

# *Kontrol Ekleme ve Özellikleri*

## **TextBox(Metin Kutusu):**

- *Metin kutusu, kullanıcıdan bilgi almak için kullanılır.*
- *TextBox isim verip program sırasında kullanıyoruz. TextBoxt' a yazılan bir değer alınıp bir yerler de kullanılabilir.*
- *Bu textbox program içerisinde kullanılabilir bu yüzden ismini değiştirmek gerekebilir.*
- ***Not:**TextBox tüm özelliklerinin özellik kısmından değiştirebiliyoruz. Üzerinde **sağ** tıklayıp **properties** diyoruz.*
- *Burada textBox'ın tüm özelliklerini değiştirebiliyoruz.*

# *Birkaç önemli olanı bakmak istersek ;*

- **Name:** Program yazarken bu isim üzerinden kodları kullanıyoruz. Mesela “txtAd” ya da “tbAd” gibi kişilerin isimlerini aldığımız metin kutusu gibi, ön ekte kullanmak iyi olur.
- **Font:** Yazı stili
- **Lines:** Çizgiler
- **TextAlign:** Metin sağa sola hizala
- **Size:** Genişliği ya da elle sürükleyerek.
- **Text:** Metin kutusunda ilk yazan/varsayılan değer(Mesela “Adınızı giriniz”)
- **Visible:** Görünürlük(Göster/Gizle)
- **BackColor:** Arka plan rengi.
- **BorderStyle:** Border görünürlüğü tek çift olması

## **LABEL(Etiket):**

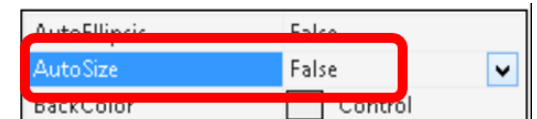
- *Yazılar ve bildirim amaçlı kullanım.*
- *Bir işlemin sonucunu ya da bir yazı gösterebiliriz.*
- *Label kontrolleri, Form üzerinde kullanıcıya bilgi vermek ve denetimler için açıklayıcı metinler sağlamak amacıyla kullanılır.*

# *Label kontrolünün özellikleri*

- **Font** :Yazı tipi özelliklerini,
- **ForeColor** : özelliğinden rengini,
- **Text** : kısmından ekranda görülecek yazıyı(örnek “Birim”),
- **Name**: Bu ilk olarak Label1 ismiyle program tarafından otomatik atanır sonra biz değiştiriyoruz. Label'ın program yazarken (kod tarafında)kullanılacak olan ismini(**lblbirim** gibi) belirtiyor

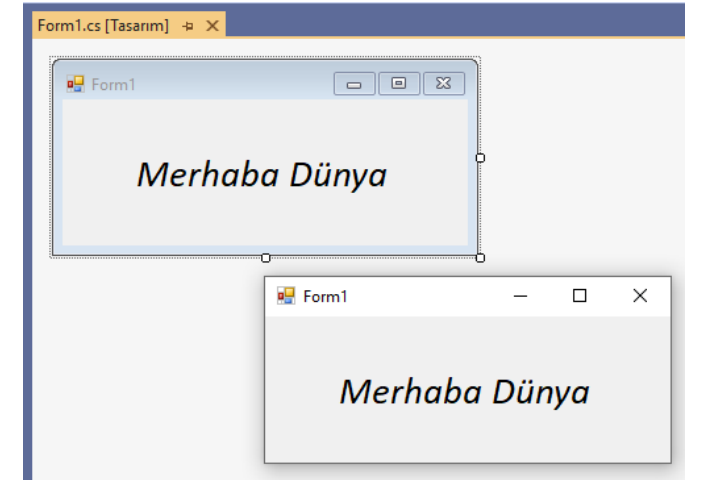
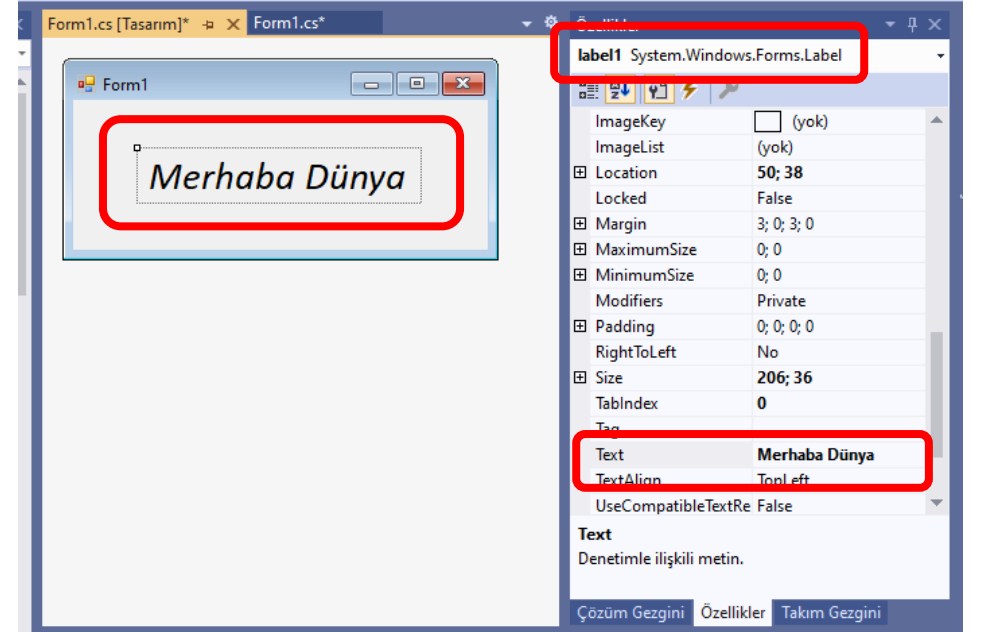
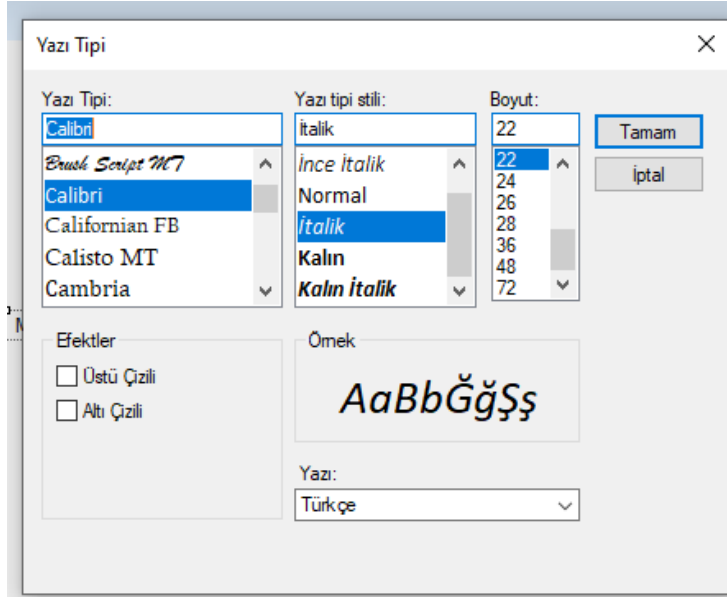
- **TextAlign:** Yazının, etiket üzerindeki pozisyonu belirler. Yazının hizalaması
- **BorderStyle:** Kontrolün kenar stilidir .
- **FixedSingle** değeri, kontrolün kenar çizgilerini gösterir. **Fixed3D** değeri, kenarların üç boyutlu olmasını sağlar.
- **Image** : Etiket üzerinde görüntülenmek istenen resmi tutar.
- **ImageAlign:** Label kontrolünün üzerindeki resmin hizalanması, etiket üzerindeki resmin nerede duracağını belirler.

İmage eklemek istediğimizde mutlaka AutoSize özelliği False yapılmalı



# Örnek : Merhaba Dünya(Label-Merhaba Dünya)

- Bir tane label ekleyelim (Text-Merhaba Dünya)
- Yazı tipi boyutunu değiştirelim



Not: Aynı form üzerinde **aynı isimde**(name,id) birden fazla nesne olmaz.

## Basit bir örnek yapalım; TextChanged

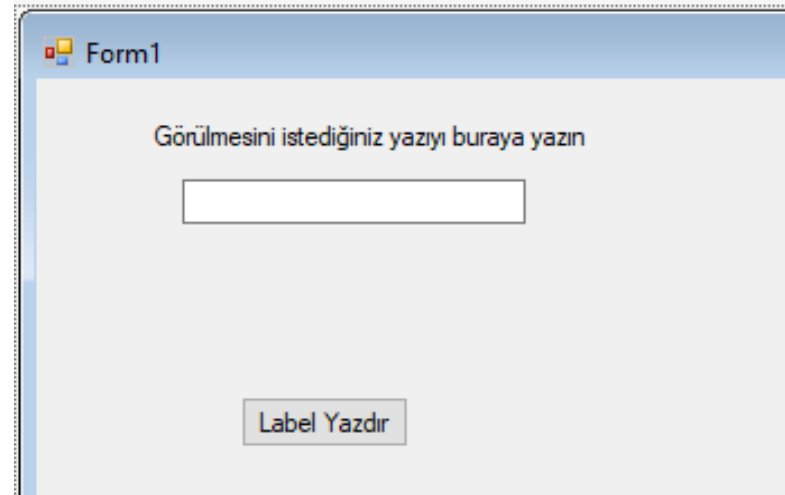
- Bir tane textBox'ımız olsun bu textBox'ın içerisinde metin yazıp sildiğimizde her bir karakter değişiminde(TextChanged) anlık olarak eklenen label'a(lblBilgi) en son halini yazdıran bir örnek.
- TextBox Ekleyelim(name : tbAd)
- Label ekleyelim(name : lblYazi)
- TextBox'a çift tıklayıp TextChanged metoduna gidip alt kısımdaki kodu yazalım
  - lblYazi.Text = tbAd.Text;



## Örneğimize eklemeler yapalım; Buton ekleyelim

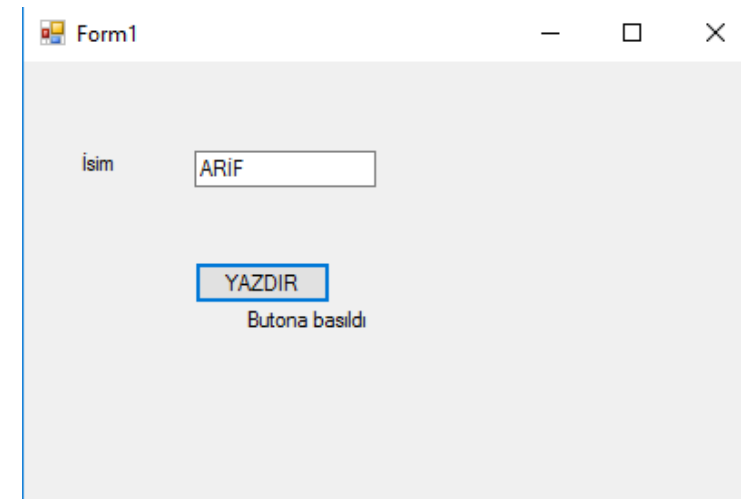
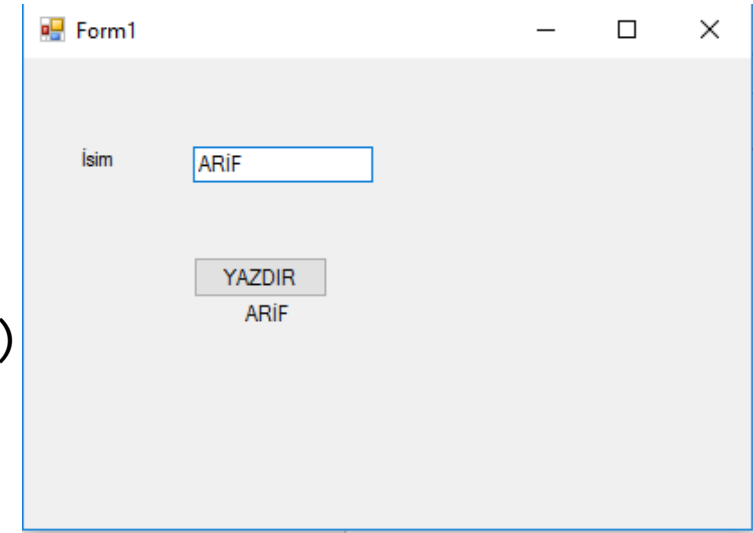
*Bir adet buton ekleyelim butona basılınca textBox içerisindeki bilgileri yazan label da “Butona basıldı” yazsın.*

- Buton ekleyelim name:**btnYazdir** olsun(isimlendirmelerde boşluk yok)
- Butonun varsayılan özelliği `butonClick`’tir. Bunu olaylardan yapabildiğimiz gibi buton üzerine çift tıkladığımızda kod ekranından yazabiliriz.



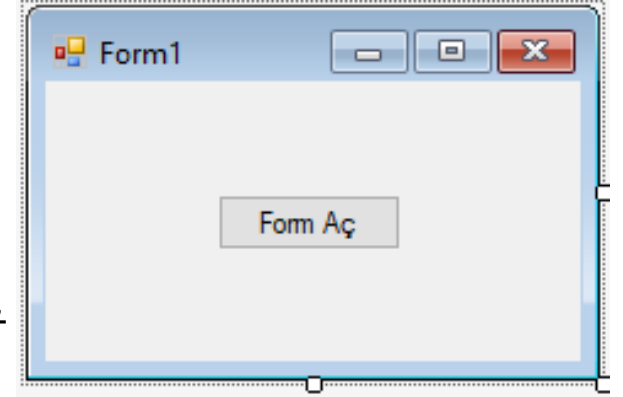
```
private void btnHesapla_Click(object sender, EventArgs e)
{
    lblYazi.Text = "Butona basıldı";
}
```

**Özetleyecek olursak;** Form üzerinde eklenen nesneler ve bunlara uygulanan olaylara kod yazmak istediğimizde, Form1.cs dosyasında otomatik olarak birer **metot** olarak karşımıza çıkmakta. Bizlerde istediğimiz işlemleri hesaplama veri tabanına okuma yazma, yeni bir form açma vb bu alanlara yazabiliriz.



## Örneğimize eklemeler yapalım: Form Açma

Bir form daha ekleyelim butona basılınca başka form açılsın ve merhaba dünya yazsın.



- Projeye MerhabaDunya isminde bir form daha ekledik
- Ana formda bir buton koyalım Bu butona basılınca merhaba dünya formu açılsın
- MerhabaDunya merhaba = new MerhabaDunya();
- merhaba.Show();

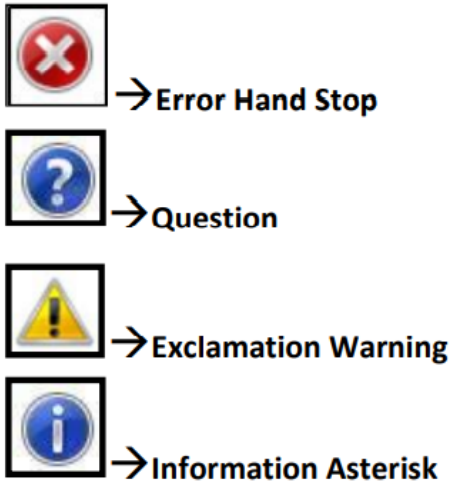
```
private void button1_Click(object sender, EventArgs e)
{
    MerhabaDunya merhaba = new MerhabaDunya();
    merhaba.Show();
}
```

*Burada yeni bir form açmanın iki yolu var;*

- *Show(Arka pencerede hala işlem yapılabilir.)*
- *ShowDialog(Sadece en son açılan pencerede işlem yapılabilir, arka pencereler tıklanamaz)*

# MessageBox

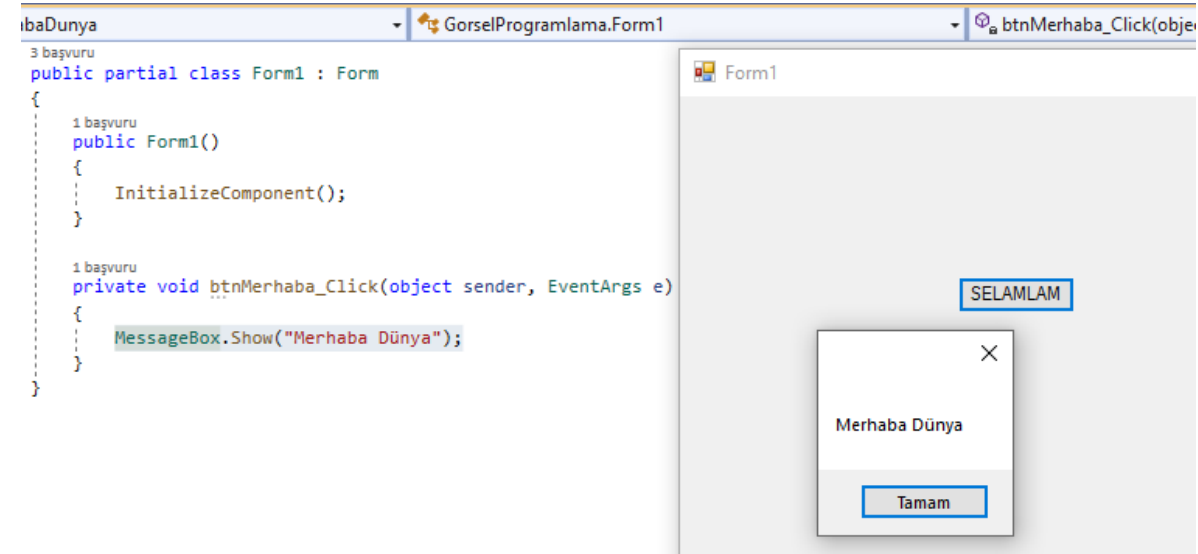
- Kullanıcıya mesaj vermek için Show metodu ile beraber kullanılır.
- Eğer mesaj verilip mesajın durumuna göre Evet, Hayır, Tamam ve İptal gibi butonlardan birine tıklanınca hangisine tıklandığı bilinmek isteniyorsa geri dönüş değeri olan DialogResult özelliğinden yararlanılır



## Örneğimize eklemeler yapalım: MessageBox Açma

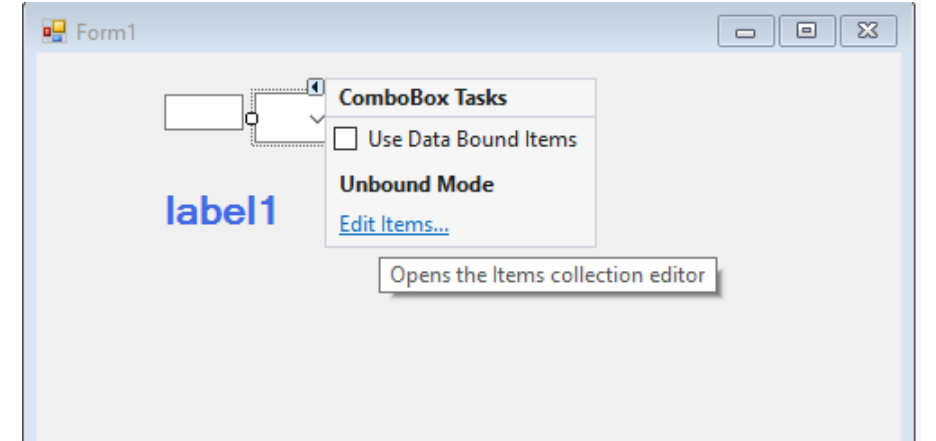
- Butona tıklandığında merhaba dünya yazan bir mesaj kutusu çıksın
- İlk olarak bir buton ekleyelim(name btnMerhaba, text: Selamlama)
- Butona çift tıklayıp gelen kod bloğunun içine alt kısımdaki kodu yazıp çalıştıralım;

`MessageBox.Show("Merhaba Dünya");`

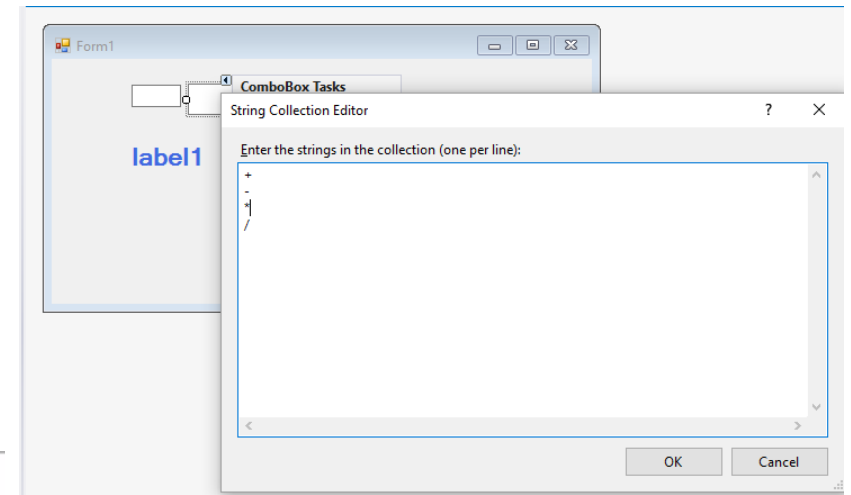
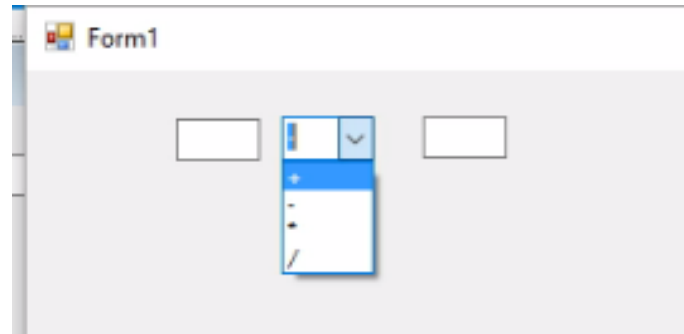


## Örnek : Basit Dört İşlem

- *Girilen iki sayıyı seçilen dört işlemten birine göre hesaplayan programı yapalım.*
- *TextBox 2 tane (name,tbSayi1,tbSayi2)(ctrl+c ve ctrl+V çalışır )*
- *1 adet ComboBox (Yapılacak işlemin türünü seçeceğiz) 1 tane (name, cbIslem)*
- *ComboBox'ın üzerindeki butona tıklayıp Edit Items diyoruz.*
- *Buraya hem dinamik hem de statik olarak değer girebiliriz. Dinamik olarak değer girme programsal olarak o anki verileri çekerek atar statik olarak sabit elle şimdi yapacak olduğumuz gibi değer girilmesidir.*



- *Gelen ekranda sırasıyla + - \* / alt alta yazıyoruz bu yazım şekliyle her bir satır ayrı bir eleman diye gelecektir*





- *Sonucu göstermek için label ekleyelim.*
  - *Label ekleyelim (name lblSonuc)(font özelliklerini değiştirelim daha büyük gözüksün)*
- *Butona tıklanınca hesaplama yapması için buton ekleyelim;*
  - *Buton ekleyelim (text Hesapla, name btnHesapla)*

- *Şimdi butona çift tıklayıp kod yazmamız gerekiyor.*

```
decimal sayi1, sayi2, sonuc=0;  
  
sayi1 = Convert.ToDecimal( tbSayi1.Text);  
  
sayi2 = Convert.ToDecimal(tbSayi2.Text)  
  
string islem;  
  
islem = cbIslem.Text;
```

```
switch (islem)
```

```
{
```

```
    case "+": sonuc = sayi1 + sayi2; break;
```

```
    case "-": sonuc = sayi1 - sayi2; break;
```

```
    case "*": sonuc = sayi1 * sayi2; break;
```

```
    case "/": sonuc = sayi1 / sayi2; break;
```

```
    default: break;
```

```
}
```

```
lblSonuc.Text = "Hesaplanan sonuç:" + sonuc.ToString();//sonuç değerini metine  
dönüştürüyoruz.
```

```
decimal sayi1, sayi2, sonuc = 0;
sayi1 = Convert.ToDecimal(tbSayi1.Text); //textbox tan string olduğundan sayıya dönüştürüyoruz.toint yerine virgüllü olabi
sayi2 = Convert.ToDecimal(tbSayi2.Text);
string islem;
islem = cbIslem.Text; //combobox gelen değeri text özelliği olarak alıyoruz
switch (islem)
{
    case "+": sonuc = sayi1 + sayi2; break;
    case "-": sonuc = sayi1 - sayi2; break;
    case "*": sonuc = sayi1 * sayi2; break;
    case "/": sonuc = sayi1 / sayi2; break;

    default: break;
}
lblSonuc.Text = "Hesaplanan sonuç:" + sonuc.ToString(); //sonuç değerini metine dönüştürüyoruz.
```

The screenshot shows a Windows application window titled "Form1". Inside the window, there are two text input fields. The first field contains the number "23" and the second field contains the number "45". Between these two fields is a dropdown menu currently showing the multiplication symbol "\*". To the right of the second text field is a button with the text "HESAPLA". Below these input elements, the text "Hesaplanan sonuç:1035" is displayed, indicating the result of the calculation (23 \* 45).

*Not: Biz bu örnekte ComboBox kullanımını biz statik olarak yani elle değer vererek yaptık bunu kod ile dinamik olarak da yapabiliriz. Yani veri tabanında var olan değerleri dinamik olarak atayıp gelen değerlere göre açılan bir tasarımda yapabiliriz.*

## Ekleme yapalım. ComboBox' a otomatik değer atayalım

- 1800'den 2020'e kadar olan yılların ComboBox'a yazılması:
- *ComboBox ekleyelim (name cbYillar)*
- *Form yüklenirken comboBox'ın değerleri ile gelmesi gerektiği için FormLoad olayının altına yazıyoruz.*
- *Formun üzerinde çift tıklıyoruz*

```
for (int yıl = 1800; yıl <= 2020; yıl++)  
  
    {  
  
        cbYillar.Items.Add(yıl) ;  
  
    }
```

The screenshot shows a Windows Form titled "Form1". It contains three text boxes labeled "SAYI 1", "İŞLEM TÜRÜ", and "SAYI 2". Below "İŞLEM TÜRÜ" is a dropdown menu. A "HESAPLA" button is located below the dropdown. A ComboBox is open, displaying a list of years from 1800 to 1829. The form has a light gray background and a standard Windows window border.

## Ekleme yapalım : MessageBox ekleyelim

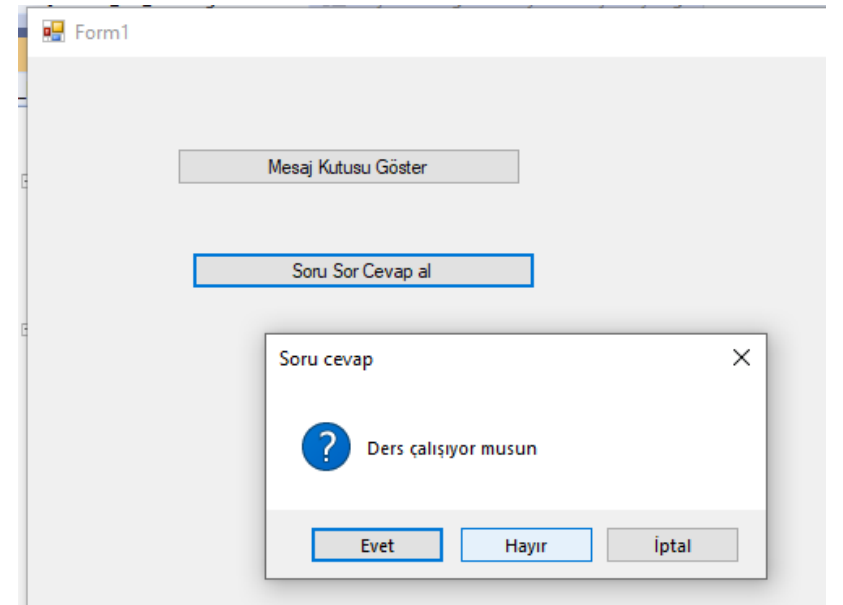
*Form uygulamalarında kullanıcılara soru sorduğumuz bilgilendirme yaptığımız yani kısaca kullanıcı ile iletişime geçtiğimizde kullandığımız MessageBox kullanımına bakalım.*

- Bir adet buton ekleyip bu butona basılınca mesaj kutusu çıkıp kullanıcıya bir mesaj verelim ok deyip mesaj kutusu kapansın.*
- MesajBox kullanımının kalıbı;*

**MessageBox.Show("Bilgi mesajı", "Başlık", Buton türü, İkon);**

- *Bir tane buton ekleyelim (text: Mesaj Kutusu Göster name:Mesaj: btnMesaj)*
- *Butona çift tıklayıp kodları yazalım;*

```
MessageBox.Show("Ders çalışıyor musun", "Soru cevap",  
MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);
```



## EKLEME YAPALIM : DialogResult

- *Kullanıcının tıkladığı butona yani bize verdiği cevabı bir değer olarak alıp işlememizi sağlıyor.*

1 başvuru

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

```
{
```

```
    DialogResult donus= MessageBox.Show("Ders çalışıyor musun", "Soru cevap", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);
```





```
private void btnMesaj_Click(object sender, EventArgs e)
{
    DialogResult donus=MessageBox.Show("Ders çalışıyor musun",
    "Soru cevap", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);
    if (donus == DialogResult.Yes)
    {
    }
    else if (donus == DialogResult.No)
    {
    }
    else if (donus==DialogResult.Cancel)
    {
    }
}
```

## Not-Ekleme:

- Form çalıştığında ekranın orta noktasında açılması için;
- Form özelliğinde “**StartPosition**” ayarını “**CenterScreen**” yapıyoruz

