

# *1. GİRİŞ*

- **Form** uygulamaları Visual Studio.NET üzerinden dersi işleyeceğiz.
- Windows uygulamalarında kullanılan düğmeler (button), listeler (listbox), metin alanları (textbox), etiketler (label) gibi nesneler birer kontroldür. (.NET) Framework'de kontroller `System.Windows.Controls` kütüphanesinde yer alır.
- Form Application sayesinde ekran üzerindeki görsel argümanları kullanarak komut veriyoruz.
- Form uygulamalarında da geçen dönem console uygulamalarında gördüğümüz sınıf uygulamaları, sınıflardan nesne türetme, döngüler vb. tüm durumlar geçerli.
- Visual Studio bize Windows form üzerinden hazır form nesneleri sunmakta. Bu hazır nesnelerin bir takım özelliklerini kullanarak programlar yazabiliyoruz. Hazır olaylar sunarak bu olayların üzerinden program yazabiliyoruz.

## ***Görsel programlamada Visual Studio.NET platformunun tercih edilmesini sağlayan bazı özellikler şunlardır:***

- *Esnek, kullanıcı tercihlerine göre özelleştirilebilir editöre sahip olması*
- *Tümleşik bir derleyici ve hata ayıklayıcıya sahip olması*
- *Kod dosyalarının hiyerarşik bir şekilde görülebilmesine imkân vermesi*
- *Değişik programlama dillerini desteklemesi*
- *Programcılar için çeşitli hazır araçlar sunması*

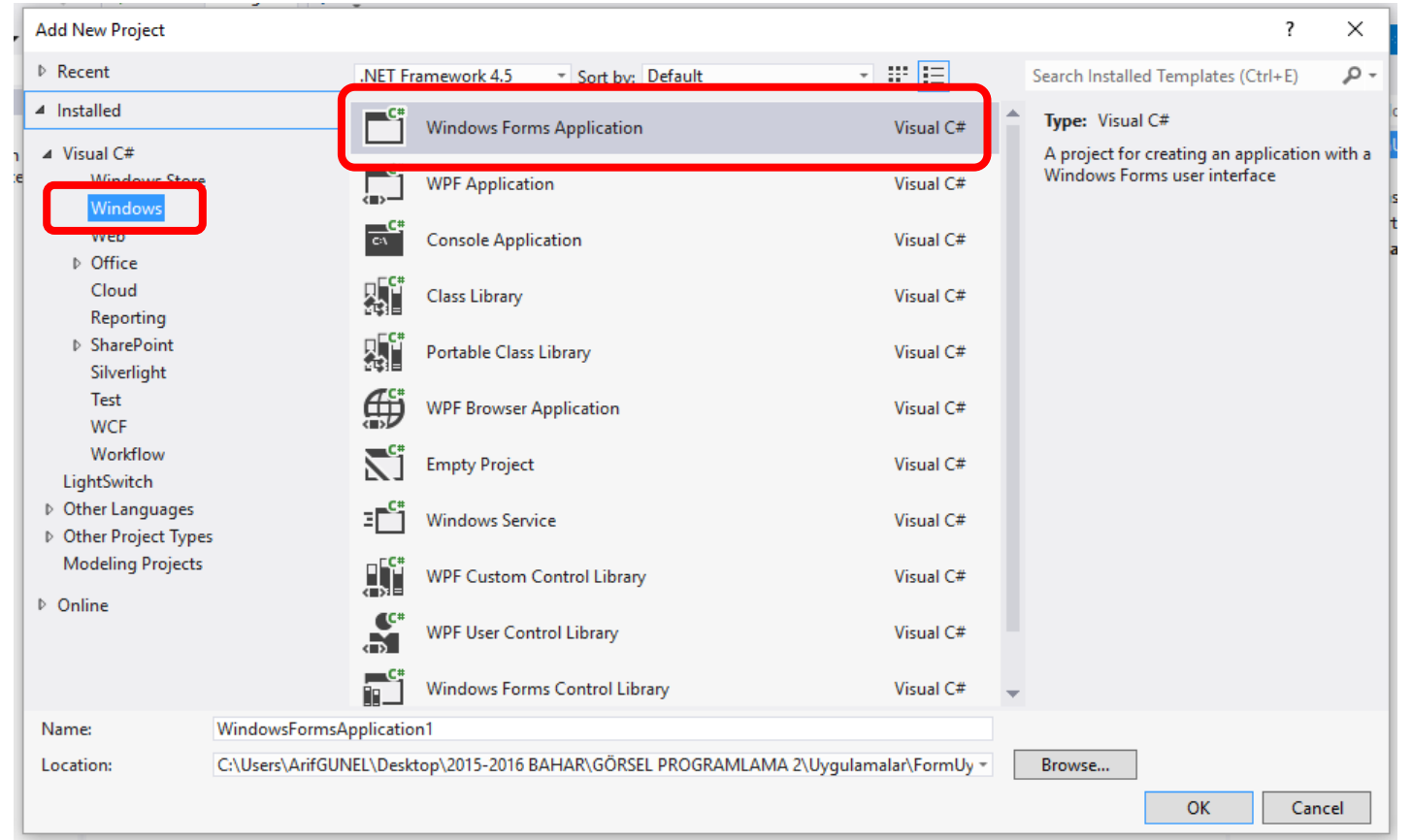
## ***Windows Formlar ile neler yapılabilir;***

- *Windows uygulamalar üzerine exe oluşturup setup uygulamalar yapabildiğimiz*
- *Uygulamayı clint lara kurup çalışmasını sağlayabildiğimiz masa üstü uygulamalar*

*Bir tane proje dosyası açıp üzerine konuşalım.*

## Proje dosyasının açılması:

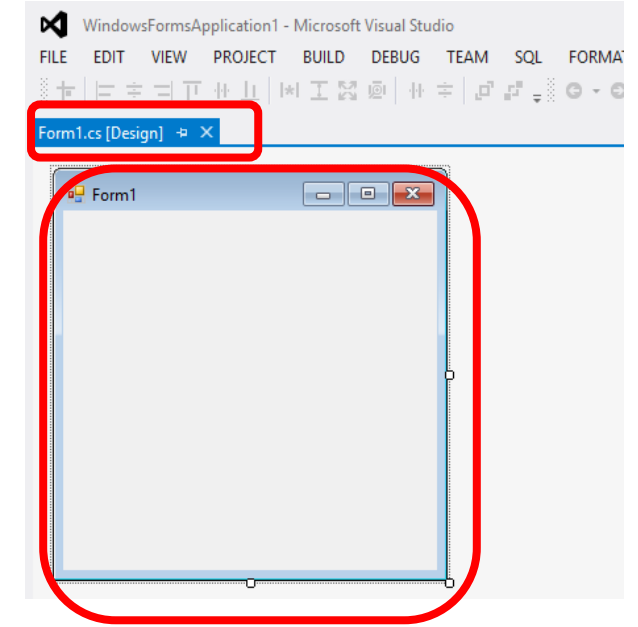
- İlk olarak **Windows** üzerinden **Windows Forms Application** seiyoruz.



## Windows Forms Application Dosyasında Gelen Sayfalar:

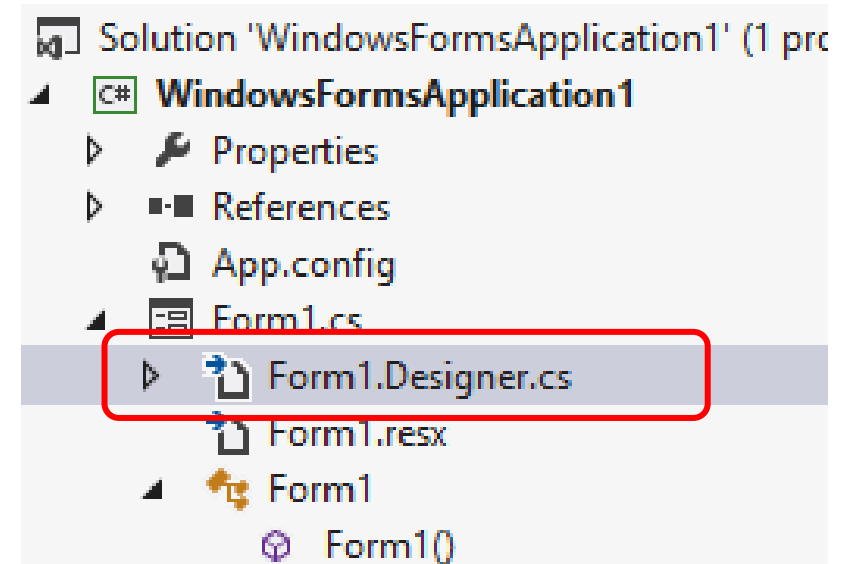
### Desing (dizayn) :

- Windows Form uygulamalarında ilk olarak desing (dizayn) görünümünde **Form** penceresi gelir.
- Gelen pencere Windows tabanlı tüm programlarda standart olduğu gibi simge durumuna küçültme, tam ekran ve kapatma butonları gelir.
- Bu gelen Windows formudur web formu değildir. Bu Windows form penceresine istediğimiz elemanları yerleştirebiliriz.

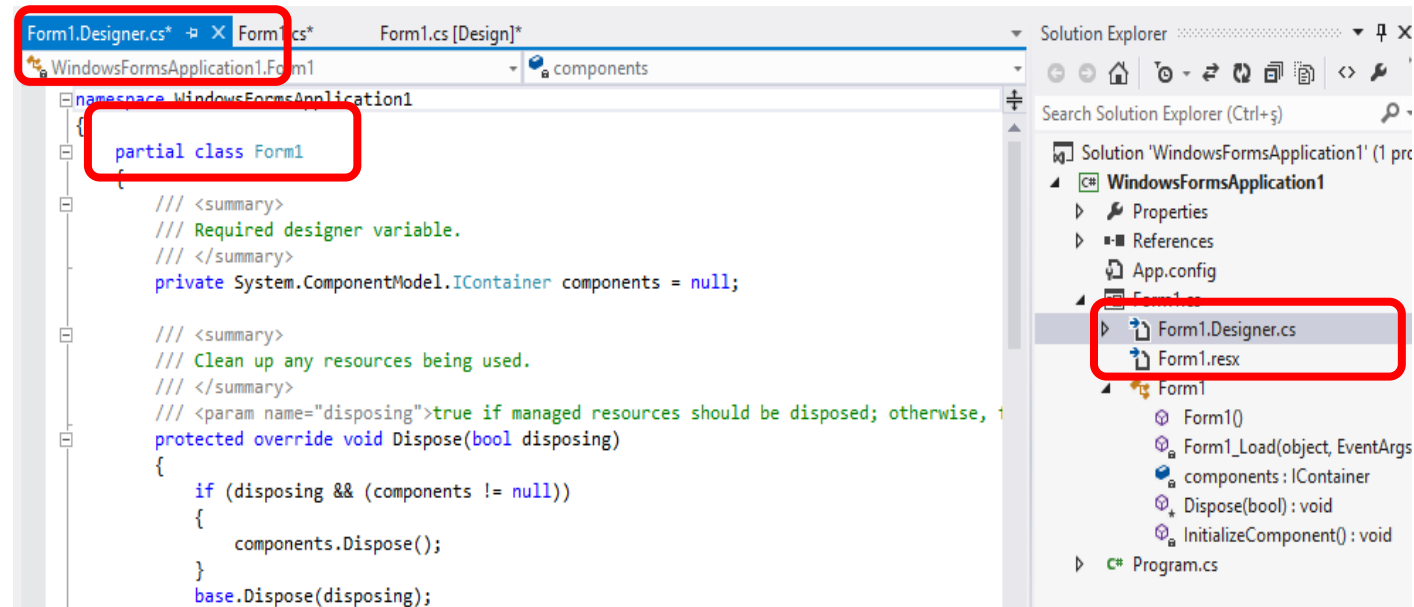


## ***Designer.cs(Form'un Kod Dosyası):***

- *Designer.cs dosyası visual studio'nun arka tarafta otomatik olarak oluşturduğu **kod dosyası**. Bu kısım ile çok fazla işimiz yok.*
- *Bir buton eklendiğinde, formun rengi değiştiğinde vb özellikleri arka tarafta bu designer.cs dosyasıda tutuyor.*
- *Kısaca;*
  - *Tasarım özelliklerinin tutulduğu kısım.*

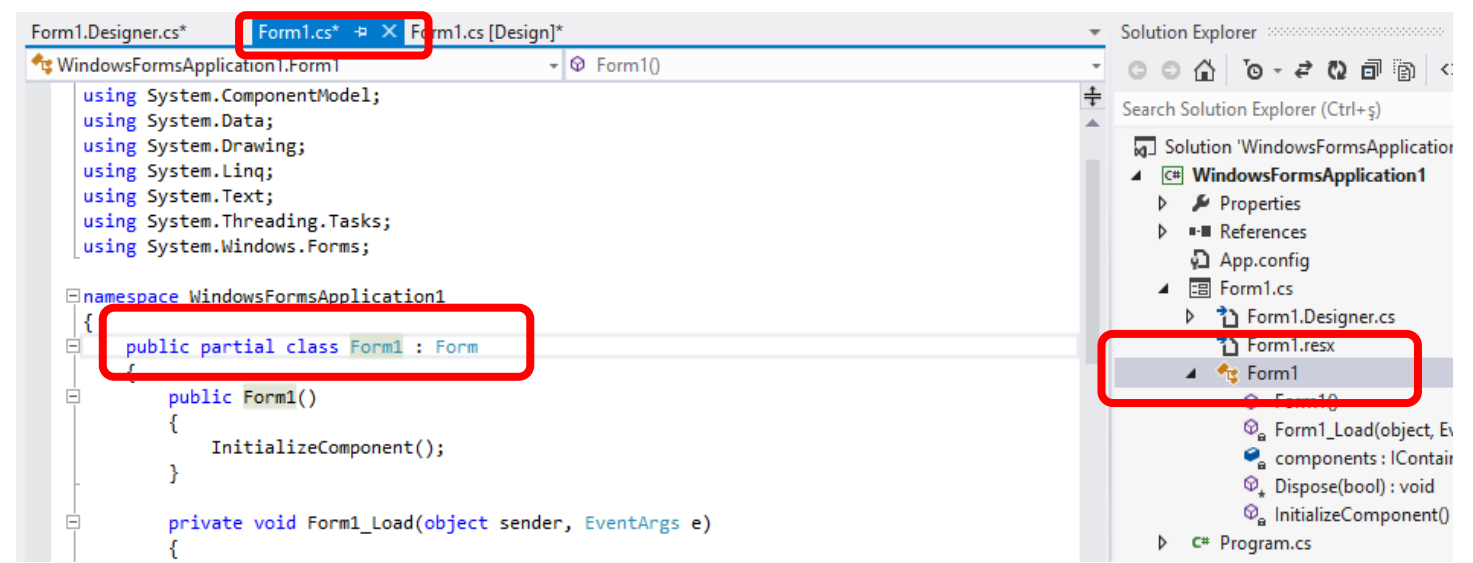


- *Form 1 Partial class'ın altında bu özellikler tutuluyor.*
- *Geçen dönem private, ve public kelimelerini kullanıyorduk.*
- *Partial class olmasının nedeni tek bir isim uzayının altında fakat birden fazla .cs dosyasında kodları parçalı olarak bulunan sınıf tanımlamalarında kullanılıyor. Yani **farklı cs dosyalarında** parçaları olan bir class tanımlamak istersek **Partial** anahtar kelimesini kullanıyoruz.(parçalı class diyoruz.)*



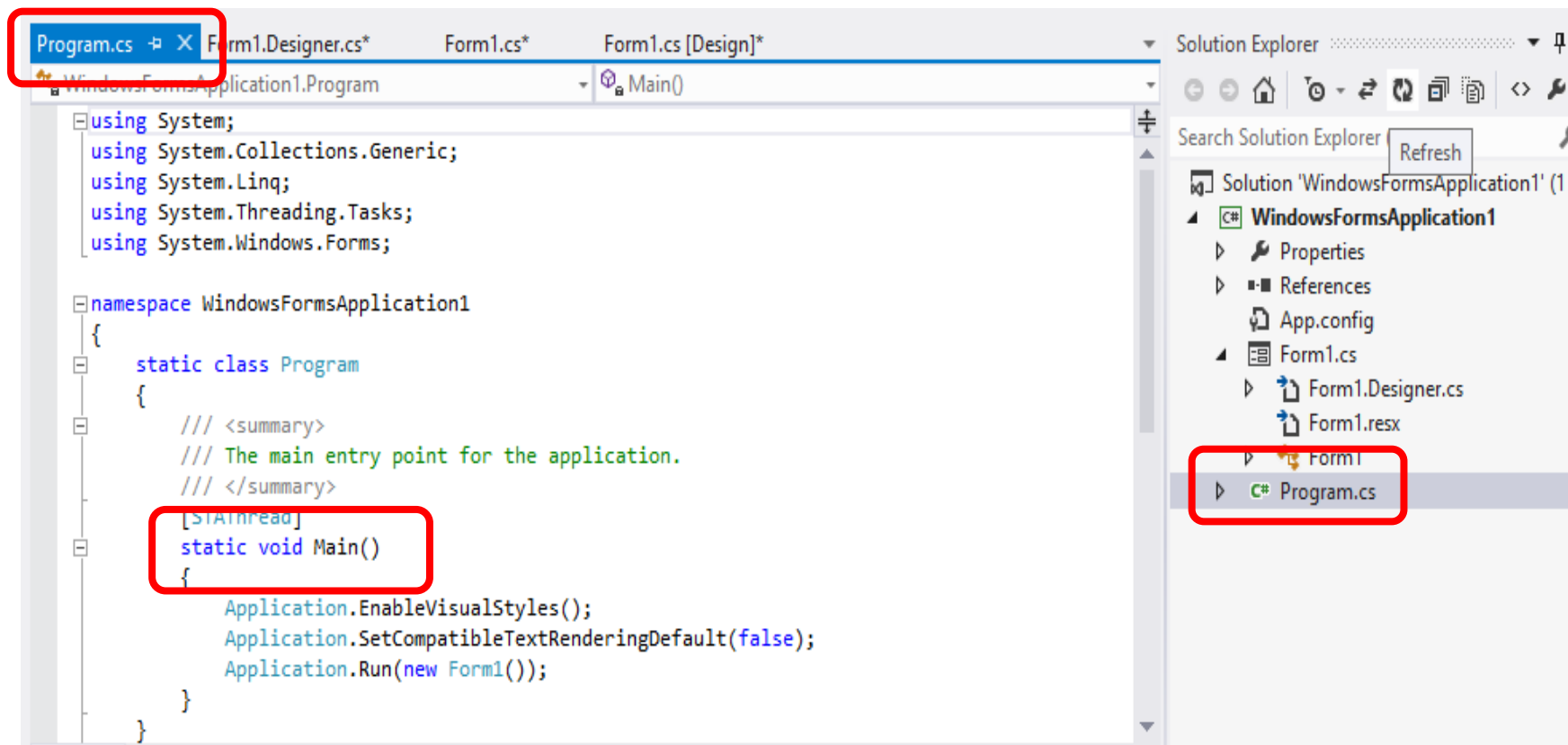


## Form.cs:



- *Bizim kodlarımızı ve istediğimiz değişiklik ve olayları yazdığımız kısımdır. Yani dikkat edilirse farklı bir parçalı(partial) class yapısı var .*
- *İlk çalışan metot olan Public Form1'in altındaki InitializeComponent(); kısmı form üzerindeki tüm kontrollerin özellikleri ile beraber getirilmesini sağlıyor.*
- *Form üzerine eklediğimiz buton vb. nesneler ve elemanlara eklediğimiz olaylara yazdığımız kodların bulunduğu kısımdır.*
- *Form1 yapısı ana **FORM** yapısı üzerinden **miras alma** işlemi gerçekleşiyor.*

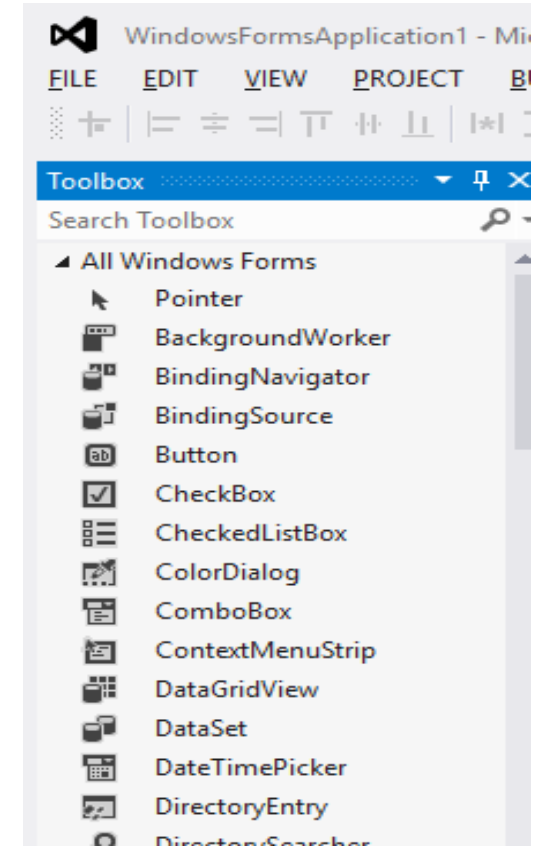
## Program.cs



- *Program.cs* dosyasında daha önceden söylediğimiz gibi **otomatik** olarak kendisinin oluşturduğu **Main metoduna sahip**, her programın bir adet main metodu olmak zorunda.
- Bu kısımda bilinmesi gereken en önemli kısım program ilk çalıştığında *program.cs* dosyasını arar ve bulunca içindeki main metodu bakılarak **ilk olarak ekranda hangi form ara yüzünün** görüleceğinin belirlenmesidir.  
(Application.Run(new Form1()) )

## Toolbox(Araç kutusu):

- Formun üzerine yerleştireceğimiz tüm hazır nesneler (argümanlar-kontroller) mevcut.(Buton,textbox,drop).
- ToolBox panelinde “All Windows Form” gurubu içerisinde 68 kontrol nesnesi vardır.
- Bunların büyük bir kısmı Windows Form projelerinde bazıları ise Wep, Silverlight ve Cloud gibi diğer proje tiplerinde kullanılır.



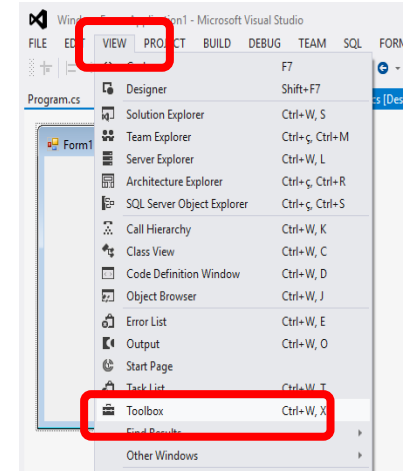
Not: Silverlight,

Microsoft .NET tabanlı gelişmiş medya deneyimi ve web için birçok seçeneğe sahip olan etkileşimli, birbirleriyle entegre çalışabilen, gelişime açık uygulamalar hazırlamaya olanak veren tarayıcılar arası bir multimedia eklentisidir. Silverlight yeni zenginlikler, güvenlik ve çapraz platform deneyimi sunar. Kurumların tasarım ihtiyaçlarını karşılar. Bir başka tanımlamayla web, masaüstü ve mobil uygulamalar için ilgi çekici, interaktif kullanıcı deneyimleri oluşturmak için online ve offline geliştirme platformu sunan **Microsoft** arka planlı bir teknolojidir.

### Toolbox gözükmiyorsa;

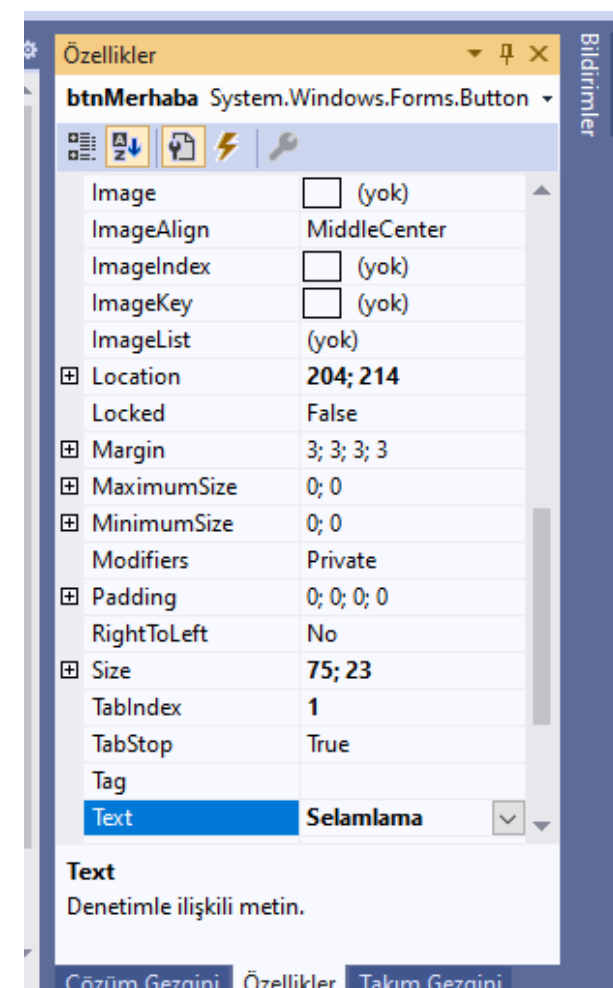
Wiew(Görünüm) sekmesinden  
Toolbox seçiliyor.

ToolBox panelini görüntülemek  
için **CTRL+ALT+X** tuş  
kombinasyonu kullanılır



## Properties(özellikler)

- *Form üzerine eklediğimiz her nesnenin ayarlarını yaptığımız kısım.*
- *Bu kısım nesnenin boyutunu, rengini, kod yazarken ulaşmamızı sağlayan name kısmını vb bir çok özelliği değiştirdiğimiz kısım.*
- **Name:** *Kontrolle erişme için kullandığımız adı.*
- **Text:** *Üzerindeki yazıyı beliriyoruz*
- **Size:** *Ölçülerini belirliyoruz*



## Events (Olaylar):

- *Nesneye ait olayları yazmak istediğimizde.*
- *Burada seçilen olaylara kod yazıyoruz.*
- **Click:** *Tek tıklandığında*
- **DoubleClick:** *Mouse çift tıklandığında*
- **MouseClick:** *TextBox nesnesine tıklandığında*
- **Mouse** *kısımında fare ile üzerine geldiğinde ayrıldığında neler olacağı*
- *Bu kısımda her nesnenin kullanım amacına uygun olarak varsayılan olayı mevcuttur. Bu metot yapısında otomatik olarak oluşturuluyor.*
- *Bu kısımda textbox'ın üzerine çift tıklandığında Form1.cs dosyasında; **Metot oluşturuyor***

