

BİL 201

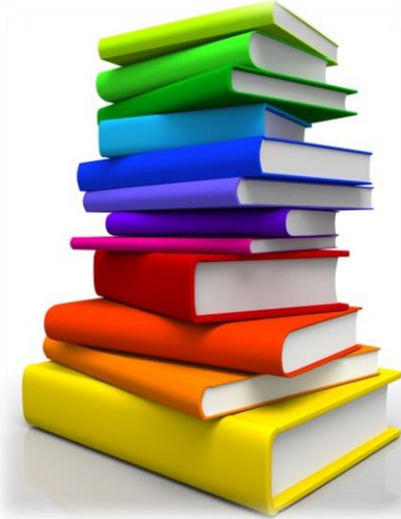
NESNE YÖNELİMLİ PROGRAMLAMA (NYP)

DERS #1.1

Öğretim Üyesi: Doç. Dr. Deniz Kılınç

BÖLÜM 1 - GUI Nesneleri ve Visual Studio IDE Kullanımı

- Bu bölümde aşağıdaki konular anlatılacaktır
 - Programlama Dilleri
 - .Net Uygulama Geliştirme Platformu
 - Visual Studio IDE ile Form Yaratma
 - Toolbox Kullanarak Buton ekleme
 - Form yaratma
 - Forma Label ve TextBox ekleme
 - Adlandırma Yöntemleri ve Düzenleme



Programlama Dilleri

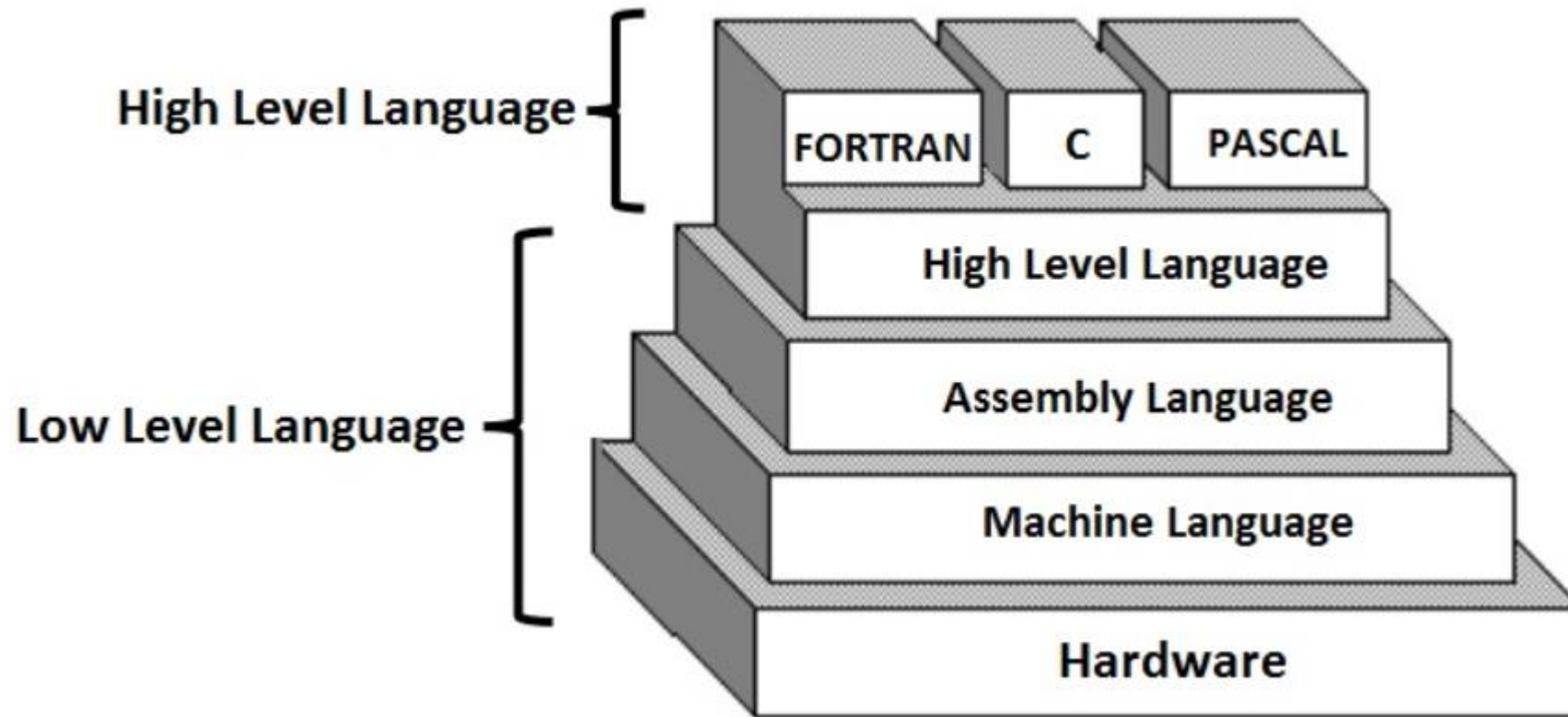
Programlama dilini gözünüzde büyötmeyin !



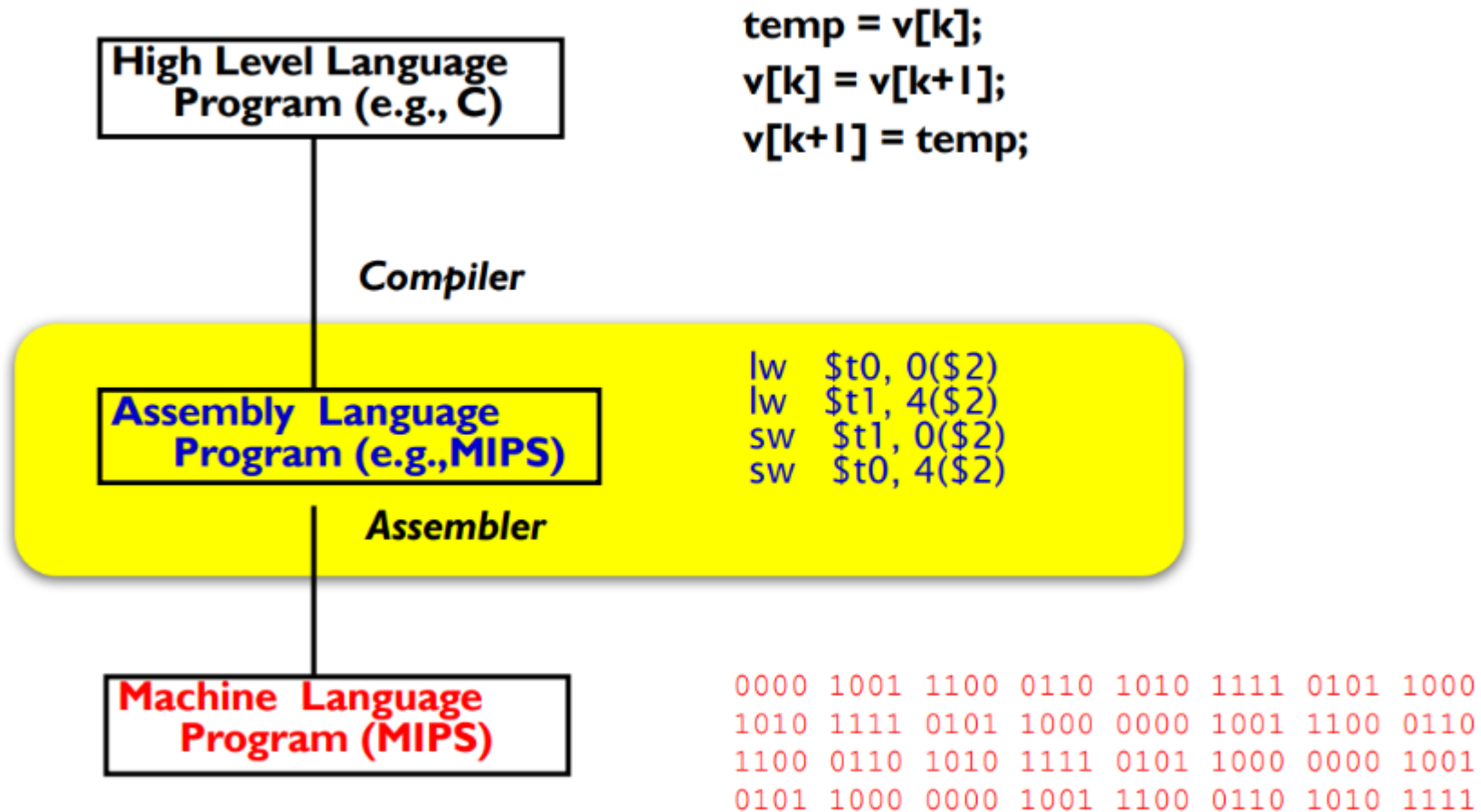
Programlama Dilleri

- Bir **programlama dili**, bir problemin çözümünün bilgisayardaki gerçekleştirimini ifade etmek amacıyla programlar oluşturulması için kullanılan bir dildir.
- **Programlama Dilleri Değerlendirme Kriterleri**
 - **Okunabilirlik**: Bir programlama dilinin değerlendirilmesinde en önemli kriterlerden birisi, programların okunabilme ve **anlaşılabilme kolaylığıdır**. Programlama dillerinin okunabilir olmaları, programlarda hata olasılığını azaltır ve programların bakımını kolaylaştırır. Okunabilirlik, bir programlama dilinin genelindeki **basitlik**, **sadelik** ve **ortogonalite** tarafından etkilenir. *Karmaşık bir sözdizim, bir programın yazımı sırasında kısa yollar sağlayabilir ancak programın daha sonra değiştirilmek amacıyla okunmasını ve anlaşılmasını zorlaştırır.*
 - **Yazılabilirlik**: Bir programlama dilinde program yazma kolaylığını belirleyen en önemli etkenlerden birincisi, programlama dilinin **sözdizimidir**. Buna ek olarak, programlama dilinin **soyutlama** yeteneği, dilin yazılabilirliğini önemli ölçüde etkilemektedir.
 - **Güvenilirlik**: Programlama dillerinde güvenilirlik, çeşitli faktörler tarafından belirlenir. Örneğin; dilde bulunan **tip denetimi** ve **istisnai durum işleme** verilebilir.

Programlama Dilleri (devam...)



Programlama Dilleri (devam...)



Programlama Paradigmaları

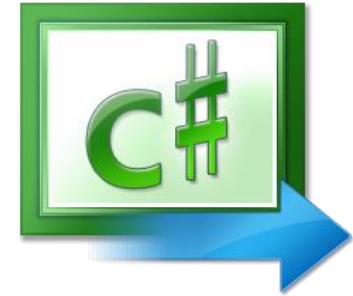
- Belirli bir paradigmayı destekleyen dillere paradigma-yönelik diller denir. Örneğin, Smalltalk ve Eiffel gibi diller nesneye yönelik programlama paradigmasını desteklerler. Benzer şekilde, FORTRAN ve Pascal, imperative paradigmayı desteklerler. Bazı diller paradigma bağımsız olup, birden çok paradigmayı destekleyebilirler. Örneğin, C++ hem imperative hem de nesneye yönelik programların geliştirilmesini destekler.
 - **Imperative Paradigma:** Bu dillerde bir program, **bir dizi deyimden oluşur** ve her deyimin çalışması, bellekteki bir veya daha fazla yerleşimin değerinin değiştirilmesine neden olur. Örneğin, iki değişkenin toplanması ve üçüncü bir değişkenin elde edilmesi, bu değerlerin birleştirilerek sonucun yeni bir yerleşimde saklanması olarak gösterilebilir (**C, Pascal, Ada**).
 - **Nesneye Yönelik Paradigma:** Nesnelerin sınıf ve alt sınıflara gruplanması, nesneye yönelik programlamanın temel noktasıdır (**C++, C#, Java**).
 - **Fonksiyonel Paradigma:** Fonksiyonel dillerde, veriler ve sonucu elde etmek için veriye uygulanacak fonksiyonel dönüşümler, paradigmanın temelini oluşturur (**Lisp-AI**).
 - **Mantık Paradigması:** Mantık programlama paradigmasında programlama, bir işin nasıl yapılacağını belirtilmesi yerine, ne yapılması istendiğinin belirtilmesi olarak görülür (**Prolog-AI**).

C# Programlama Dili

- **Microsoft** tarafından geliştirilmiş olan bir programlama dilidir.
- C++ ve Java dillerine **syntax** olarak oldukça benzer, ancak C#'ın bu dillere benzerliği yanında farkları da vardır.

Örneğin:

- C#, C++'dan farklı olarak % 100 **nesneye yönelik paradigmayı** sahiptir.
- Java'dan farklı olarak ise C#'ta **işaretçi (pointer)** kullanılabilir. Bu sayede eski yazılım bileşenleriyle uyumlu bir şekilde çalışılabilir.



C# ve Java Farkları

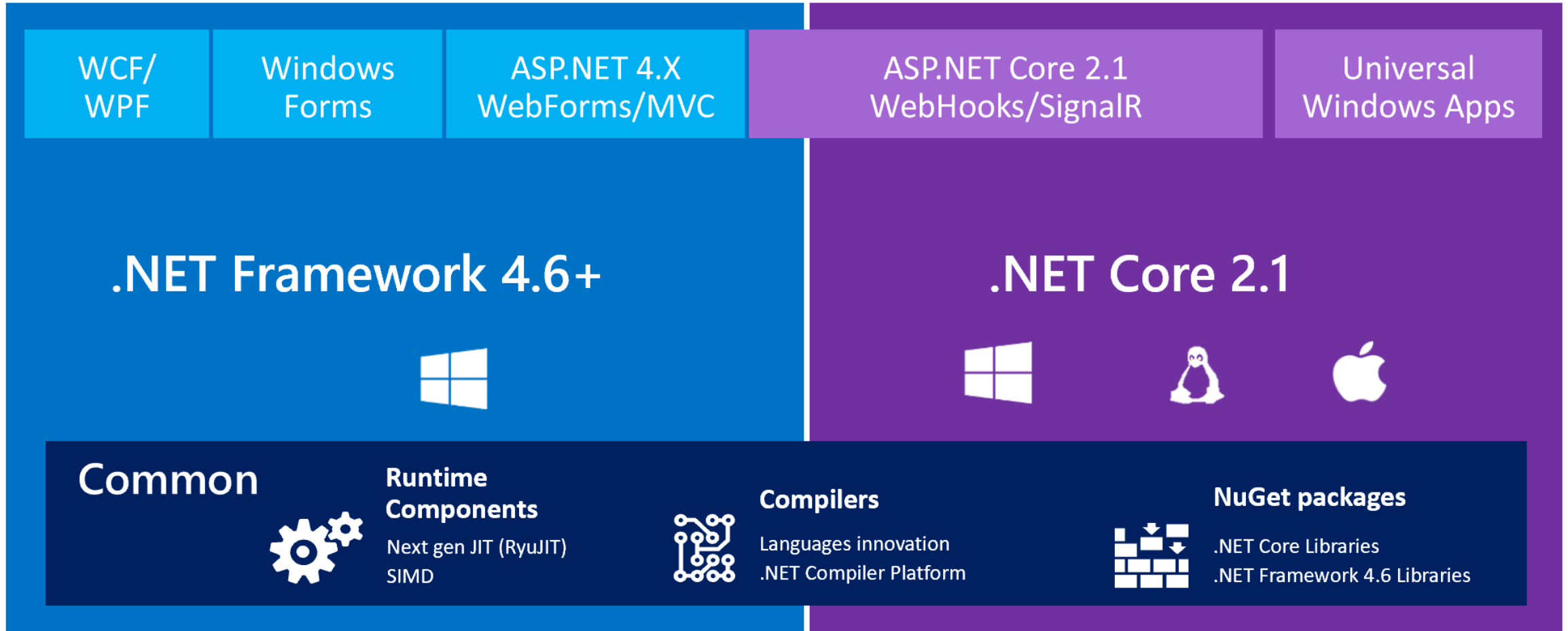
- C++ dilindeki **struct** yapısı C# da korunmuş, Java'da bulunmuyor.
- C# da **goto** kavramı var fakat buna karşın Java da continue ve label ifadeleri bulunuyor.
- C# da **delegate** kavramı ile fonksiyondan fonksiyona parametre aktarımı söz konusu.
- C# da yordamlara parametre iletimi **ref, out** gibi özelliklerle belirtilebiliyor.
- C# da **attribute** kavramı var; Java'da ise derleyiciler için direktifler yok.
- C#' da **pointer** kavramından bahsedilebilir. '**unsafe**' anahtar sözcüğü kullanılırsa bellek erişilebilir duruma geliyor.
- C# da **operatör overloading** varken, Java'da bu kavram yok.

MS.NET Uygulama Geliştirme Platformu

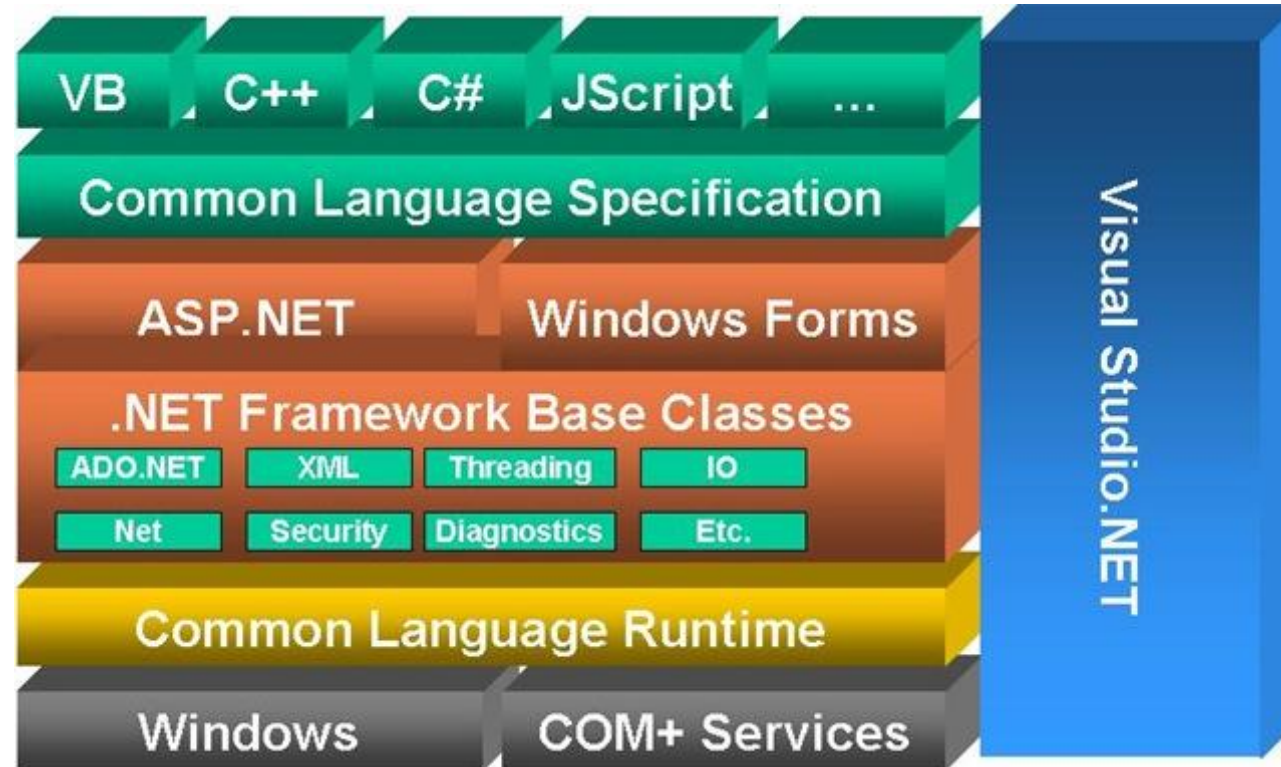
- .NET Framework, Microsoft tarafından geliştirilen, **açık İnternet protokolleri ve standartları** üzerine kurulmuş bir **uygulama geliştirme** platformudur.
- **Masaüstü** uygulamasından, **web** tabanlı uygulamaya, **mobil** uygulamadan, **servis** uygulamalarına kadar her şey bu platform içinde düşünülmüştür ve desteklenmiştir.



MS.NET Uygulama Geliştirme Platformu

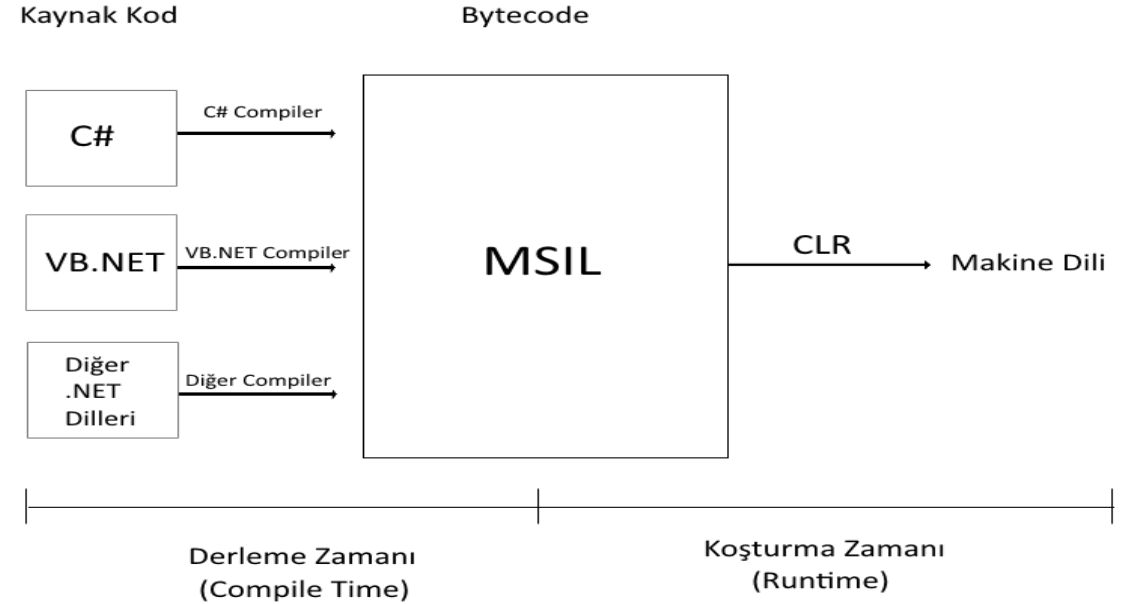


MS.NET Uygulama Geliştirme Platformu



Ortak Dil Çalışma Platformu (CLR)

- Ortak Dil Çalışma Platformu (Common Language Runtime), .NET altyapısında programların çalışmasını denetleyen, programın işletim sistemiyle haberleşmesini sağlayan birimdir.



Ortak Dil Çalışma Platformu (CLR)

- Bir C# kodu yazıp derlediğimizde bu kod **M**icrosoft **I**ntermediate **L**anguage (MSIL) 'a dönüştürülür.
- Bu kod "sözde kod" içeren bir dosyadır.
- CLR, MSIL'i sistem üzerinde çalıştırır. Yani CLR'ın olduğu her ortamda MSIL çalışacaktır.



Ortak Dil Çalışma Platformu (CLR)

C# Kaynak Kodu

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace HelloWorld
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World");
            Console.ReadLine();
        }
    }
}
```

MSIL

```
.method private hidebysig static void Main(string[] args) cil managed
{
    .entrypoint
    // Code size      19 (0x13)
    .maxstack 8
    IL_0000: nop
    IL_0001: ldstr      "Hello World"
    IL_0006: call        void [mscorlib]System.Console::WriteLine(string)
    IL_000b: nop
    IL_000c: call        string [mscorlib]System.Console::ReadLine()
    IL_0011: pop
    IL_0012: ret
} // end of method Program::Main
```

İlk Windows Form Uygulaması

Senaryo: Bu form uygulamamızda kullanıcıdan alınan sayıları toplayan sonrasında ekrana yazdıran bir uygulama yazalım.







- Projeyi oluştur.
- Geliştirme işlemini yap.
- İsimlendirme standartlarına dikkat et.
- Validasyon kontrolleri yap.
- Toplama işleminin tekrar kullanılabilmesi için bir fonksiyon oluştur.
- MessageBox kullan.

İlk Windows Form Uygulaması


Yeni Proje:


Create a new project


Recent project templates


-  Console App (.NET Core) C#
-  ASP.NET Web Application (.NET Framework) C#
-  Blank Solution
-  Windows Forms App (.NET Framework) C#
-  ASP.NET Core Web Application C#
-  Polls Django Web Project Python


windows forms x Language Platform Project type


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

 ASP.NET Web Application (.NET Framework)
Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.
C# Windows Web

 Windows Forms Control Library (.NET Framework)
A project for creating controls to use in Windows Forms (WinForms) applications
C# Windows Desktop Library

 ASP.NET Web Application (.NET Framework)
Project templates for creating ASP.NET applications. You can create ASP.NET Web Forms, MVC, or Web API applications and add many other features in ASP.NET.
Visual Basic Windows Web

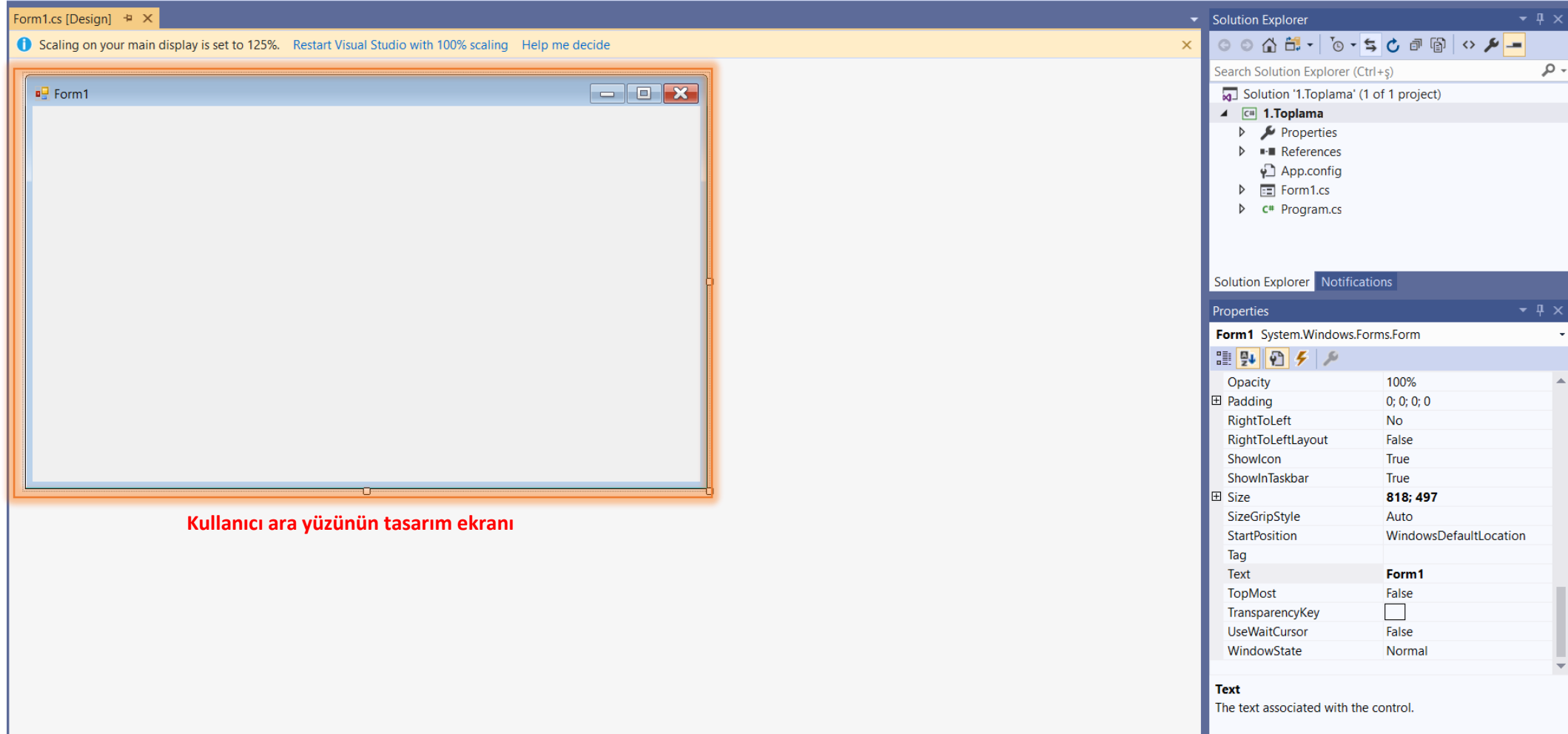
 Windows Forms App (.NET Framework)
A project for creating an application with a Windows Forms (WinForms) user interface
Visual Basic Windows Desktop

 Windows Forms Control Library (.NET Framework)
A project for creating controls to use in Windows Forms (WinForms) applications
Visual Basic Windows Desktop Library

Back Next

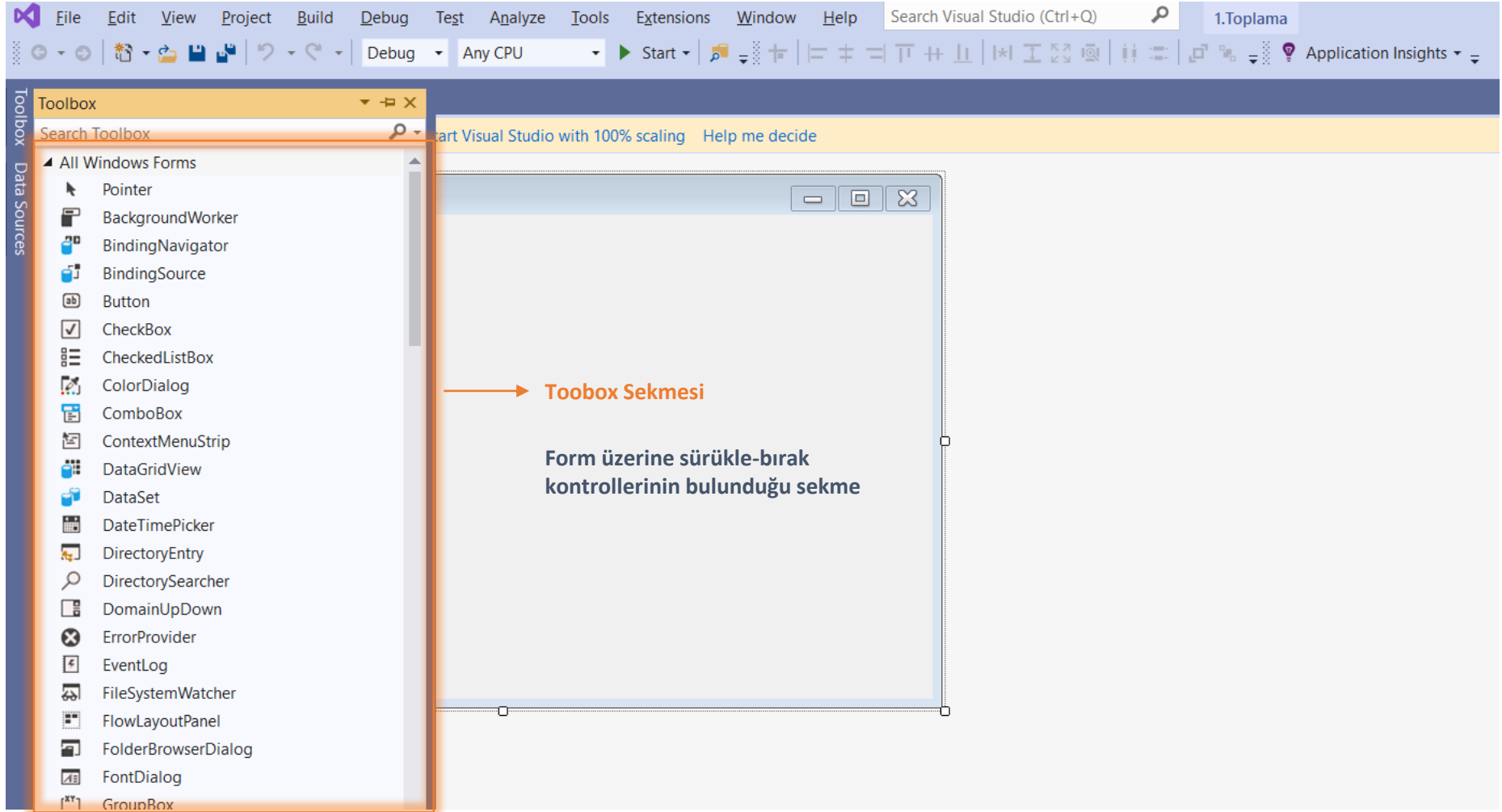
İlk Windows Form Uygulaması

Tasarım:



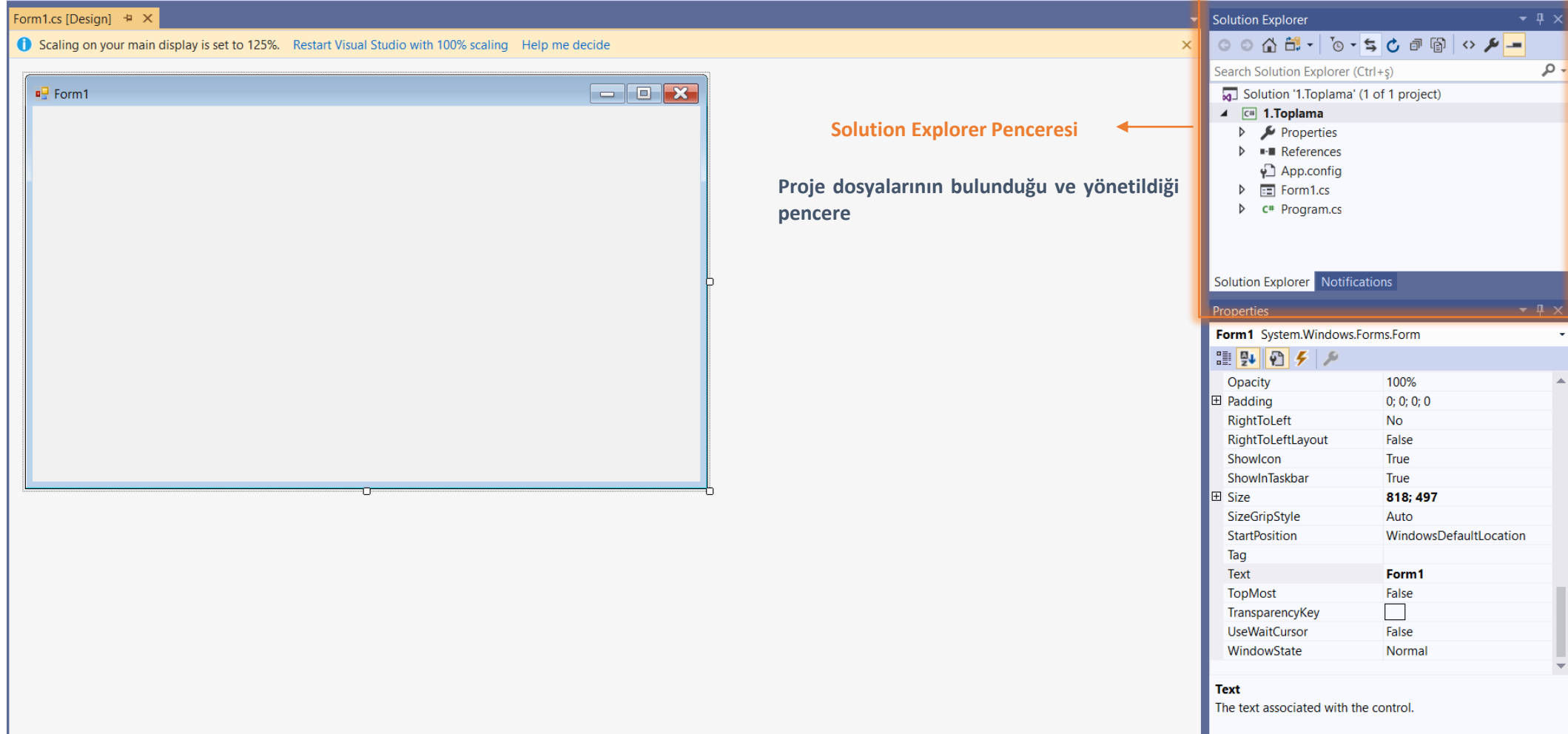
İlk Windows Form Uygulaması

Toolbox:



İlk Windows Form Uygulaması

Solution Explorer:



İlk Windows Form Uygulaması

Properties:

Form1.cs [Design] X

Scaling on your main display is set to 125%. Restart Visual Studio with 100% scaling Help me decide

Form1

Solution Explorer

Search Solution Explorer (Ctrl+ş)

Solution '1.Toplama' (1 of 1 project)

- 1.Toplama
 - Properties
 - References
 - App.config
 - Form1.cs
 - Program.cs

Properties

Form1 System.Windows.Forms.Form

Opacity	100%
Padding	0; 0; 0; 0
RightToLeft	No
RightToLeftLayout	False
ShowIcon	True
ShowInTaskbar	True
Size	818; 497
SizeGripStyle	Auto
StartPosition	WindowsDefaultLocation
Tag	
Text	Form1
TopMost	False
TransparencyKey	
UseWaitCursor	False
WindowState	Normal

Text
The text associated with the control.

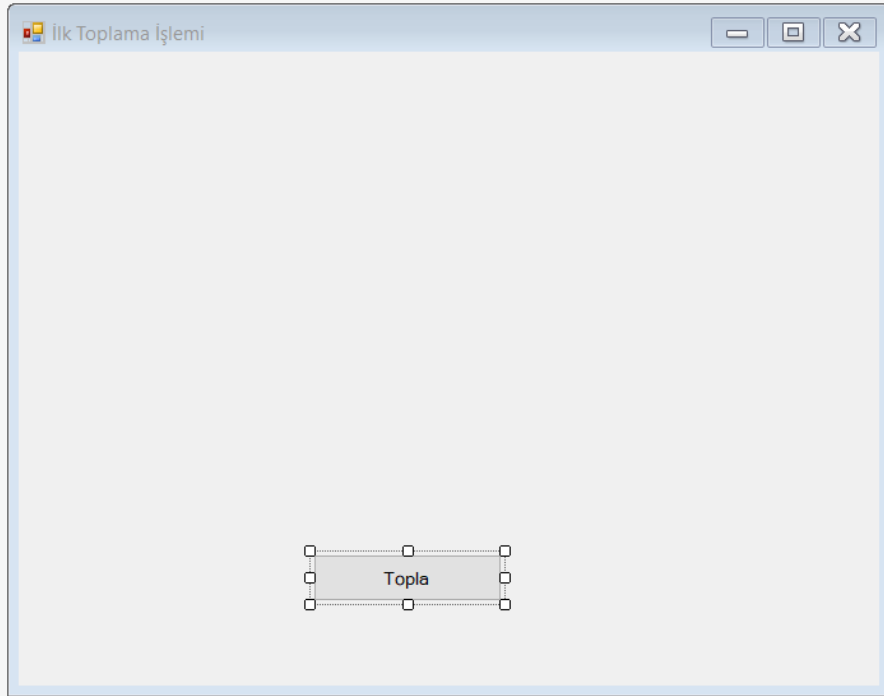
Properties Penceresi

Kullanıcı arayüzünde özellikler ve olayların düzenlendiği pencere

İlk Windows Form Uygulaması

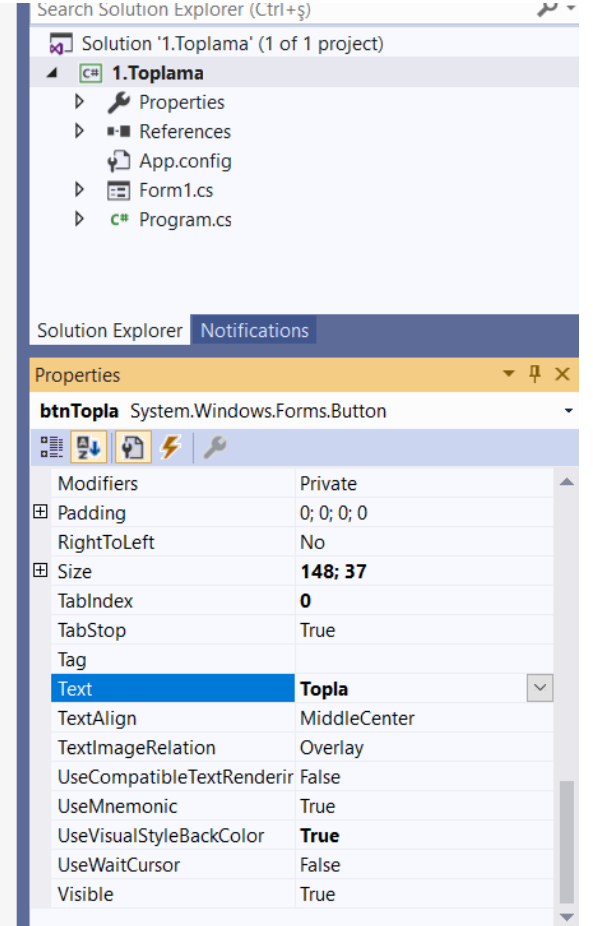
Property Adı	Açıklama
AcceptButton	Aktif formda Enter tuşuna basıldığı zaman hangi butonun çalışacağını belirtir.
BackColor	Aktif formun arka plan rengini düzenlememizi sağlar
CancelButton	Aktif formda Esc tuşuna basıldığı zaman hangi butonun çalışacağını belirtir.
ForeColor	Aktif formda kontroller üzerindeki yazıların renklerini düzenlememizi sağlar
Name	Formun Adını düzenlememizi sağlar
Size	Formun Boyutlarını düzenlememizi sağlar
Text	Formun başlığını düzenlememizi sağlar
Visible	Formumuzun görünürlüğünü düzenlememizi sağlar

İlk Windows Form Uygulaması



Toolbox sekmesinden
sürükleyip bırakarak
formumuza buton
ekliyoruz

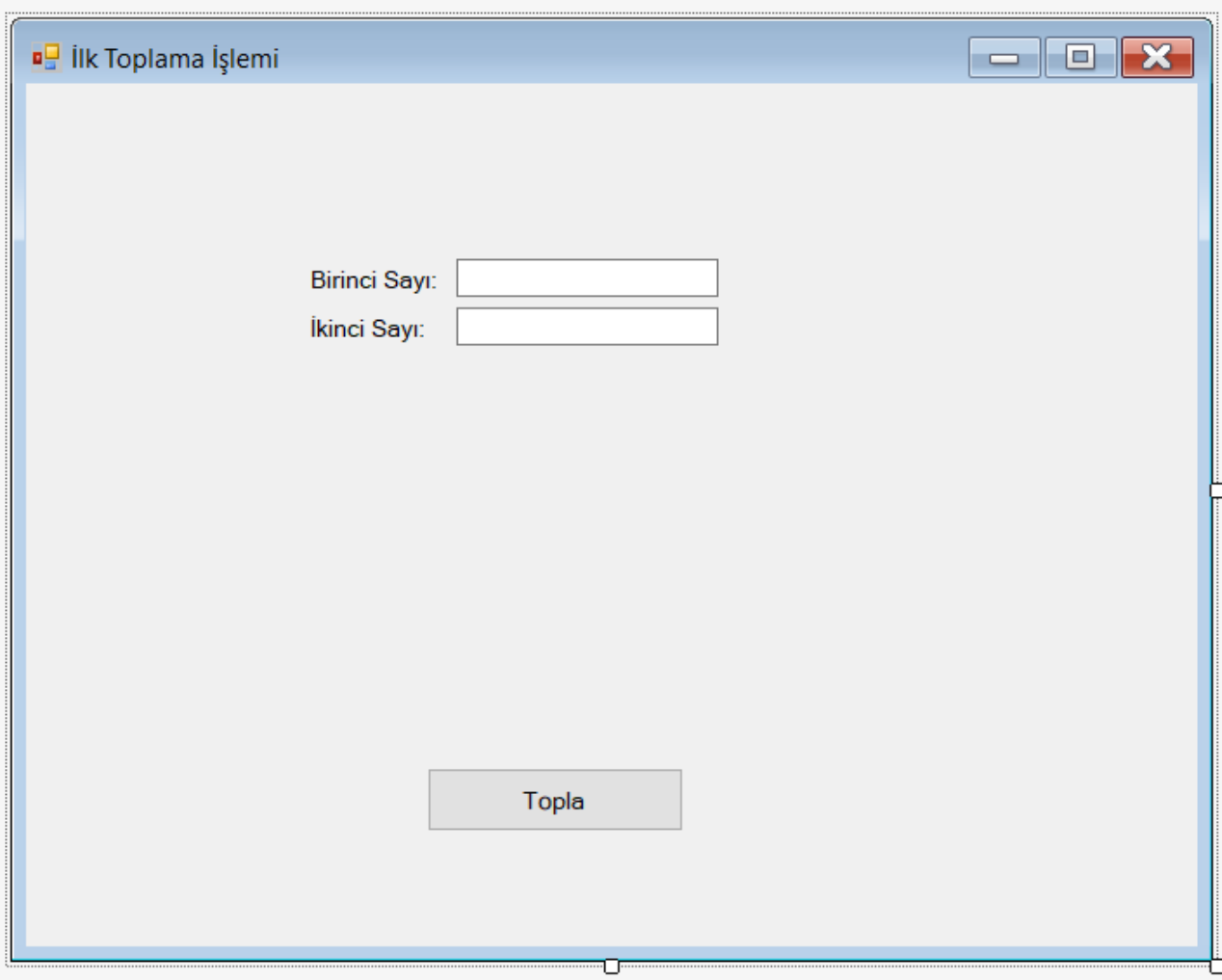
Properties penceresinden
oluşturduğumuz butonun
Text'ini değiştiriyoruz



İlk Windows Form Uygulaması

Property Adı	Açıklama
BackColor	Butonun arka plan rengini düzenlememizi sağlar
Enabled	Butonun Etkin/EtkinDeğil olarak düzenlememizi sağlar
Text	Butonun üzerindeki yazıyı düzenlememizi sağlar
Font	Butonun Text'inin fontunu düzenlememizi sağlar
Name	Butonun adını düzenlememizi sağlar
ForeColor	Butonun Text'inin rengini düzenlememizi sağlar
Size	Butonun boyutlarının düzenlememizi sağlar
Visible	Butonun görünürlüğünü düzenlememizi sağlar

İlk Windows Form Uygulaması



İlk Toplama İşlemi

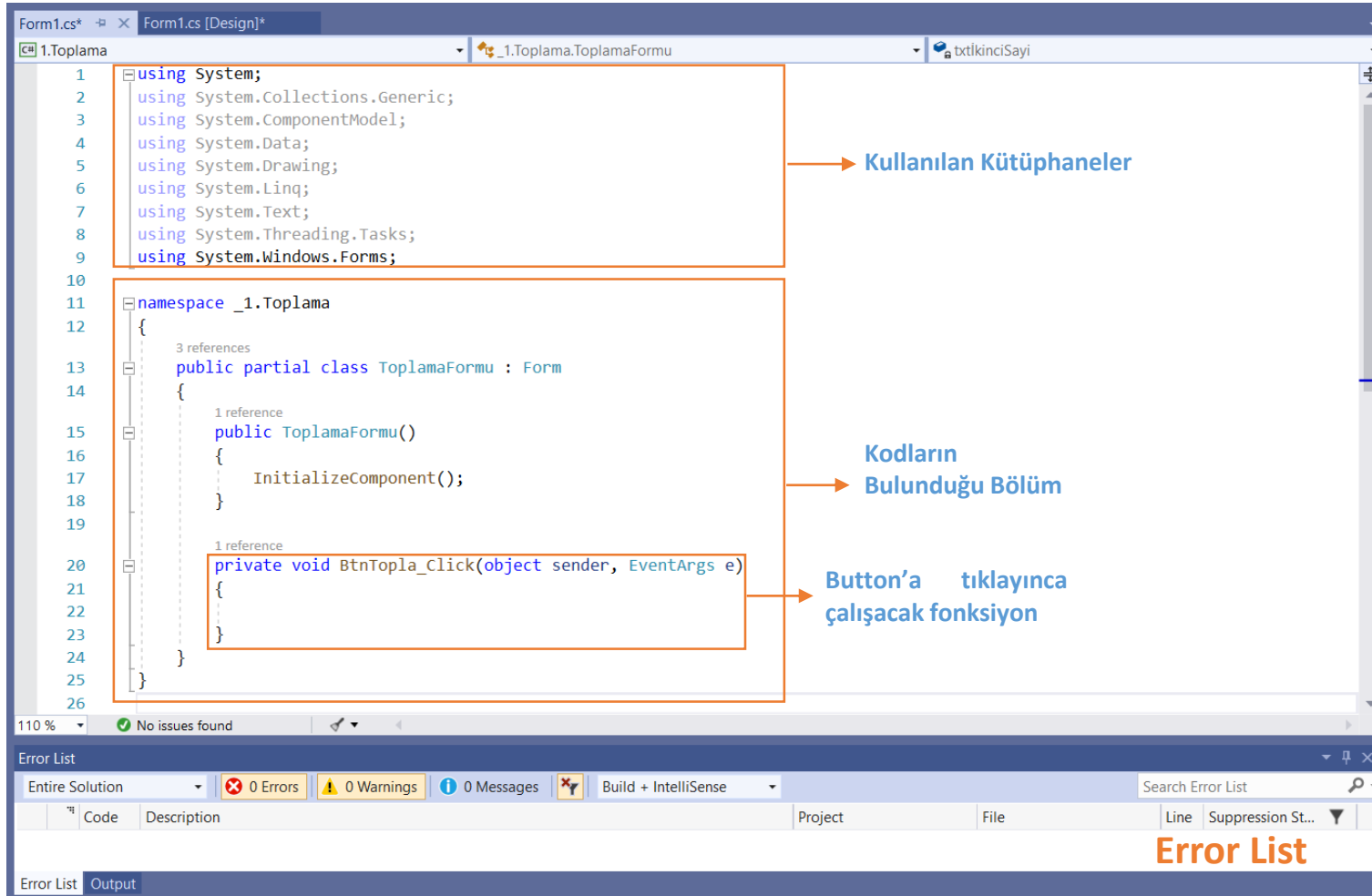
Birinci Sayı:

İkinci Sayı:

Topla

- Form ismini değiştirdik.
- İki label ekledik.
- İki textbox ekledik.
- İsimlendirmeleri yaptık.
- Ekranda biraz küçük kaldılar ve işin doğrusu biraz şekilsizler.
- Yani bir yerlerde eksiklik var ama olsun şimdilik görmezden geleyim ve **«Functionality First»** diyelim 😊

İlk Windows Form Uygulaması



- Butona çift tıkladık ve **kodlama penceresi** açıldı.
- **BtnTopla_Click** isimli bir fonksiyon oluştu.
- Bunun adı **event** yani **olay**.
- Tıklama olayı, ama tek tıklama!
- Her bileşenin birden çok event'i var.
- **Event:** Kullanıcının uygulama bileşenleriyle etkileşime girdiğinde çalışan aksiyon ve kod parçası.

İlk Windows Form Uygulaması

1 reference

```
private void BtnTopla_Click(object sender, EventArgs e)
{
    int birinciSayi, ikinciSayi, toplam;
    birinciSayi = Convert.ToInt32(txtBirinciSayi.Text);
    ikinciSayi = Convert.ToInt32(txtİkinciSayi.Text);
    toplam = birinciSayi + ikinciSayi;
    MessageBox.Show("Toplam: " + toplam);
}
```

Versiyon 1.0

- Kodu yazdık
- İsimlendirme standartlarına uyduk. X, Y vs. kullanmadık.
- Toplam sonucunun ekranda görünmesi için de Mesaj olarak gösterdik.

Eksikler

- Event'lerin içerisinde bu şekilde direk **iş fonksiyon kodu** yazılmaz, ayıptır.
- Validasyon yani giriş doğruluk kontrolü yok. Boşluk ya da metin girilebiliyor.
- Hata yönetimi yok.

İlk Windows Form Uygulaması

```
private int Topla (int sayi1, int sayi2)
{
    return sayi1 + sayi2;
}

1 reference
private Boolean Dogrula()
{
    try
    {
        Convert.ToInt32(txtBirinciSayi.Text);
        Convert.ToInt32(txtIkinciSayi.Text);
        return true;
    }
    catch (Exception)
    {
        return false;
    }
}

1 reference
private void BtnTopla_Click(object sender, EventArgs e)
{
    if (Dogrula())
    {
        int toplam = Topla(Convert.ToInt32(txtBirinciSayi.Text),
                           Convert.ToInt32(txtIkinciSayi.Text));
        MessageBox.Show("Toplam: " + toplam);
    }
    else
        MessageBox.Show("Girdiğiniz değerlerde hata var!");
}
```

Versiyon 1.1

- *Topla()* isimli bir fonksiyonumuz var.
- *Dogrula()* isimli bir metodumuz var.
- Doğrulama için hızlı bir trick yaptık.

Eksikler

- Doğrulama trick iyi değil. Hatanın hangi kontrolde olduğunu anlamıyoruz.
- Ekleme işlemi başarılıysa ekran temizlenmiyor.
- Bileşene focus yok.
- Ekran hazırlama gibi bir fonksiyona ihtiyaç olabilir.

İlk Windows Form Uygulaması

```
private void EkraniAyarla()
{
    txtBirinciSayi.Text = txtİkinciSayi.Text = "0";
    txtBirinciSayi.Focus();
}

private void BtnTopla_Click(object sender, EventArgs e)
{
    if (Dogrula())
    {
        int toplam = Topla(Convert.ToInt32(txtBirinciSayi.Text),
                           Convert.ToInt32(txtİkinciSayi.Text));
        MessageBox.Show("Toplam: " + toplam);
    }
    else
        MessageBox.Show("Girdiğiniz değerlerde hata var!");

    EkraniAyarla();
}

1 reference
private void ToplamaFormu_Shown(object sender, EventArgs e)
{
    EkraniAyarla();
}
```

Versiyon 1.2

- *EkraniAyarla()* isimli bir fonksiyonumuz var. Fonksiyonu ihtiyacımız olduğunda çağırıyoruz.
- *_Shown()* isimli formumuz açıldığında otomatik olarak açılan bir metodumuz/event'imiz var.

Eksikler

- Doğrulama trick iyi değil. Hatanın hangi kontrolde olduğunu anlamıyoruz.
- Ekran UI (ön yüz) tarafını biraz daha iyileştirebiliriz.
- Sorularımız var/olmalı/neler?
 - *Ortamı ve oluşturduğumuz dosyaları biraz daha tanımamız lazım.*
 - Hata olursa nasıl bulacağız? Debug?

İlk Windows Form Uygulaması

Dosya Hiyerarşisi

Ad	Değiştirme tarihi	Tür	Boyut
1.Toplama	25.08.2019 11:25	Dosya klasörü	
1.Toplama.sln	24.08.2019 22:06	Visual Studio Solut...	2 KB

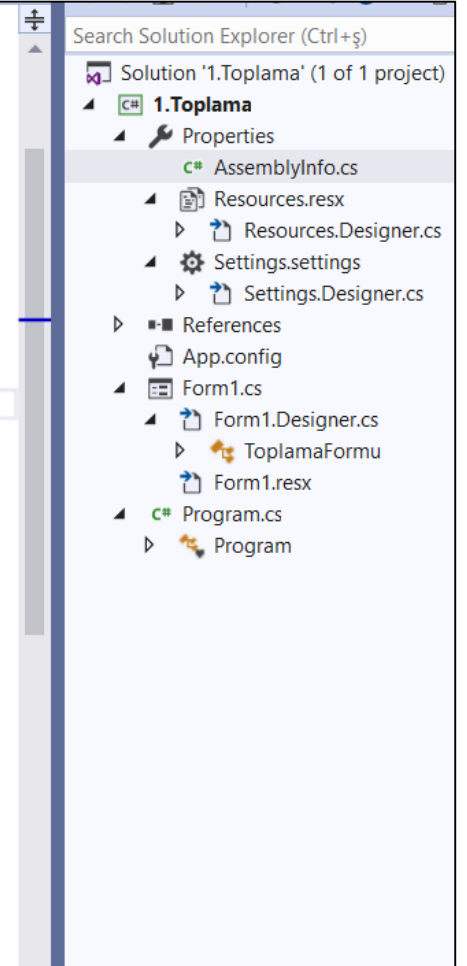
Ad	Değiştirme tarihi	Tür	Boyut
bin	24.08.2019 22:05	Dosya klasörü	
obj	24.08.2019 22:05	Dosya klasörü	
Properties	24.08.2019 22:05	Dosya klasörü	
1.Toplama.csproj	24.08.2019 22:27	Visual C# Project F...	
App.config	25.08.2019 11:25	XML Configuration...	
Form1.cs	25.08.2019 10:07	Visual C# Source F...	
Form1.Designer.cs	25.08.2019 10:02	Visual C# Source F...	5 KB
Form1.resx	25.08.2019 10:02	Microsoft .NET Ma...	6 KB
Program.cs	24.08.2019 22:23	Visual C# Source F...	1 KB

Ad	Değiştirme tarihi	Tür	Boyut
AssemblyInfo.cs	24.08.2019 22:05	Visual C# Source F...	2 KB
Resources.Designer.cs	24.08.2019 22:05	Visual C# Source F...	3 KB
Resources.resx	24.08.2019 22:05	Microsoft .NET Ma...	6 KB
Settings.Designer.cs	24.08.2019 22:05	Visual C# Source F...	2 KB
Settings.settings	24.08.2019 22:05	Settings-Designer ...	1 KB

İlk Windows Form Uygulaması

AssemblyInfo.cs – Assembly Ayarları

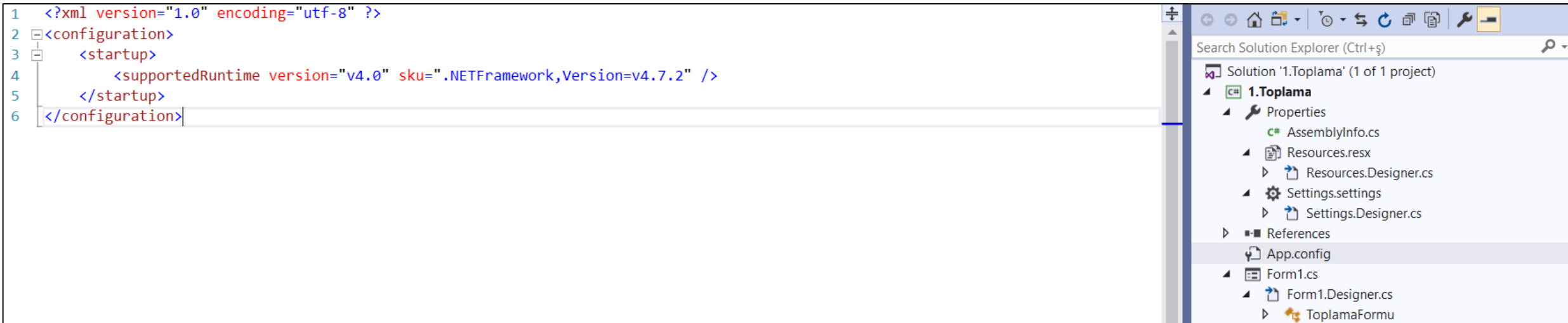
```
7 // associated with an assembly.
8 [assembly: AssemblyTitle("1.Toplama")]
9 [assembly: AssemblyDescription("")]
10 [assembly: AssemblyConfiguration("")]
11 [assembly: AssemblyCompany("")]
12 [assembly: AssemblyProduct("1.Toplama")]
13 [assembly: AssemblyCopyright("Copyright © 2019")]
14 [assembly: AssemblyTrademark("")]
15 [assembly: AssemblyCulture("")]
16
17 // Setting ComVisible to false makes the types in this assembly not visible
18 // to COM components. If you need to access a type in this assembly from
19 // COM, set the ComVisible attribute to true on that type.
20 [assembly: ComVisible(false)]
21
22 // The following GUID is for the ID of the typelib if this project is exposed to COM
23 [assembly: Guid("f6a7b134-95d8-47aa-9b65-58773f8e9ddb")]
24
25 // Version information for an assembly consists of the following four values:
26 //
27 //      Major Version
28 //      Minor Version
29 //      Build Number
30 //      Revision
31 //
32 // You can specify all the values or you can default the Build and Revision Numbers
33 // by using the '*' as shown below:
34 // [assembly: AssemblyVersion("1.0.*")]
35 [assembly: AssemblyVersion("1.0.0.0")]
36 [assembly: AssemblyFileVersion("1.0.0.0")]
```



İlk Windows Form Uygulaması

App.config – Uygulama Parametreleri

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <configuration>
3   <startup>
4     <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
5   </startup>
6 </configuration>
```



The screenshot displays the Visual Studio IDE. The main editor window shows the content of the App.config file, which is an XML configuration file. The code is as follows:

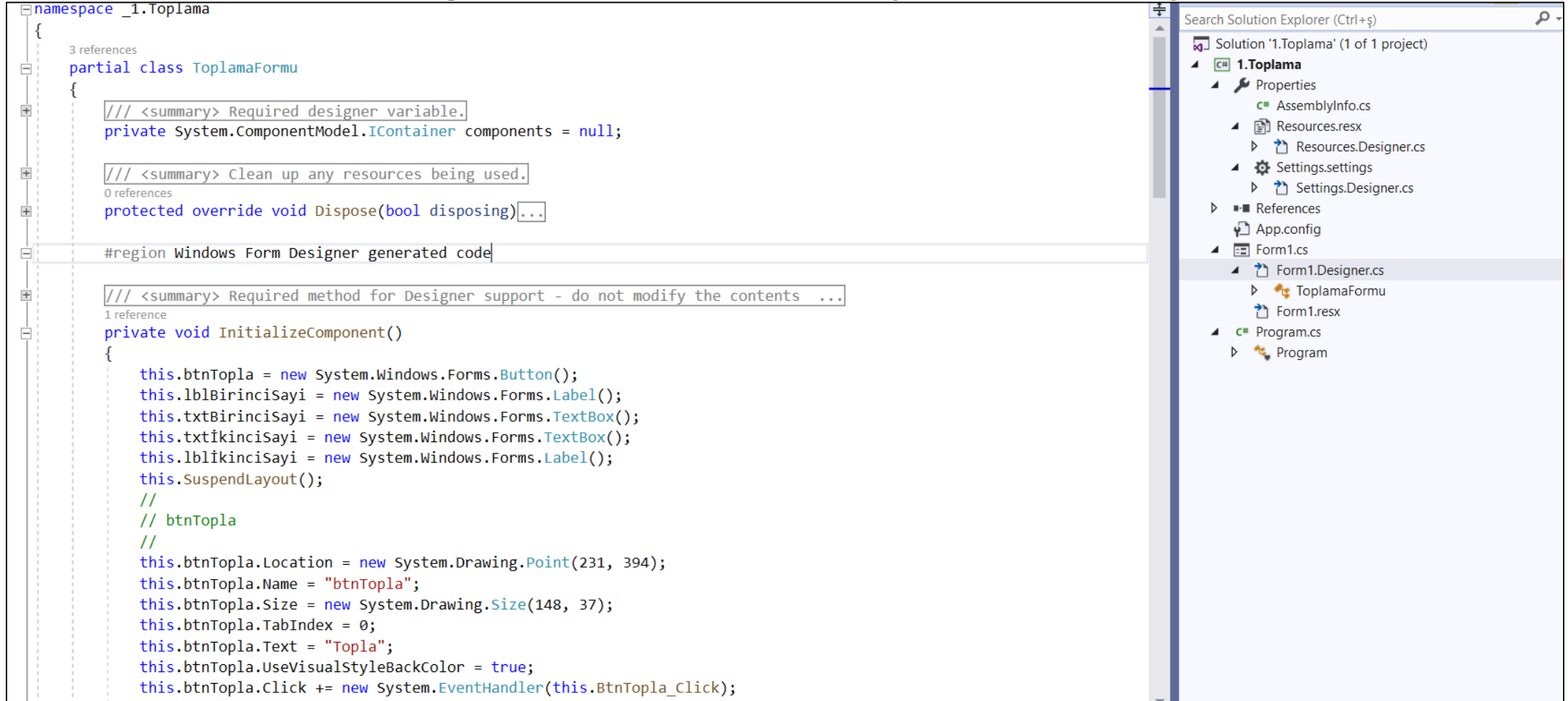
```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <configuration>
3   <startup>
4     <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
5   </startup>
6 </configuration>
```

The Solution Explorer on the right side of the IDE shows the project structure for '1.Toplama'. The files listed are:

- Properties
 - AssemblyInfo.cs
- Resources.resx
 - Resources.Designer.cs
- Settings.settings
 - Settings.Designer.cs
- References
- App.config (selected)
- Form1.cs
 - Form1.Designer.cs
 - ToplamaFormu

İlk Windows Form Uygulaması

Form1.Designer.cs – Form ve UI Bileşenlerinin Detayları



The screenshot displays the Visual Studio IDE. The main window shows the `Form1.Designer.cs` file, which is part of the `1.Toplama` project. The code defines a `ToplamaFormu` class that inherits from `Form`. It includes a `Dispose` method and a `InitializeComponent` method that initializes various UI components.

```
namespace _1.Toplama
{
    3 references
    partial class ToplamaFormu
    {
        /// <summary> Required designer variable.
        private System.ComponentModel.IContainer components = null;

        /// <summary> Clean up any resources being used.
        0 references
        protected override void Dispose(bool disposing)
        {
            //
        }

        #region Windows Form Designer generated code

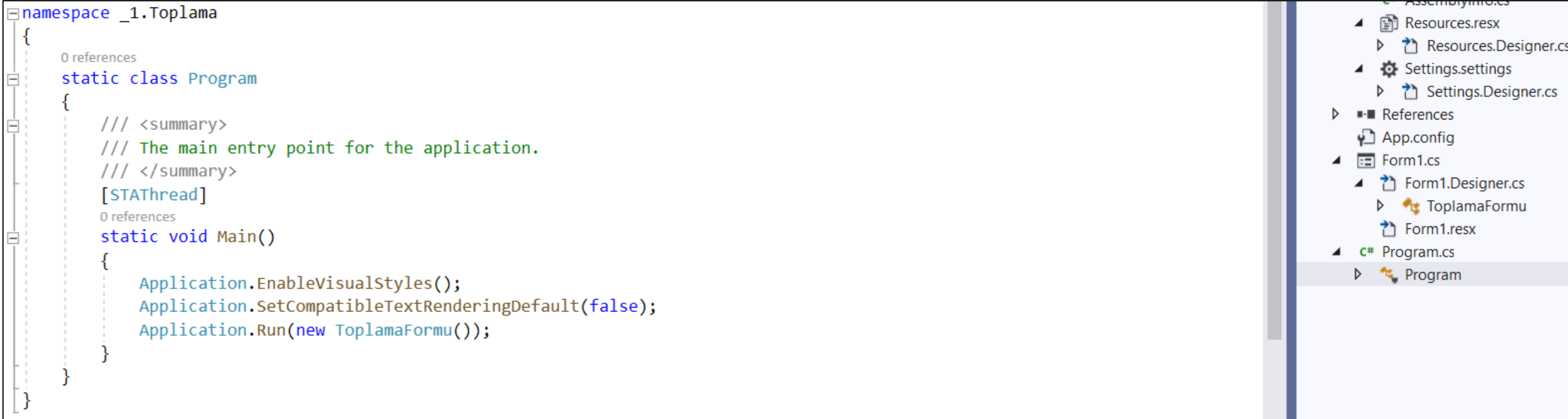
        /// <summary> Required method for Designer support - do not modify the contents ...
        1 reference
        private void InitializeComponent()
        {
            this.btnTopla = new System.Windows.Forms.Button();
            this.lblBirinciSayi = new System.Windows.Forms.Label();
            this.txtBirinciSayi = new System.Windows.Forms.TextBox();
            this.txtIkinciSayi = new System.Windows.Forms.TextBox();
            this.lblIkinciSayi = new System.Windows.Forms.Label();
            this.SuspendLayout();
            //
            // btnTopla
            //
            this.btnTopla.Location = new System.Drawing.Point(231, 394);
            this.btnTopla.Name = "btnTopla";
            this.btnTopla.Size = new System.Drawing.Size(148, 37);
            this.btnTopla.TabIndex = 0;
            this.btnTopla.Text = "Topla";
            this.btnTopla.UseVisualStyleBackColor = true;
            this.btnTopla.Click += new System.EventHandler(this.BtnTopla_Click);
        }
    }
}
```

The Solution Explorer on the right shows the project structure for `1.Toplama`, including files like `AssemblyInfo.cs`, `Resources.resx`, `Resources.Designer.cs`, `Settings.settings`, `Settings.Designer.cs`, `App.config`, `Form1.cs`, `Form1.Designer.cs`, `ToplamaFormu`, `Form1.resx`, `Program.cs`, and `Program`.

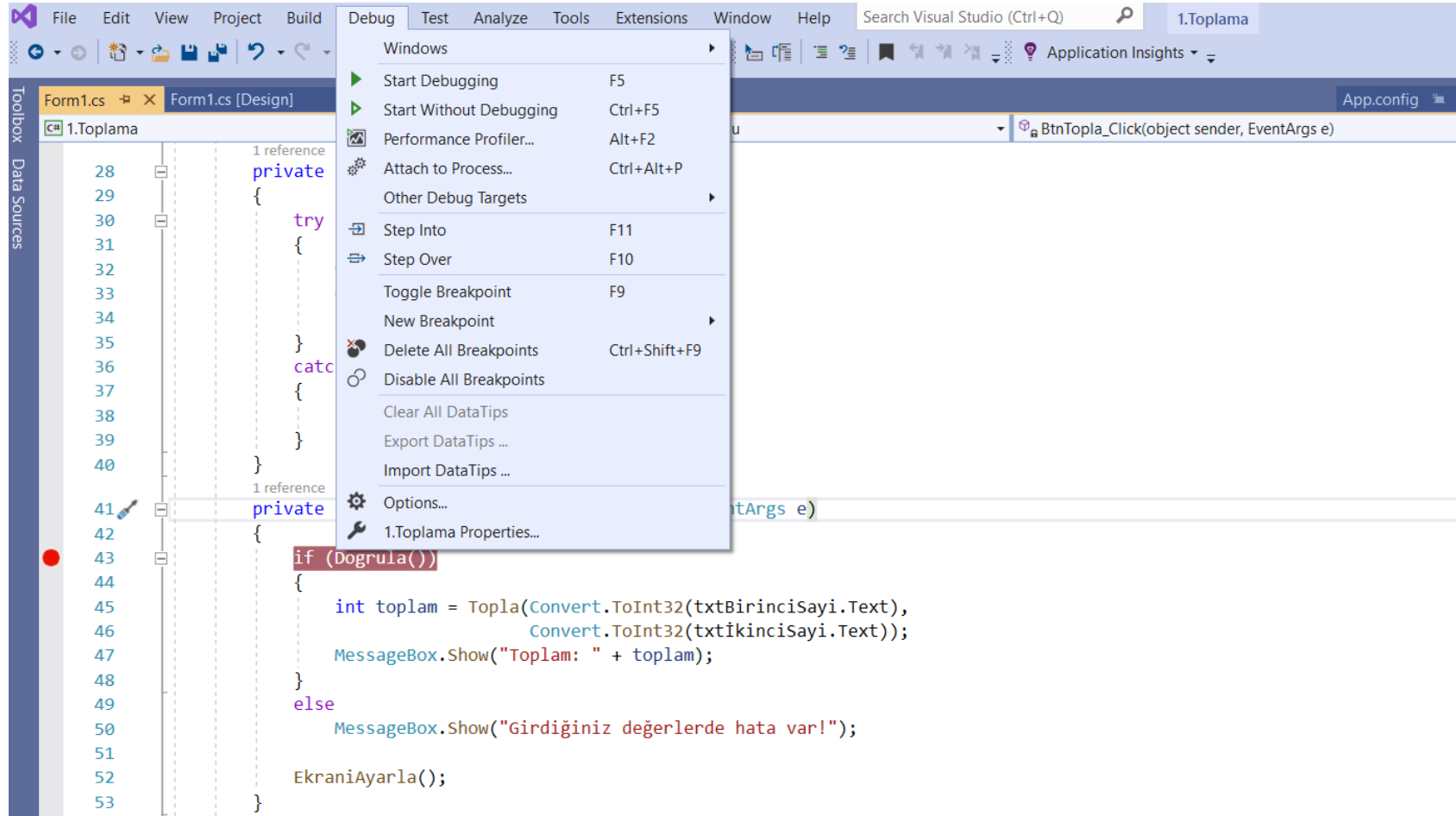
İlk Windows Form Uygulaması

Program.cs – İlk Çalışan Application C# dosyası (Main() fonksiyonu)

```
namespace _1.Toplama
{
    0 references
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        0 references
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new ToplamaFormu());
        }
    }
}
```



İlk Windows Form Uygulaması



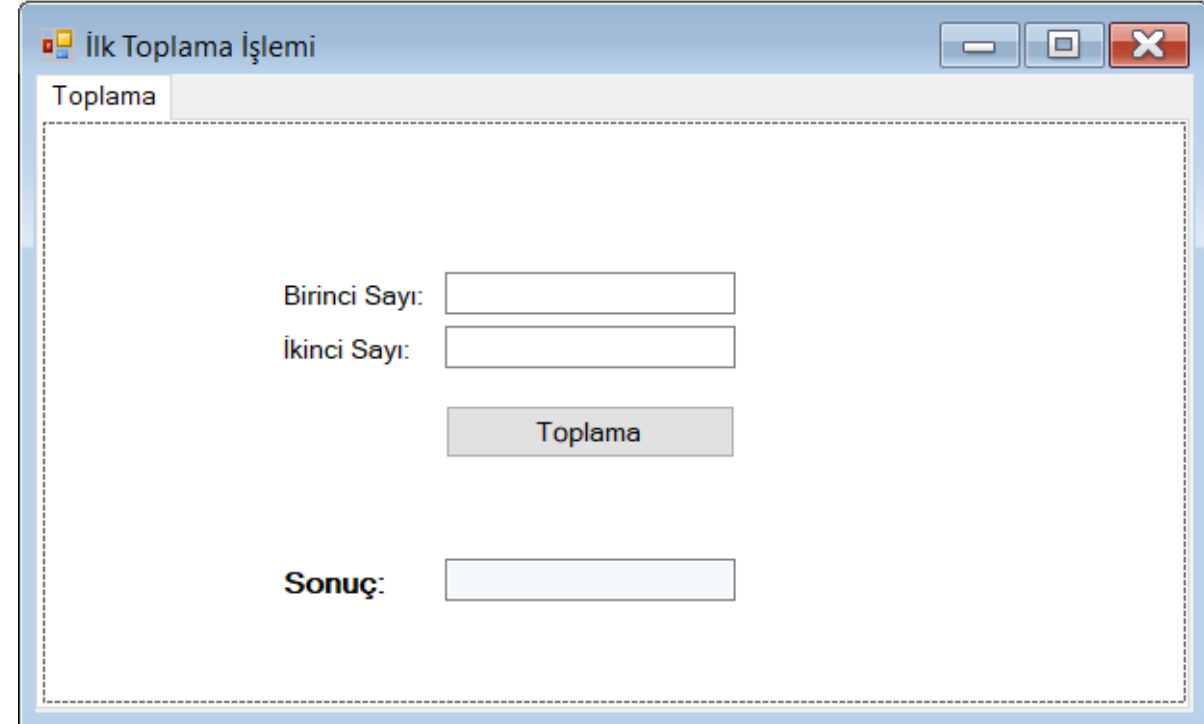
Debug !!!

- *Debug edebilme, uygulamayı adım adım izleyebilmemizi ve hataları bulmamızı sağlar.*
- *Breakpoint koyarak ilgili satırlardan ilerleyebiliriz.*
- *Step-Into ve Step-Over yapabilmek çok önemli.*
- *Breakpoint'ler disable edilebilmeli.*
- *Değişkenlerin değerleri izlenebilmeli.*

İlk Windows Form Uygulaması

Versiyon 1.3 - İhtiyaçlar

- TabControl bileşeni atalım ve bu bileşenin özelliklerini inceleyelim.
 - İki tane tabpage oluşacak, ikincisini silelim.
- Anchor özelliğini görelim (Maksimize edince nasıl görüntü bozuluyor).
- Toplam için yeni bir textbox ekleyelim ve Enabled false yapalım.
- Form açılınca bilgisayar ekranını ortulasın (StartPosition özelliği).
- Toplam değerini bu bileşene yazacak şekilde kodu revize edelim.



The screenshot shows a Windows application window titled "İlk Toplama İşlemi". Inside the window, there is a tab control with a single tab labeled "Toplama". The tab contains the following elements:

- Two text input fields, one labeled "Birinci Sayı:" and the other "İkinci Sayı:", stacked vertically.
- A button labeled "Toplama" positioned below the input fields.
- A text label "Sonuç:" followed by a text input field, positioned below the button.

İlk Windows Form Uygulaması

Versiyon 1.3 - Çıktı

- TabControl işi tamamlandı.
- Anchor işi tamamlandı.
- Form açılınca bilgisayar ekranını ortulasın (StartPosition özelliği).
- Toplam değerini göstermek için yeni bileşen kullanıldı.

Eksikler

- Doğrulama trick iyi değil. Hatanın hangi kontrolde olduğunu anlamıyoruz.
- Yeni bileşen temizlenmiyor. Nasıl temizlenmeli? (Hata/Eksik?)

```
private void BtnTopla_Click(object sender, EventArgs e)
{
    if (Dogrula())
    {
        int toplam = Topla(Convert.ToInt32(txtBirinciSayi.Text),
                           Convert.ToInt32(txtikinciSayi.Text));
        txtSonuc.Text = toplam.ToString();
    }
    else
        MessageBox.Show("Girdiğiniz değerlerde hata var!");

    EkranıAyarla();
}
```

Kodda değişen satır ...

İlk Windows Form Uygulaması

Versiyon 1.3.1 - Çıktı

- EkraniAyarla() fonksiyonu ve Toplama event'i değiştirildi.

Eksikler

- Doğrulama trick iyi değil ya da diğer bir deyişle yeterli değil. Hatanın hangi kontrolde olduğunu anlamıyoruz.
 - Bu eksik beklemeye devam ediyor.
 - Bazı işler böyledir hiç sıra gelmez 😊
 - Burada validasyon için daha **köklü bir çözüm** yapabiliriz/yapmalıyız.
 - **Nedir? Tartışalım...**

Varsayılan değere sahip parametre eklendi...

```
private void EkraniAyarla(bool toplamBasarili = false)
{
    if (!toplamBasarili)
        txtBirinciSayi.Text = txtİkinciSayi.Text = txtSonuc.Text = "0";
    txtBirinciSayi.Focus();
}
```

```
private void BtnTopla_Click(object sender, EventArgs e)
{
    bool dogrulamaSonucu = Dogrula();
    if (dogrulamaSonucu)
    {
        int toplam = Topla(Convert.ToInt32(txtBirinciSayi.Text),
                           Convert.ToInt32(txtİkinciSayi.Text));
        txtSonuc.Text = toplam.ToString();
    }
    else
        MessageBox.Show("Girdiğiniz değerlerde hata var!");

    EkraniAyarla(dogrulamaSonucu);
}
```

İlk Windows Form Uygulaması

Versiyon 1.3.2 - Çıktı

- Textbox bileşenlerine sadece sayı girilmesine izin verildi.
 - Bu özelliğin desteklenmesi için bileşenlerin KeyPress event'leri kullanıldı.

Eksikler

- Textbox bileşen girişinde bir sıkıntı var (mı)?

Yeni Talep

- Toplama butonuna her basıldığında bir loglama yapmak istiyoruz?

Soru?

- Neden sadece tam sayı toplayabiliyoruz? Ondalık sayı toplayamaz mıyız?

```
private void TxtBirinciSayi_KeyPress(object sender, KeyPressEventArgs e)
{
    if (
        char.IsLetter(e.KeyChar) ||
        char.IsSymbol(e.KeyChar) ||
        char.IsWhiteSpace(e.KeyChar) ||
        char.IsPunctuation(e.KeyChar)
    )
        e.Handled = true;
}

1 reference
private void TxtikinciSayi_KeyPress(object sender, KeyPressEventArgs e)
{
    if (
        char.IsLetter(e.KeyChar) ||
        char.IsSymbol(e.KeyChar) ||
        char.IsWhiteSpace(e.KeyChar) ||
        char.IsPunctuation(e.KeyChar)
    )
        e.Handled = true;
}
```

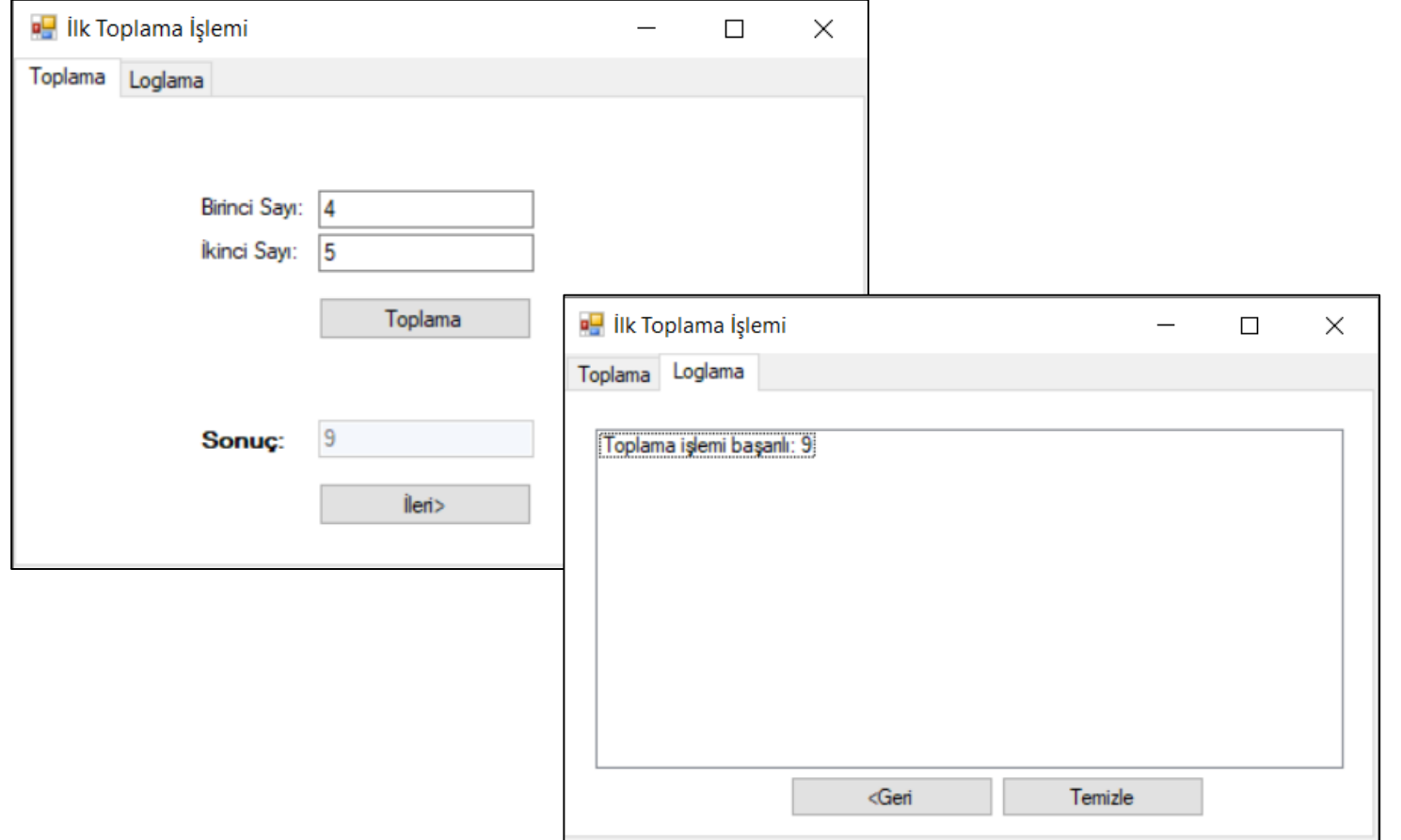
İlk Windows Form Uygulaması

Versiyon 1.4 - Çıktı

- Toplama butonuna her basıldığında bir loglama işlemi yaptık. Bunun için bir listview kullandık. Listview'ı ikinci bir TabControl içerisine yerleştirdik.

Soru?

- Neden sadece tam sayı toplayabiliyoruz? Ondalıklı sayı toplayamaz mıyız?



Yararlanılan Kaynaklar

- Sefer Algan , HER YÖNÜYLE C# , Pusula Yayıncılık, İstanbul, 2003
- Volkan Aktaş, HER YÖNÜYLE C# 5.0 , Kodlab Yayıncılık, İstanbul, 2013
- Milli Eğitim Bakanlığı "Nesne Tabanlı Programlama", 2012

İyi Çalışmalar...

Doç. Dr. Deniz Kılınç

deniz.kilinc@bakircay.edu.tr

drdenizkilinc@gmail.com

www.denizkilinc.com