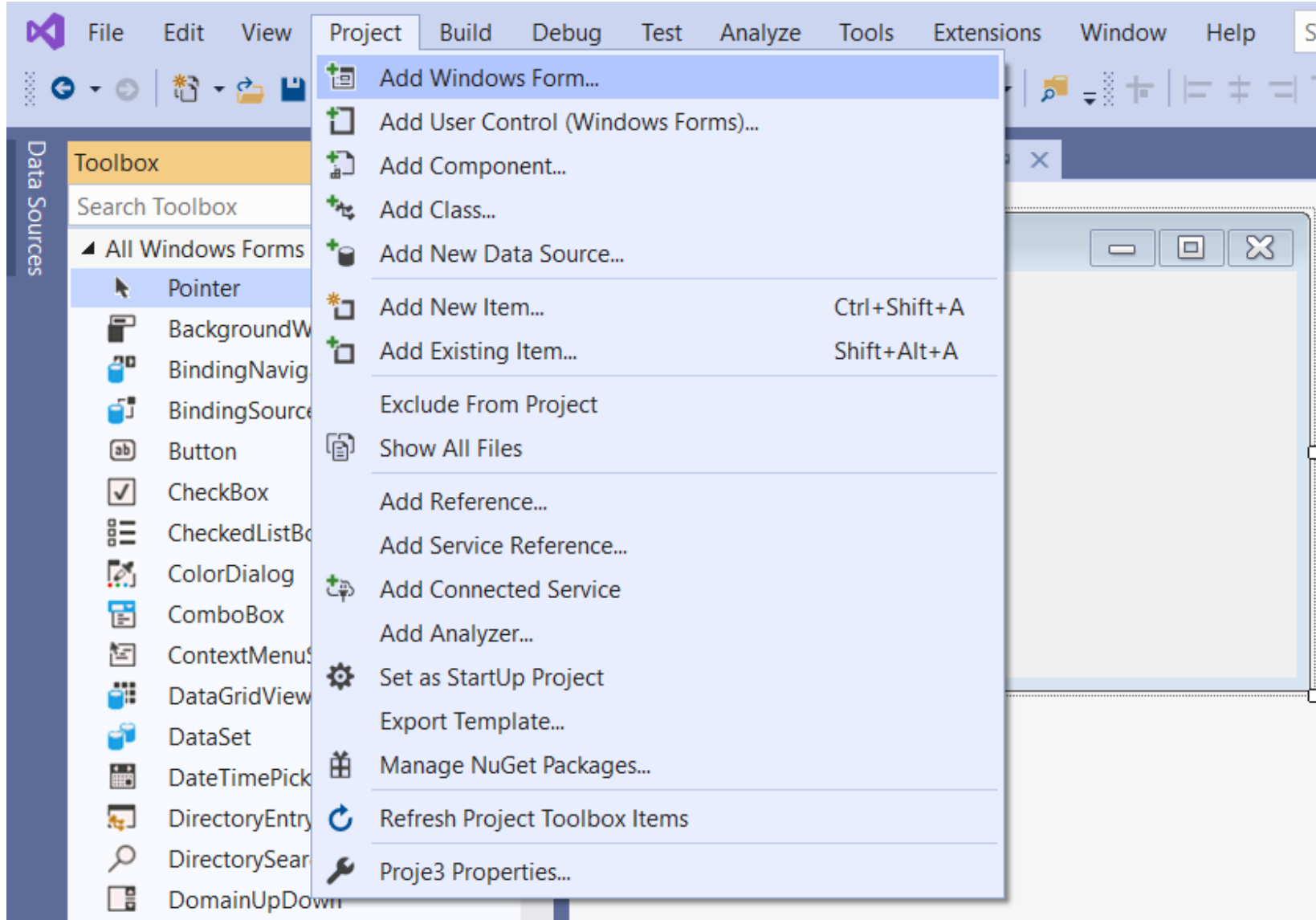
```
listView1.Sorting = SortOrder.Ascending; // Sıralama }
```



ADI	SOYADI
Cemal	Kaplan
Kemal	Arslan



Yeni Form Ekleme



- Project - > Add Windows Forms tıklanarak projeye yeni form eklenebilir.



Formlar Arası Bilgi Aktarımı

The image shows two overlapping Windows-style forms. The background form, titled 'Form1', has a light gray background and a white title bar with standard minimize, maximize, and close buttons. It contains two text input fields: the first is labeled 'Öğr No:' and contains the value '202320404074'; the second is labeled 'Ad-Soyad:' and contains the value 'Kemal Arslan'. Below these fields is a blue button with the text 'İleri'. The foreground form, titled 'Form2', is slightly offset to the right and bottom. It also has a light gray background and a white title bar. It displays the data from Form1 within a white rectangular box. The box has a title 'Alınan Bilgiler' and contains two lines of text: 'Öğrenci No: 202320404074' and 'Ad-Soyad: Kemal Arslan'.

Form1

Öğr No: 202320404074

Ad-Soyad: Kemal Arslan

İleri

Form2

Alınan Bilgiler

Öğrenci No: 202320404074

Ad-Soyad: Kemal Arslan



Formlar Arası Bilgi Aktarımı

Form1

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        String ogrNo = textBox1.Text;
        String adSoyad = textBox2.Text;
        Form2 yeniForm = new Form2(ogrNo, adSoyad);
        yeniForm.Show();
    }
}
```

Form2

```
public Form2()
{
    InitializeComponent();
}

String number, isim;
public Form2(string ogrNo, string adSoyad)
{
    number = ogrNo;
    isim = adSoyad;
    InitializeComponent();
}

private void Form2_Load(object sender, EventArgs e)
{
    label1.Text = "Öğrenci No: " + number;
    label2.Text = "Ad-Soyad: " + isim;
}
```

Kaynakça

- C# Eğitim Kitabı 2. Baskı, Murat Yücedağ, Dikeyksen Yayınları, 2019.
- C# Ders Notları, YMT218 Nesne Tabanlı Programlama, Fırat Üniversitesi (Erişim Tarihi 29 Ekim 2019).
- C# Programlama Ders Notları, Dr. Serkan Dişlitaş, Hitit Üniversitesi (Erişim Tarihi 29 Ekim 2019).

